Zygmunt Vetulani · Joseph Mariani
Marek Kubis (Eds.)

LNAI 10930

Human Language Technology

Challenges for Computer Science and Linguistics

7th Language and Technology Conference, LTC 2015
Poznań, Poland, November 27–29, 2015
Revised Selected Papers

Springer

Lecture Notes in Artificial Intelligence 10930

Subseries of Lecture Notes in Computer Science

More information about this series at http://www.springer.com/series/1244

Zygmunt Vetulani · Joseph Mariani
Marek Kubis (Eds.)

Human Language Technology

Challenges for Computer Science and Linguistics

7th Language and Technology Conference, LTC 2015
Poznań, Poland, November 27–29, 2015
Revised Selected Papers

Springer

Editors
Zygmunt Vetulani
Adam Mickiewicz University
Poznań
Poland

Marek Kubis
Adam Mickiewicz University
Poznań
Poland

Joseph Mariani
LIMSI-CNRS
Orsay Cedex
France

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-93781-6 ISBN 978-3-319-93782-3 (eBook)
https://doi.org/10.1007/978-3-319-93782-3

Library of Congress Control Number: 2018947321

LNCS Sublibrary: SL7 – Artificial Intelligence

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

We began the series of the LTC conferences at a time when Europe was preparing for the “Great Enlargement” of Central European countries, which in particular involved Poland. At that time not only for visionaries, but also for leaders and enlightened European politicians, and “information society” were gaining in popularity. Globalization as a leading trend of technological development, information society as an environment and at the same time the recipient of global products.

The concept of an information society assumes general circulation of information beyond language and cultural barriers while preserving multicultural heritage as a guarantor of the preservation of European multilingualism not only in Europe but on a greater global scale. When in 1995 the co-author of this preface, Zygmunt Vetulani, organized the LTC conference for the first time, it was difficult to predict that a two-day meeting of Polish linguists and computer scientists with experts from the European Commission was a kick-off event of an international conference series to be organized systematically over more than 20 years.

The LTC 1995 meeting was organized in the context of the “Language and Technology: Awareness Days,” a series of European Commission events (DG XIII) located in various European countries, and the aforementioned assumptions recurred in the presentations of eminent experts including Jan Roukens, Antonio Zampolli, and Dafydd Gibbon. The programmatic paper by Jan Roukens emphasized the concern of controlling globalization in a civilized manner. Roukens recalled the findings of the summit in Corfu (June 1994) regarding multilingualism. “The people of the Union, with their different languages, cultures, history and educational systems, should be able to communicate with each other and the external world in ways that allow them to live and work together in an effective, productive, tolerant, democratic and cohesive way in their common European ‘home’.” As a dominant technology, the European Commission advocated – and still advocates – machine translation (Stroerup and Maegaard) as a development paradigm, which a visionary of the field, Antonio Zampolli, postulated for the development of a human language technologies industry. This industry has developed far beyond the scope of these technologies in 1995. At that time, Microsoft was already 20 years old, Apple Computer Inc. was 19, the IBM PC was just 14 years old, the Apple Macintosh was 12 years old, the World Wide Web was only 6 years old, and there was no Google, no Amazon; portable computers still weighted as much as sewing machines and there were no Internet-capable devices like tablets or smartphones. By contrast, the current state of the art includes tablets and smartphones, these small, lightweight ubiquitous universal multimedia devices for gaming, searching, messaging, socially interacting, video conferencing, as well as phoning.

The enlargement of the European Union on 1 May 2004 created a further incentive to draw lessons from the success of the LTC in 1995, which was attended by almost all active researchers in the field of language technology in Poland, and to convene a regular conference on a European scale. We have undertaken the organization of the

LTC conference in Poznań every two years and we consistently fulfil the vision of 1995, as evidenced by the 20th anniversary of LTC in 2015. In the period from 1995 to date, an extensive international community has been created, which, at the time of writing (2018), exceeds 1,200 authors from more than 60 countries around the world. Similarly, the global reach has a community of Program Committee members and reviewers who uphold the high scientific level of the conference presentations. At the same time, we put the emphasis on promoting the best work of young researchers by establishing awards for the best student papers, as well as trying to encourage the continuation of studies recognized as outstanding in the qualification process by creating, in a productive cooperation with Springer publishers, a series of monographs published in the LNAI series where selected conference works are published as Revised Extended Papers, including continuations of the research results presented at the LTC.

The LTC 2015 conference was dedicated to the memory of Adam Kilgarriff, who passed away in 2015. Adam Kilgariff made major contributions in the fields of lexical semantics and corpora. He especially allowed for achieving scientific advances in the semantic disambiguation of word meaning through the organization of evaluation campaigns.

The analytical development of the LTC achievements until 2015 was depicted at the conference in the form of a plenary presentation entitled "Rediscovering 20 Years of Discoveries in Language and Technnology" (Mariani, Paroubek, Francopoulo, and Vetulani), which navigates through the 555 papers and the corresponding 959 authors who contributed to the conference since its launch in 1995 (paper accessible from http://ltc.amu.edu.pl/a2015/).

As a continuation of the series initiated in 2009, LTC 2015 expressed the same interest for less-resourced languages within a special session devoted to that topic, which addressed languages such as Malagasy, Vietnamese, Sambalpuri, Swiss German, Irish, Scottish Gaelic or Welsh, but also Sanskrit or Ancient Greek. An invited talk presented by Dafydd Gibbon specifically stressed the case of endangered languages, while Mikel L. Forcada described the situation regarding machine translation of such languages.

Regardless of the impressive achievements of the past 20 years in many areas, including the development of multimedia technologies, the integration of speech and language technologies, and information and communication technologies for an increasing number of languages, we must note that building a global information society does not bring only advantages, but also dangers. The need for civilized control of the complex processes of creating a global information society has been demonstrated by many violent attacks on civilization as we know, endangered by the activities of terrorist organizations that reject our model of civilized globalization but – paradoxically – use the technological instruments of our information society in order to promote their own fanatical and violent world view. It is evident that – despite the development of the language industries and the global development of technology according to Zampolli's vision, and despite the work done in many cooperative European initiatives such as EAGLES, ISLE, FlaReNet and Meta-Net, as well as long-term regular international conferences in the field such as LREC or LTC – a long road still lies ahead of us if we are to put Roukens's vision of civilized globalization into practice. In other terms, there is still a lot to do for the LTC community.

In this book the reader will find a selection of 31 revised and in most cases substantially extended and updated versions of papers presented at the 7th Language and Technology Conference in 2015. The reviewing process was done by an international jury composed of the Program Committee members or experts nominated by them. The selection was made among 108 contributions presented at the conference and represents the preferences of the reviewers. Finally, the 82 authors of the selected contributions represent research institutions from the following countries: Czech Republic, Canada, France, Germany, Hungary, India, Japan, Nigeria, Poland, Romania, Serbia, Slovakia, Turkey, and the UK.

What are the presented papers about?

The papers selected in this volume belong to various fields of human language technologies and illustrate a large thematic coverage of the LTC conferences. To make the presentation of the papers possibly transparent we have "structured" them into 11 chapters. These are:

1. Speech Processing (4 papers)
2. Multiword Expressions (2)
3. Parsing (1)
4. Language Resources and Tools (4)
5. Ontologies and Wordnets (3)
6. Machine Translation (2)
7. IR/IE (Information and Data Extraction) (3)
8. Text Engineering and Processing (3)
9. Applications (2)
10. Emotions-Decisions-Opinions (EDO) (3)
11. Less-Resourced Languages (LRL) (4)

Clustering the articles into chapters is approximate as many papers addressed more than one thematic area. The ordering of the chapters does not have any "deep" significance; it roughly approximates the order in which humans proceed in natural language production and processing: starting with (spoken) speech analysis, through lexical and morphological analysis, (syntactic) parsing, etc. Within chapters we order the contributions in alphabetical order according to the name of the first author.

Following this thematic order, we start this volume with the "Speech Processing" chapter containing four contributions. In the paper "Intelligent Speech Features Mining For Robust Synthesis System Evaluation" the authors (Moses E. Ekpenyong, Udoinyang G. Inyang, and Victor E. Ekong) present their work on deep neural networks (DNNs) applied to the evaluation of speech synthesis systems. The aim of the paper "Neural Networks Revisited for Proper Name Retrieval from Diachronic Documents" (Irina Illina and Dominique Fohr) is related to the application of neural networks in out-of-vocabulary (OOV) proper names retrieval and the vocabulary extension of a speech recognition system. The third contribution concerning speech, "Cross-Lingual Adaptation of Broadcast Transcription System to Polish Language Using Public Data Sources" (Jan Nouza, Petr Červa, and Radek Šafařík), presents methods and procedures that have been used to adapt to Polish a large-vocabulary continuous speech recognition system (LVCSR) applicable to automatic broadcast transcription. The research on speech segmentation and recognition is the main issue of the last text "Automatic Subtitling System for Transcription, Archiving, and Indexing of Slovak Audiovisual Recordings" (Ján Staš, Peter Viszlay, Martin Lojka, Tomáš Koctúr, Daniel Hládek, and Jozef Juhár).

The "Multiword Expressions" chapter contains two papers. The first one, "SEJF—A Grammatical Lexicon of Polish Multi-Word Expressions" (Monika Czerepowicka and Agata Savary), presents a lexical resource of Polish nominal, adjectival, and adverbial multi-word expressions. The second, "Lemmatization of Multi-Word Entity Names for Polish Language Using Rules Automatically Generated Based on the Corpus Analysis" (Jacek Małyszko, Witold Abramowicz, Agata Filipowska, and Tomasz Wagner) is about the automatic corpus-based lemmatization of multi-word units for highly inflective languages.

The "Parsing" chapter contains a single paper: "Parsing of Polish in Graph Database Environment" (Jan Posiadała, Hubert Czaja, Eliza Szczechla, and Paweł Susicki). The paper presents a rule-based syntactic parsing system for the Polish language using the Langusta natural language processing environment embedded in a graph database.

The "Language Resources and Tools" part comprises of four papers. The contribution "RetroC – A Corpus for Evaluating Temporal Classifiers" (Filip Graliński and Piotr Wierzchoń) presents a corpus for training and evaluating systems for text dating. The authors of the second article in this section, "Reinvestigating the Classification Approach to the Article and Preposition Error Correction" (Roman Grundkiewicz and Marcin Junczys-Dowmunt), reinvestigate the classifier-based approach to article and preposition error correction and claim that state-of-the-art results can be achieved with "(almost) no linguistic knowledge." The next paper, "Binary Classification Algorithms for the Detection of Sparse Word Forms in New Indo-Aryan Languages" (Rafał Jaworski, Krzysztof Jassem, and Krzysztof Stroński), presents an annotation tool used for semi-automatic tagging of new Indo-Aryan texts. The last paper of this chapter, "Multilingual Tokenization and Part-of-Speech Tagging. Lightweight Versus Heavyweight Algorithms" (Tiberiu Boroş and Stefan Daniel Dumitrescu), proposes a framework for mobile devices that offers the essential state-of-the-art NLP tools.

The "Ontologies and Wordnets" chapter is composed of three papers. The paper "A Semantic Similarity Measurement Tool for WordNet-like Databases" (Marek Kubis) proposes a new framework for computing semantic similarity of words and concepts using WordNet-like databases. The next contribution, "Similarity Measure for Polish Short Texts Based on WordNet-Enhanced Bag-of-Words Representation" (Maciej Piasecki and Anna Gut), presents a Wordnet-based approach to semantic comparison of short texts. The last article of this chapter, "Methods of Linking Linguistic Resources for Semantic Role Labeling" (Balázs Indig, Márton Miháltz, and András Simonyi), is concerned with enriching the verb frame database of a Hungarian natural language parser by application of semantic resources as existing linguistic resources such as VerbNet and WordNet.

Two papers constitute the "Machine Translation" section. The authors of the first one, "A Quality Estimation System for Hungarian " (Zijian Győző Yang, Andrea Dömötör, and László János Laki), present their approach to MT quality estimation in opposition to the existing automatic evaluation methods based on reference translations. The second paper, "Leveraging the Advantages of Associative Alignment Methods for PB-SMT Systems" (Baosong Yang, and Yves Lepage) discusses multi-processing-based new ideas in the statistical machine translation.

The "Information and Data Extraction" section contains three contributions. The first one, "Events Extractor for Polish Based on Semantics-Driven Extraction Templates" (Jolanta Cybulka and Jakub Dutkiewicz), presents a tool for identifying events in texts. The next paper, "Understanding Questions and Extracting Answers: Interactive Quiz Game Application Design" (Volha Petukhova, Desmond Darma Putra, Alexandr Chernov, and Dietrich Klakow), presents a tool that extracts answers from unstructured Wikipedia texts. The last paper of the section, "Exploiting Wikipedia-Based Information-Rich Taxonomy for Extracting Location, Creator, and Membership Related Information for ConceptNet Expansion" (Marek Krawczyk, Rafał Rzepka and Kenji Araki), presents a method for extracting a number of semantic relations from Japanese Wikipedia XML dump files.

The next chapter comprises three contributions to "Text Engineering and Processing." The chapter opens with the text "Lexical Analysis of Serbian with Conditional Random Fields and Large-Coverage Finite-State Resources" (Mathieu Constant, Cvetana Krstev and Duško Vitas) describing an approach to lexical tagging of Serbian texts combining three fundamental NPL instruments: part-of-speech tagging, compound and named entity recognition. The paper "Improving Chunker Performance Using a Web-Based Semi-automatic Training Data Analysis Tool" (István Endrédy) follows, focusing on issues related to noun phrase extraction from texts. The third contribution, "A Connectionist Model of Reading with Error Correction Properties" (Max Raphael Sobroza Marques, Xiaoran Jiang, Olivier Dufor, Claude Berrou, and Deok-Hee Kim-Dufor), addresses a connectionist model of written word recognition with correction properties using associative memories.

Two papers are included in the "Applications in Language Learning" chapter. The first one, "The Automatic Generation of Nonwords for Lexical Recognition Tests" (Osama Hamed and Torsten Zesch), proposes an automatic generation of tests for vocabulary proficiency of foreign language students. The second one, "Teaching Words in Context: Code-Switching Method for English and Japanese Vocabulary Acquisition Systems" (Michał Mazur, Rafał Rzepka, and Kenji Araki), presents a system for computer-assisted vocabulary learning using a code-switching method, in application to teaching Japanese vocabulary.

Contributions particularly concerned with linguistic expressions of "Emotions, Decisions and Opinions" were presented at the LTC within the EDO Workshop, integrated with the conference in the form of a special track. Three papers were selected for inclusion in this volume. The contribution "Automatic Extraction of Harmful Sentence Patterns with Application in Cyberbullying Detection" (Michał Ptaszyński, Fumito Masui, Yasutomo Kimura, Rafał Rzepka, and Kenji Araki) describes a method for the automatic detection of malicious content on the Internet. The second paper, "Sentiment Analysis in Polish Web-Political Discussions" (Antoni Sobkowicz and Marek Kozłowski), is a comparative study of dictionary-based versus machine-learning-based methods in sentiment identification in Web discussion texts. Finally, the paper "Saturation Tests in Application to Validation of Opinion Corpora: A Tool for Corpora Processing" (Zygmunt Vetulani, Marta Witkowska, Suleyman Menken, and

Umut Canolat) contributes to the discussion on corpora creation and validation issues with special attention paid to opinion corpora.

"Less-Resourced Languages" are considered of special interest for the LTC community and since 2009 the LRL workshop constitutes an integral part of the conference. In this volume, the LRL workshop is represented by four papers. The paper "Issues and Challenges in Developing Statistical POS Taggers for Sambalpuri" (Pitambar Behera, Atul Kr. Ojha, and Girish Nath Jha) reports on corpus collection and POS-annotation for Sambalpuri – a less-resourced language spoken by some 0.5 million people only and with a small number of written texts available for NLP research. The next one, "Cross-Linguistic Projection for French-Vietnamese Named Entity Translation " (Ngoc Tan Le and Fatiha Sadat), faces the problem of named entity machine translation for the Vietnamese–French language pair. The third article, "National Language Technologies Portals for LRLs: A Case Study" (Delyth Prys and Dewi Bryn Jones), presents the initiative of a new Welsh National Language Technologies Portal as a "resource for researchers, developers in the ICT and digital media spheres, open source enthusiasts and code clubs who may have limited understanding of language technologies but which nevertheless have a need for incorporating linguistic data and capabilities into their own projects, products, processes and services in order to better serve their wider LRL community." The paper "Challenges for and Perspectives on the Malagasy Language in the Digital Age" by Joro Ny Aina Ranaivoarison is the last of this section – as well as of the whole volume – and reports on the ongoing construction of an NLP dictionary of simple words (nouns, adjectives, adverbs, grammatical words) by using conventional dictionaries of the Malagasy language.

We wish you all an interesting reading.

February 2018

Zygmunt Vetulani
Joseph Mariani

Organization

Organizing Committee

Zygmunt Vetulani (Chair)
Jolanta Bachan
Bartłomiej Kochanowski
Marek Kubis (Secretary)
Jacek Marciniak
Tomasz Obrębski
Hanna Szafrańska
Grzegorz Taberski
Daria Waszak
Marta Witkowska
Mateusz Witkowski

LTC Program Committee

Co-chairs: Zygmunt Vetulani, Joseph Mariani

Victoria Arranz
Jolanta Bachan
Núria Bel
Krzysztof Bogacki
Christian Boitet
Gerhard Budin
Nicoletta Calzolari
Nick Campbell
Khalid Choukri
Adam Dąbrowski
Elżbieta Dura
Katarzyna Dziubalska-Kołaczyk
Moses Ekpenyong
Cedrick Fairon
Christiane Fellbaum
Piotr Fuglewicz
Maria Gavrilidou
Dafydd Gibbon
Marko Grobelnik
Eva Hajičová
Roland Hausser
Krzysztof Jassem
Girish Nath Jha
Adam Kilgarriff
Steven Krauwer
Cvetana Krstev
Eric Laporte
Yves Lepage
Gerard Ligozat
Natalia Loukachevitch
Wiesław Lubaszewski
Bente Maegaard
Bernardo Magnini
Jacek Martinek
Gayrat Matlatipov
Keith J. Miller
Asunción Moreno
Agnieszka Mykowiecka
Jan Odijk
Karel Pala
Pavel S. Pankov
Patrick Paroubek
Adam Pease
Maciej Piasecki
Stelios Piperidis
Gabor Proszeky
Adam Przepiórkowski
Georg Rehm
Mike Rosner
Justus Roux
Vasile Rus
Rafał Rzepka
Kepa Sarasola Gabiola
Frédérique Segond
Zhongzhi Shi
Ryszard Tadeusiewicz
Marko Tadić
Dan Tufiş
Hans Uszkoreit
Tamás Váradi
Andrejs Vasiljevs
Cristina Vertan
Dusko Vitas
Piek Vossen
Tom Wachtel
Jan Węglarz
Bartosz Ziółko
Mariusz Ziółko
Richard Zuber

EDO Workshop Program Committee

Co-chairs: Kenji Araki, Paweł Dybała, Bartosz Ziółko

Alladin Ayesh
Karen Fort
Dai Hasegawa
Yasutomo Kimura
Fumito Masui
Mikołaj Morzy
Koji Murakami
Michal Ptaszynski
Tyson Roberts
Rafal Rzepka
Marcin Skowron
Yuzu Uchida
Zygmunt Vetulani
Katarzyna Węgrzyn-Wolska
Adam Wierzbicki
Bartosz Ziółko

LRL Workshop Program Committee

Co-chairs: Khalid Choukri, Joseph Mariani, Claudia Soria, Zygmunt Vetulani

Delphine Bernhard
Laurent Besacier
Nicoletta Calzolari
Jeremy Evans
Mikel Forcada
Daffyd Gibbon
Tatjana Gornostaja
Sabine Kichmeier-Andersen
Andras Kornai
Girish Nath Jha
Maite Melero
Asunción Moreno
Justyna Pietrzak
Stellios Piperidis
Laurette Pretorius
Gabor Proszeky
Georg Rehm
Kevin Scannell
Virach Sornlertlamvanich
Marko Tadić
Marianne Vergez-Couret

SAIBS Workshop Program Committee

Co-chairs: Adam Wojciechowski, Alok Mishra

Frederic Andres
Richard Chbeir
Wojciech Complak
Michele Dasisti
Joao Paulo Costa
Arianna D'Ulizia
Fernando Ferri
Pedro Godinho
Patrizia Grifoni
Patric Hamilton
Mario Lezoche
Alok Mishra
Miroslaw Ochodek
Rory O'Connor
Herve Panetto
Robert Susmaga
Zygmunt Vetulani
Agnieszka Węgrzyn
Adam Wojciechowski
Milan Zdravkovic

Reviewers

Kenji Araki
Jolanta Bachan
Núria Bel
Delphine Bernhard
Laurent Besacier
Krzysztof Bogacki
Christian Boitet
Gerhard Budin
Nicoletta Calzolari
Khalid Choukri
Wojciech Complak
João Paulo Costa
Adam Dąbrowski
Alessia D'Andrea
Elżbieta Dura
Paweł Dybała
Katarzyna Dziubalska-Kołaczyk
Moses Ekpenyong
Jeremy Evas
Christiane Fellbaum
Mikel Forcada
Piotr Fuglewicz
Maria Gavrilidou
Dafydd Gibbon
Tatiana Gornostay
Filip Graliński
Patrizia Grifoni
Tiziana Guzzo
Eva Hajičová
Dai Hasegawa
Krzysztof Jassem
Girish Nath Jha
Sabine Kirchmeier-Andersen
Bartłomiej Kochanowski
Andras Kornai
Steven Krauwer
Cvetana Krstev
Marek Kubis
Eric Laporte
Yves Lepage
Gerard Ligozat
Maciej Lison
Natalia Loukachevitch
Wiesław Lubaszewski
Bente Maegaard
Jacek Marciniak
Jacek Martinek
Fumito Masui
Gayrat Matlatipov
Maite Melero
Alok Mishra
Asunción Moreno
Mikołaj Morzy
Agnieszka Mykowiecka
Tomasz Obrębski
Mirosław Ochodek
Jan Odijk
Jędrzej Osiński
Karel Pala
Pavel S. Pankov
Patrick Paroubek
Maciej Piasecki
Justyna Pietrzak
Stelios Piperidis
Laurette Pretorius
Gábor Proszeky
Michał Ptaszyński
Mike Rosner
Vasile Rus
Rafał Rzepka
Kepa Sarasola Gabiola
Kevin Scannell
Marcin Skowron
Virach Sornlertlamvanich
Robert Susmaga
Michał Szychowiak
Ryszard Tadeusiewicz
Trond Trosterud
Dan Tufiş
Tamás Váradi
Andrejs Vasiljevs
Marianne Vergez-Couret
Grażyna Vetulani
Kadri Vider
Dusko Vitas
Piek Vossen
Tom Wachtel
Adam Wojciechowski
Bartosz Ziółko
Mariusz Ziółko

The reviewing process was effected by the members of Program Committees and invited reviewers recommended by Program Committee members.

Contents

Speech Processing

Multiword Expressions

Parsing

Language Resources and Tools

Ontologies and Wordnets

Machine Translation

Information and Data Extraction

Text Engineering and Processing

Applications in Language Learning

Emotions, Decisions and Opinions

Less-Resourced Languages

Speech Processing

Intelligent Speech Features Mining for Robust Synthesis System Evaluation

Moses E. Ekpenyong(✉), Udoinyang G. Inyang, and Victor E. Ekong

Department of Computer Science, University of Uyo, P.M.B. 1017, Uyo 520003, Nigeria
{mosesekpenyong,udoinyanginyang,victoreekong}@uniuyo.edu.ng,
mosesekpenyong@gmail.com

Abstract. Speech synthesis evaluation involves the analytical description of useful features, sufficient to assess the performance of a speech synthesis system. Its primary focus is to determine the degree of semblance of synthetic voice to a natural or human voice. The task of evaluation is usually driven by two methods: the ***subjective*** and ***objective*** methods, which have indeed become a regular standard for evaluating voice quality, but are mostly challenged by high speech variability as well as human discernment errors. Machine learning (ML) techniques have proven to be successful in the determination and enhancement of speech quality. Hence, this contribution utilizes both supervised and unsupervised ML tools to recognize and classify speech quality classes. Data were collected from a listening test (experiment) and the speech quality assessed by domain experts for naturalness, intelligibility, comprehensibility, as well as, tone, vowel and consonant correctness. During the pre-processing stage, a Principal Component Analysis (PCA) identified 4 principal components (intelligibility, naturalness, comprehensibility and tone) – accounting for 76.79% variability in the dataset. An unsupervised visualization using self organizing map (SOM), then discovered five distinct target clusters with high densities of instances, and showed modest correlation between significant input factors. A Pattern recognition using deep neural network (DNN), produced a confusion matrix with an overall performance accuracy of 93.1%, thus signifying an excellent classification system.

Keywords: Deep neural network · Dimension reduction · Machine learning
Pattern recognition · Speech quality evaluation

1 Introduction

Speech synthesis has remained a challenging task due to high variability in speech signals. This problem stems from the fact that speakers may have dissimilar accents, dialects, or pronunciations, and converse in different styles – at different rates, and in variable emotional states. Also, the presence of environmental noise, reverberation, microphone type and recording devices could introduce further variability. Synthesized speech can be evaluated using different methods and at several levels. All methods give some information on speech quality, but the correctness of the evaluated data can be brought to question. Perhaps the most suitable way to test a speech synthesizer is to select several methods to assess each feature separately. For instance using segmental,

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 3–18, 2018.
https://doi.org/10.1007/978-3-319-93782-3_1

sentence level, prosody, and overall tests together provides lots of useful information, but this on the other hand is labour intensive and time-consuming.

Perceptual Evaluation of Speech Quality (PESQ)[1] represents a family of standards comprising a test methodology for automated assessment of speech quality – as experienced by a user of a telephony system. It is an objective method developed to model subjective tests commonly used in telecommunications (e.g. ITU-T P.800) for the assessment of voice quality, perceived by humans. Although PESQ utilizes true voice samples as test signals, and has been proved to be the most reliable measure for assessing speech quality, it is computationally demanding and requires access to the whole utterance. In some applications, this might not be acceptable. Ideally, the objective measure should predict the quality of speech independent of the type of distortions introduced by the system – be it a network, a speech coder or a speech enhancement algorithm. Hence, many systems are now optimized for speech and can respond in an unpredictable way to signal degradation.

1.1 Speech Synthesis Framework

The Hidden Markov Model (HMM) speech synthesis [1, 2] has provided a flexible framework for synthesizing speech. Although HMM has tremendously improved flexibility by enabling: (i) varying speaking styles and modified speaker/voice characteristics; and (ii) small memory footprint and robustness; it however suffers from speech quality degradation [3], compared to the unit selection approach [4, 5]. Factors responsible for the degraded quality include: over-simplified vocoder techniques; acoustic modelling inaccuracy; and over-smoothing of the generated speech parameters [2].

Recently, Deep Neural Networks (DNNs) have achieved great improvements in both quality and accuracy in speech synthesis [3], as their trainings converge faster and outperform other approaches such as Gaussian Mixture Models (GMMs) and HMMs – if their initial parameter values are pre-trained instead of randomly initialized [6]. The pre-training methods use unsupervised techniques [7]. DNN simulates human speech production as a layered hierarchical structure that transforms linguistic texts into speech utterances. The Neural Network (NN) is trained to map the input phonetic transcriptions of the training text into sequences of acoustic feature vectors, sufficient to yield predefined speech waveforms when processed by a signal generation module. The training data may correspond to written transcription of speech carried in a predefined speech waveform. Before training a DNN, a set of utterances is transcribed phonetically with sequences of phonetic-context descriptors – each containing a set of phonetic speech units – indicating contexts of the respective speech units and their durations. The trained NN may then map the sequence of phonetic-context descriptors to predicted feature vectors, which are finally transformed into synthesized speech by the signal generation module. DNN-based speech synthesizers are therefore (more) likely to overcome the limitations of HMMs, given its high accuracy level and intelligent prediction models. Deep learning is at the intersection of neural networks, graphical modelling, optimization, pattern recognition, and signal processing. Deep learning methods have grown

[1] http://www.itu.int/rec/T-REC-P.862/en.

increasingly richer, encompassing those of neural networks, hierarchical probabilistic models, and a variety of unsupervised and supervised feature learning algorithms [8]. The three main reasons for the wide applications of deep learning in recent times are the drastically increased chip processing abilities, the reduced cost of computing hardware, and the recent advances in machine learning and signal/information processing research. These advances have enabled the deep learning methods to effectively exploit complex, composite non-linear functions, to learn distributed and hierarchical feature representations, and to make effective use of both labelled and unlabeled data. Three major categories of deep learning networks include [8]:

(i) Unsupervised or generative deep learning networks, which captures high-order correlation of the observed or visible data for pattern analysis, or synthesis purposes when no information about target class labels is available. Subclasses of generative or unsupervised deep networks includes, the energy-based deep models;

(ii) Deep networks for supervised learning, which are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data. Target label data are always available in direct or indirect forms for such supervised learning. They are also called discriminative deep networks;

(iii) Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. The goal can also be accomplished when discriminative criteria for supervised learning are used to estimate the parameters in any of the deep generative or unsupervised deep networks in category 1 above.

In [9], DNN was used as an intelligent tool for robust evaluation of speech synthesis systems. Data for the training were measures obtained from listening experiments and an objective evaluation of the various listening tests (to establish the speech quality range) by experts. The evaluation measures or criteria considered *naturalness*, *intelligibility*, *comprehensibility*, *tone correctness*, *vowel correctness* and *consonant correctness*. They observed that a 10-layer DNN gave better performance compared to 3- and 5-layer DNNs.

In this paper the work done in Ekpenyong, Inyang and Ekong [9] is extended, by studying the mappings and patterns presented by expert knowledge for efficient knowledge synthesis and discovery. To achieve this, we propose a framework that integrates a self organizing map (SOM) – an unsupervised technique useful for visualizing patterns inherent in the evaluation measures obtained in the listening test experiment [9]; and a DNN-Pattern recognizer – a supervised machine learning approach – for efficient classification of the simulated target classes. Early speech synthesizers were evaluated primarily for intelligibility and naturalness. But in [9], comprehensibility, tone correctness, vowel correctness, and consonant correctness were added, to examine their relevance in the overall speech quality prediction.

This paper proceeds as follows: Sect. 2 discusses speech synthesis evaluation and the approaches involved; Sect. 3 presents the proposed methodology; Sect. 4 discusses the results obtained; and, Sect. 5 concludes the paper with a pointer to future work.

2 Speech Synthesis Evaluation

Speech synthesis evaluation involves the analytic description of system performance in terms of defined factors, and focuses on whether or not the voice is accepted as quality – by attempting to discover how closely it is to the human voice [10–12]. Synthesized speech may be evaluated with regards to the following measures: intelligibility, naturalness, and appropriateness for used application [13, 14], but most applications discriminate these measures based on the target users. For instance, in reading systems for the blind, intelligibility with high speech rate is often the most preferred measure than naturalness. Conversely, prosodic features and naturalness are essential for multimedia applications or electronic mail readers. The evaluation procedure is usually done by subjective listening tests with response to a set of syllables, words, sentences, or with other questions. The test material is mainly focused on consonants, because they are more problematic to synthesize than vowels. Precise and reliable assessment of speech quality is becoming vital, to meet with users' satisfaction of the deployed speech processing system. Individual users have varying internal standards of what constitutes "excellent" or "poor" speech quality, which results in wide variability in rating scores among listeners. Several methods for evaluating synthetic speech have evolved over the last decades. The most commonly used methods for evaluating intelligibility include: *segmental evaluation methods* (where the diagnostic and modified rhyme tests are the most famous – [15, 16]; *sentence level tests* (where the Haskins sentences and semantically unpredictable sentences (SUS) are the notable ones [17]; and c*omprehension tests* [18, 19]. To evaluate the overall speech quality, the following methods are often preferred: *mean opinion score (MOS)* Goldstein [15]; categorical estimation (CE) [20]; *pairwise comparison (PC)* [20]; and, *magnitude and ratio estimation* [21]. Two approaches are outstanding in the evaluation of synthetic voices: the subjective and objective approaches. These approaches are discussed in the following sub sections.

2.1 Subjective Approach

The most reliable method of speech synthesis evaluation relies on measurement of the perceptual performance of human listeners (often referred to as ***subjective*** test of speech quality), yet there are strong demands for ***objective*** evaluation and is widely driven by the desire to determine the quality of speech technology systems, without requiring significant number of human listeners [22]. Although subjective techniques to evaluating synthetic speech make use of human experts, the expertise required may vary from technique to technique – ranging from untrained native speakers of the language to trained phoneticians. However, subjective votes are heavily influenced by numerous factors such as the preferences of individual subjects and the experiment context. Robust procedures using untrained listeners would of course be extremely useful since besides the extra cost of sourcing for trained phoneticians, there is also the danger of experts being influenced by theoretical biases, particularly in such controversial domain such as prosody

2.2 Objective Approach

Measuring whether or not an utterance from a synthetic voice acoustically matches the same utterance in human speech is beneficial for a number of reasons. First, by their very nature, objective measures offer a clear measurement of how a voice is performing and can be used to diagnose problem areas requiring development. Also, compared to the time and cost involved in user testing, objective measures can be an efficient form of evaluation. However, findings from such evaluations may not always match up with listener perceptions. Clark and Dusterhoff [23] have shown that some objective measures may be oversensitive compared to the human ear. The opposite may also be true, in that synthetic samples may be perfect in acoustic terms, but may be perceived as unnatural by listeners. Morton [12] had explained this based on the processing required to listen to speech. He suggested that human speakers naturally vary their speech, with the aim of being understood, and that this affects the way human speech is processed by listeners. Since a synthesizer does not have this capability, the processing needs of the listener are not accounted for. And this may cause the listener to find the synthetic speech unnatural or difficult to understand. Objective/acoustic measures can be used in evaluation of different aspects of a voice, such as prosodic features or intonation. Examples of acoustic measures which may be of interest include fundamental frequency (F0), segmental duration and intensity. A common approach is to use statistical methods (Root Mean Squared Error: RMSE) to model the expected performance of the voice against actual performance, measuring the accuracy of the synthetic voice against a natural voice. Whilst objective measures have been described as more efficient than user testing, of course specialist knowledge is required in order to run these tests. Objective measures can be a useful means of evaluation for synthetic voices, and can be a very efficient form of testing. However, such techniques require specialist knowledge.

3 System Design

3.1 Data Collection and Representation

The data collection and initial pre-processing methodology is conceptualized in Fig. 1. After a HMM-based synthesis experiment, the data collection spanned two stages. First, a listening test was performed on a set of synthesized voices. Second, domain experts' assessments were used for validating listeners' responses. Speech quality (SQ) measures considered in this experiment were *naturalness, intelligibility, comprehensibility, tone correctness, vowel correctness and consonant correctness*. The mean opinion score (MOS) test was used to determine the naturalness of the synthesized voices, while the Modified Rhyme Test (MRT) was the method for evaluating intelligibility of the voices, both of which were in the range (1–5). In order to extract results for tone, vowel and consonant correctness, the MRT results were extrapolated through a comparison of the listeners' responses with a list of correct utterances. The resultant values are elements of the set {0, 1}, corresponding to true (correct) or false (wrong). Semantically Unpredictable Sentences (SUSs) – i.e., sentences that are syntactically correct, but semantically anomalous, were used to rank the comprehensibility of the synthetic voices in the range (1–4). To achieve this, listeners were made to comprehend a set of SUSs – by listening to the sentences (at least three times), before typing what they've heard. Although SUSs seals the listeners from contextual cues that contribute to perceived intelligibility, it however maintains a reliable evaluation that keeps test results undistorted.

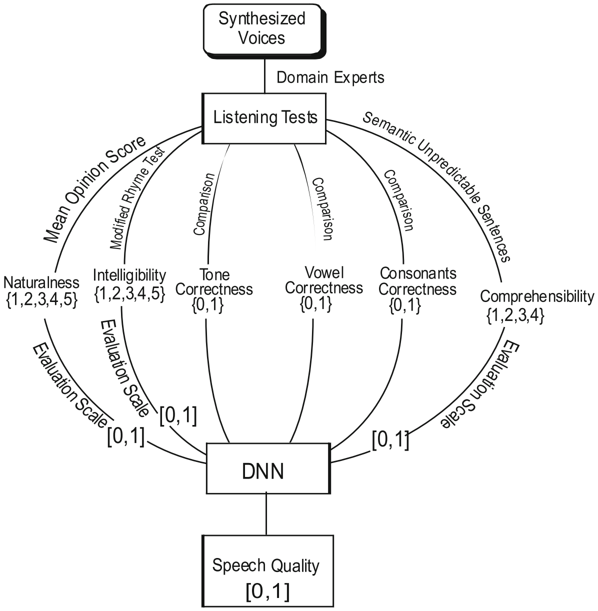


Fig. 1. Data collection and pre-processing work-flow

The coding of the dataset attributes for the experiment is as summarized in Table 1.

Table 1. Input parameter description and coding

S/No.	Input parameter	Modality	Code
1.	Naturalness	Mean opinion score	5-Excellent, 4-V. good, 3-Good, 2-Fair, 1-Poor
2.	Intelligibility	Modified rhyme test	5-Excellent, 4-V. good, 3-Good, 2-Fair, 1-Poor
3.	Tone correctness	Comparison	1-Correct, 2-Wrong
4.	Vowel correctness	Comparison	1-Correct, 2-Wrong
5.	Consonant correctness	Comparison	1-Correct, 2-Wrong
6.	Comprehensibility	SUS	4-High, 3-Medium, 2-Low, 1-No comprehension
7.	Voice quality	DNN	1-Very high, 2-High, 3-Medium, 4-Low, 5-V.low

The results obtained for naturalness, intelligibility and comprehensibility were further transformed to yield results in the range (0–1). Our dataset comprised 563 data points, out of which, 393 (70%) records was randomly selected for training the DNN-Pattern recognizer; while 15% (85) each was selected as test and validation datasets. A section of the sample dataset is shown in Table 2. The target classes were obtained through expert evaluation, and include: 1-Poor, 2-Fair, 3-Good, 4-V.Good, 5-Excellent

Table 2. Dataset for speech synthesis evaluation

S/no.	Input class							Target class				
	Nat	Intell	Comp	T	V	C	SQ	C1	C2	C3	C4	C5
1.	0.8757	0.8838	0.7505	1	1	1	5	0	0	0	0	1
2.	0.3588	0.6361	0.4129	1	1	1	4	0	0	0	1	0
3.	0.6485	0.8068	0.8187	0	1	1	5	0	0	0	0	1
4.	0.5144	0.5778	0.6667	1	1	1	3	0	0	0	0	1
5.	0.6137	0.7873	0.7713	0	1	1	5	0	0	0	0	1
6.	0.7447	0.6529	0.8115	1	1	1	4	0	0	0	0	1
7.	0.3728	0.5286	0.7491	0	1	1	3	0	0	0	1	0
8.	0.7218	0.7492	0.6195	1	0	1	4	0	0	0	0	1
9.	0.4642	0.5316	0.8104	1	1	1	3	0	0	0	0	1
10.	0.6730	0.7231	0.8259	1	1	0	4	0	0	0	0	1
11.	0.5858	0.5360	0.4565	1	1	1	3	0	0	0	0	1
12.	0.2476	0.5487	0.0292	1	1	1	3	0	0	0	1	0
13.	0.5759	0.5639	0.4016	0	1	1	3	0	0	0	1	0
14.	0.6020	0.5421	0.4277	0	1	1	3	0	0	0	1	0
15.	0.5723	0.4715	0.7010	1	1	1	3	0	0	0	0	1
16.	0.5452	0.7074	0.5017	1	1	1	4	0	0	0	0	1
17.	0.4354	0.7020	0.4864	1	1	1	4	0	0	0	1	0
18.	0.5682	0.5563	0.3072	1	1	1	3	0	0	0	0	1
19.	0.5795	0.3081	0.0457	0	1	1	2	0	0	1	0	0
20.	0.5151	0.6520	0.4131	1	1	0	4	0	0	0	1	0

3.2 SOM Methodology

SOM is used to divide the input data points into one of several groups. Training data are provided to the SOM. During training phase, the SOM groups the input data into clusters, where data with the most similar properties are clustered together. The training process in SOM is unsupervised and draws its strengths from the pioneering works of Kohonen [24]. SOM forms clusters by itself on the basis of training data, and then places any future data into similar clusters. As a data item or pattern is input to the processing (output) layer, the output neuron is considered the winner when it contains the weights most similar to the input. The similarity is then computed by comparing typically the Euclidean distance between the set of weights from each neuron. The shortest or least distance wins. A linear activation function only is employed with no bias. As usual, training of SOM starts by using random values for weights as is the case for any network, provided that no valid “guess” (estimate) for starting values is unknown, which could speed up training. Unlike other neural networks types, the training of SOM normally involves a fixed number of iterations.

3.3 DNN-Pattern Recognition Model

A DNN is a neural network or multilayer perceptron with two or more hidden layers – whose weights are fully connected and initialized using either a supervised or unsupervised pre-training approach [8]. DNN is able to represent high dimensional and correlated features efficiently and can compactly model highly complex mapping functions, which contributes to improving generalization – as weights are trained from all the training data. The training can however be optimized by back-propagating derivatives of the mean square error (MSE) cost functions that measure the discrepancy between the target and actual outputs. Zen, Senior and Schuster [3] however argued that this approach may require large computations than building decision trees, since at prediction stage DNNs require matrix multiplication at each layer, compared to decision trees which only require traversing trees from the root to terminal nodes using a subset of their input features. Hence, the restricted Boltzman machine (RBM) pre-training has been recently explored to minimize this bottleneck to achieve a fast initial reduction in training error. Optimizing DNN training can be achieved using the following steps [7, 25]:

(i) pre-training each layer, exclusively, using a greedy algorithm
(ii) applying unsupervised learning at every layer in a way that preserves information from the input and disentangling any factor of variation
(iii) fine-tuning the entire network, subject to the ultimate criterion of interest

Figure 2 shows an architecture of the proposed DNN, with L layers (L > 3). Each layer has connection weights, a bias vector ($\mathbf{b_L}$), and an output vector ($\mathbf{O_L}$). The number of neurons in each layer is denoted by *m*, where, m^1, m^2,…, m^L, are the number of neurons in Hidden Layers 1, 2, …, L, respectively. The input to the system, x_i are the various criteria for evaluation. In this paper, we have identified the following factors:

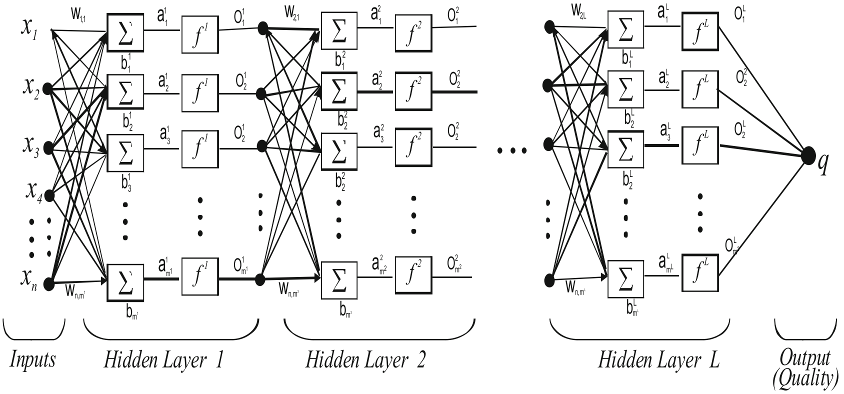


Fig. 2. System architecture of the proposed DNN

naturalness, intelligibility, comprehensibility, tone correctness, vowel correctness and consonant correctness, for use in the prototype evaluation. In each of the layers, the input vector elements enter the DNN through weights, (w_{i,m_n^L}), which represents the weight of the link between the *ith* input neuron and the nth neuron of the *Lth* hidden layer.

The matrix of weights (W) for each layer is obtained in Eq. (1):

$$\mathrm{W} = \begin{bmatrix} w_{1,m_1^L} & w_{1,m_i^L} & w_{1,m_3^L} & \cdots & w_{1,m_n^L} \\ w_{2,m_1^L} & w_{2,m_2^L} & w_{2,m_3^L} & \cdots & w_{2,m_n^L} \\ w_{3,m_1^L} & w_{3,m_2^L} & w_{3,m_3^L} & \cdots & w_{3,m_n^L} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ w_{n,m_1^L} & w_{n,m_2^L} & w_{n,m_3^L} & \cdots & w_{n,m_n^L} \end{bmatrix} \tag{1}$$

The inputs to each layer of the DNN are denoted as, a, with a^1, a^2,…, a^{L}, representing inputs to layers 1, 2, …, L, respectively. Therefore the inputs to the DNN are as follows:

$$\begin{aligned} a^1 &= \sum_{i=1}^{n} \mathrm{f}^1(\mathrm{w}_{1,\mathrm{m}^1}\mathrm{x}_\mathrm{i} + \mathrm{b}^1_{\mathrm{m}^1}) \\ a^2 &= \sum_{i=1}^{n} \mathrm{f}^2(\mathrm{w}_{2,\mathrm{m}^2}\mathrm{a}^1 + \mathrm{b}^2_{\mathrm{m}^2}) \\ a^{\mathrm{L}} &= \sum_{i=1}^{n} \mathrm{f}^{\mathrm{L}}(\mathrm{w}_{\mathrm{L},\mathrm{m}^\mathrm{L}}\mathrm{a}^{\mathrm{L}-1} + \mathrm{b}^{\mathrm{L}}_{\mathrm{m}^\mathrm{L}}) \end{aligned} \tag{2}$$

where, i = 1, 2…, n are the number of input variables (length of the input vector), *m*, is the number of neurons in *Lth* layer (L > 3), while f^L, is the transfer function of the *Lth* Layer. The output of a proceeding layer is the input of the immediate succeeding layer.

Next, a resilient back-propagation pattern recognition algorithm (RPROP), with sigmoid hidden and softmax output neurons (*patternnet*), was used to classify the input vectors. RPROP stands for *resilient propagation* and is an efficient learning scheme that performs a direct adaptation of the weight step based on local gradient information. One of the main advantages of PROP lies in the fact that for many problems no choice of parameters is needed at all to obtain optimal or at least nearly optimal convergence times. The pseudo-code in Fig. 3 shows the kernels of the RPROP adaptation and learning process. The min(max) operator is exposed to deliver the min(max) of two numbers, and the sign operator returns +1, if argument is positive, −1, if negative, and 0, otherwise.

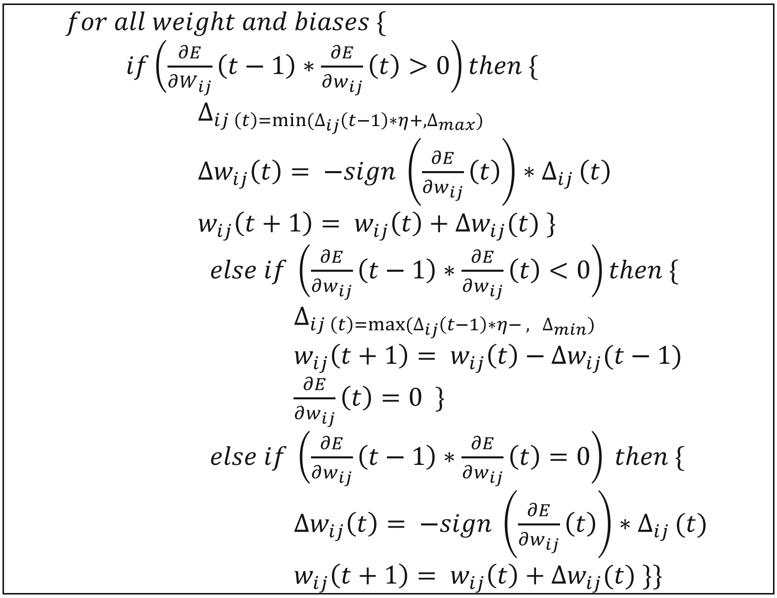
The steps used to implement the DNN-Pattern recognizer include: (i) Initialize the hidden layer size; (ii) Create the pattern recognition network; (iii) set up the division of data for training, validation, and testing; (iv) train the network; (v) test the network; (vi) view the network and display performance plots.

4 Results

4.1 Feature Dimension Reduction and Loading

It is important to assess the contributions of each of the extracted features, to speech quality. The assessment is to enhance the prediction and evaluation of the speech quality. One major objective of data mining is classification learning, and this is achieved by representing relevant data with the smallest number of feature dimensions – such that its characteristics are not lost while reducing the processing complexities in the data [27]. In addition, the accuracy and reliability of a classification or evaluation results will degrade if highly uncorrelated features of interest are used. The process is also necessary to prepare the dataset for supervised learning. The Principal Component Analysis (PCA) is one of the notable and widely used techniques for dimension reduction. In this paper, feature dimensionality assessment was performed to obtain a set of degrees of freedom with which a large proportion of the variability of a speech quality could be explained. Specifically, the speech corpus datasets of size 6×563 was mapped to the given k-principal component framework and transformed into dataset of size $563 \times k$, where k is the number of extracted features with eigenvalues of at least one (1).

The PCA analysis was implemented in Matlab 2015a, and the result is presented in Table 3. Results reveal that naturalness is the most important principal component accounting for 23.92% of the variability of the target feature; followed by intelligibility (20.33%); comprehensibility (19.03%); and tone (13.51%). The first 4 Principal Components are effective for classification since they account for 76.79% of the total variation associated with all 6 original features. This suggests that most of the variability in the



```
$for\ all\ weight\ and\ biases\ \{$
    $if\ \left(\frac{\partial E}{\partial W_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0\right) then\ \{$
        $\Delta_{ij}\ (t)=\min(\Delta_{ij}(t-1)*\eta+,\Delta_{max})$
        $\Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}\ (t)$
        $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)\ \}$
      $else\ if\ \left(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0\right) then\ \{$
            $\Delta_{ij}\ (t)=\max(\Delta_{ij}(t-1)*\eta-,\ \Delta_{min})$
            $w_{ij}(t+1) = w_{ij}(t) - \Delta w_{ij}(t-1)$
            $\frac{\partial E}{\partial w_{ij}}(t) = 0\ \}$
      $else\ if\ \left(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0\right)\ then\ \{$
            $\Delta w_{ij}(t) = -sign\left(\frac{\partial E}{\partial w_{ij}}(t)\right) * \Delta_{ij}\ (t)$
            $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)\ \}\}$
```

Fig. 3. The RPROP algorithm Source [26]

dataset can be obtained with 66.67% of the original dimension in the dataset. The other SQ features – vowel and consonants, together account for 23.21% of the variance, each with eigen value less than 1. In term of communality loadings, the principal components show higher loadings than the non-principal components, and imply that they are suitable for classification of the target feature. Correlation matrix of the principal feature components are given in Table 4. The positive correlations between these features confirm the effectiveness of classification with these features.

Table 3. Principal components and variability proportion

Feature	Eigen value	Proportion (%)	Cumulative (%)	Communality estimates	
				Prior	Final
Naturalness	1.315403	23.92	23.92	0.06384	0.15768
Intelligibility	1.099593	20.33	44.25	0.04210	0.11628
Comprehensibility	1.021843	19.03	63.28	0.04253	0.11656
Tone	1.000558	13.51	76.79	0.01252	0.05564
Consonant	0.859490	12.32	89.11	0.00855	0.04477
Vowel	0.773114	10.89	100.00	0.00197	0.01982

Table 4. Correlation matrix of the principal feature components

	Naturalness	Comprehensibility	Intelligibility	Tone
Naturalness	**1.0000**	0.1955	0.2334	0.0400
Comprehensibility	0.1955	**1.0000**	0.1006	0.0022
Intelligibility	0.2334	0.1006	**1.0000**	0.0529
Tone	0.0400	0.0022	0.0529	**1.0000**

4.2 SOM Visualization

Our SOM implemented the batch learning. The learning rate was fixed to 1, and the neighbourhood function was 1, for all neighbourhood nodes, and 0, elsewhere. This implies that after each epoch, a node's position is averaged over the data points for which the node or one of its neighbours was the winner (the nearest node). The size of the neighbourhood decreases linearly from the specified initial value to 1, in a specified number of epochs (called the ordering phase). Afterwards the neighbourhood size was fixed to 1, for the remaining training epochs (called the tuning phase).

A calibration of the cluster map is shown in Fig. 4. The calibration algorithm determines the degree of convergence of the input classes required to setting a basis for the neuron probabilities for the discovery of the target clusters, and the average variance of clustering provides a measure of the distance between the SOM neuron cluster centers. The calibration considers that each data point is valid with some probability, and could belong to any of the clusters of the SOM. The calibration step connects the clustering and classification steps in a highly logical manner and the procedure is performed for all the neurons regardless of the size and topology of the network. As seen in Fig. 4, the resultant calibration confirms the presence of five pattern classes, in the prediction of the target features. The calibration shows the probability density of feature instances of the dataset in the discovered SQ clusters.

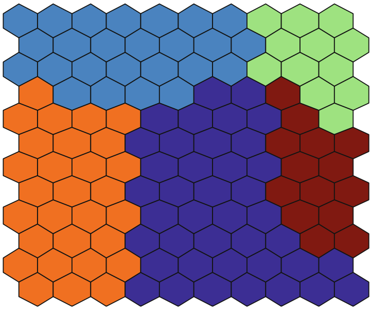


Fig. 4. Calibration plot of discovered target classes

In Fig. 5, the U-matrix: unified distance matrix (top leftmost map), is used to visualize the distances between the neurons. The distance between the adjacent neurons is computed and represented with different colourings between the adjacent nodes. Dark colours represent cluster separators while light colours represent the clusters. From the figure, presence of the clusters is noticed. The next four maps are the SOM component planes (of evaluation features) abstracted from PCA. We observe that naturalness and intelligibility followed same pattern, thus signifying modest correlation between them. In practice, intelligibility and naturalness are required to determine the speech quality, and one cannot preclude the other. Both require the evaluation of some segments of the utterance. While naturalness requires the evaluation of the entire utterance, intelligibility evaluates word segments for vowel, consonant or tone correctness. Also, comprehensibility and tone have similar patterns, but with a weaker correlation – most likely due to the small data (corpus) and uneven tone distribution. In practice, tone is a useful

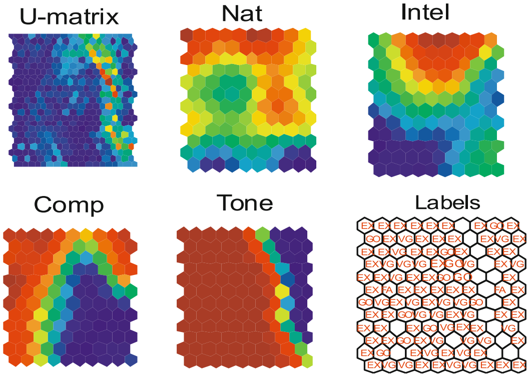
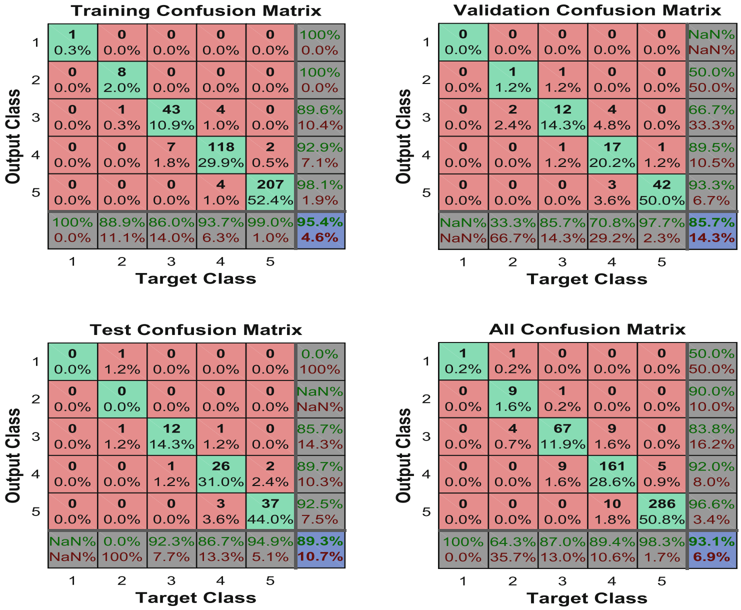


Fig. 5. U-matrix and feature component planes for selected features

determinant of speech comprehensibility. An extensive investigation into these claims with a larger corpus is expected in a future paper.

To ensure efficient knowledge synthesis and discovery, a confusion matrix which gives the classification of SQ patterns of the five distinct target categories (5-excellent, 4-very good, 3-good, 2-fair and 1-poor) is presented in Fig. 6. Samples belonging to “excellent” class were correctly classified. The classification accuracy of class “very good” was 90% (9) and 1 sample was incorrectly classified as a member of class “good’. Out of 78 (13.4% of the samples) instances of the target feature belonging to good SQ class, 67 samples were correctly classified while 4 and 9 samples were incorrectly classified as “very good” and “fair” SQ, respectively. The ‘fair’ and ‘good’ SQ dominated the speech samples used for this experiment and had the highest classification accuracy. While 161 (28.6%) samples of ‘fair’ class were correctly classified, 1.6% (9 samples) and 0.9% (5 samples) were incorrectly classified as members of “good” and “poor” classes respectively. Ten (10) samples of “poor” SQ were misclassified into “fair” quality with 286 (50.8%) samples correctly classified. Patterns of “excellent” SQ had 100% correct classification while samples belonging to ‘poor” class, were only confused with “fair” class. “very good” class had the highest percentage of incorrectly classified patterns. The overall accuracy of classification is 93.1%, signifying an efficient speech quality classifier (Fig. 6).


Training Confusion Matrix

Output Class	1	2	3	4	5	
1	1 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	8 2.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	1 0.3%	43 10.9%	4 1.0%	0 0.0%	89.6% 10.4%
4	0 0.0%	0 0.0%	7 1.8%	118 29.9%	2 0.5%	92.9% 7.1%
5	0 0.0%	0 0.0%	0 0.0%	4 1.0%	207 52.4%	98.1% 1.9%
	100% 0.0%	88.9% 11.1%	86.0% 14.0%	93.7% 6.3%	99.0% 1.0%	95.4% 4.6%

Target Class

Validation Confusion Matrix

Output Class	1	2	3	4	5	
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	0 0.0%	1 1.2%	1 1.2%	0 0.0%	0 0.0%	50.0% 50.0%
3	0 0.0%	2 2.4%	12 14.3%	4 4.8%	0 0.0%	66.7% 33.3%
4	0 0.0%	0 0.0%	1 1.2%	17 20.2%	1 1.2%	89.5% 10.5%
5	0 0.0%	0 0.0%	0 0.0%	3 3.6%	42 50.0%	93.3% 6.7%
	NaN% NaN%	33.3% 66.7%	85.7% 14.3%	70.8% 29.2%	97.7% 2.3%	85.7% 14.3%

Target Class

Test Confusion Matrix

Output Class	1	2	3	4	5	
1	0 0.0%	1 1.2%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	1 1.2%	12 14.3%	1 1.2%	0 0.0%	85.7% 14.3%
4	0 0.0%	0 0.0%	1 1.2%	26 31.0%	2 2.4%	89.7% 10.3%
5	0 0.0%	0 0.0%	0 0.0%	3 3.6%	37 44.0%	92.5% 7.5%
	NaN% NaN%	0.0% 100%	92.3% 7.7%	86.7% 13.3%	94.9% 5.1%	89.3% 10.7%

Target Class

All Confusion Matrix

Output Class	1	2	3	4	5	
1	1 0.2%	1 0.2%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
2	0 0.0%	9 1.6%	1 0.2%	0 0.0%	0 0.0%	90.0% 10.0%
3	0 0.0%	4 0.7%	67 11.9%	9 1.6%	0 0.0%	83.8% 16.2%
4	0 0.0%	0 0.0%	9 1.6%	161 28.6%	5 0.9%	92.0% 8.0%
5	0 0.0%	0 0.0%	0 0.0%	10 1.8%	286 50.8%	96.6% 3.4%
	100% 0.0%	64.3% 35.7%	87.0% 13.0%	89.4% 10.6%	98.3% 1.7%	93.1% 6.9%

Target Class

Fig. 6. Confusion matrix

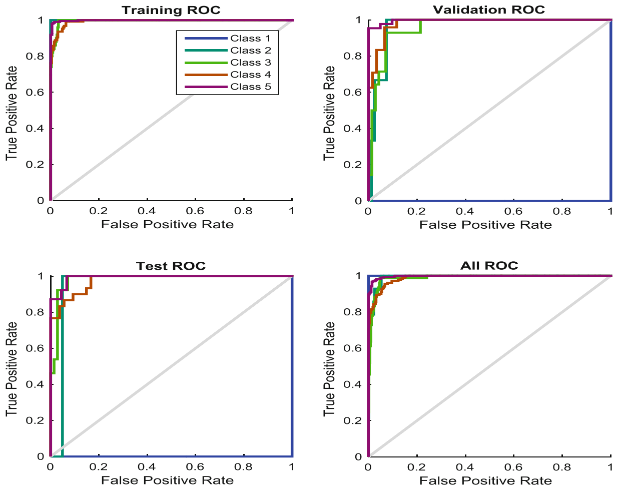


Fig. 7. ROC curve for training, validation and test data

5 Conclusion

Speech quality evaluation procedures for synthesized speech have played significant role in the development and enhancement cycle of synthesis systems. Although a host of factors can influence the speech quality, these factors depend (to a great extent) on the deployed application and can impair listening. Relevant feature identification has therefore become an essential task to apply data mining algorithms effectively in this context. This paper demonstrated the effectiveness of SOM and DNN in the evaluation of speech synthesis systems. The experiment conducted utilized speech quality data obtained from a collection of synthesized voices in a previous experiment [9]. A dimension reduction technique (PCA) was first employed to assess the relevance of each input feature, and select the relevant feature or feature subsets. Using SOM, the selected features were subjected to an unsupervised calibration and visualization – to cluster the target classes, and study the degree of association amongst these features. The SOM enabled the visualization of the selected dataset and evaluation of the feature dimensions of the overall speech quality. To ensure efficient knowledge discovery, DNN-Pattern recognition was exploited evaluated the performance of our classifier. An overall accuracy of 93.1% was achieved from a confusion matrix, signifying an efficient speech quality classifier.

References

1. Yoshimura, T., Tokuda, K., Masuko, T., Kobayashi, T., Kitamura, T.: Simultaneous modeling of spectrum, pitch and duration in HMM based speech synthesis. In: Proceedings of EUROSPEECH Conference (1999)
2. Zen, H., Oura, K., Nose T., Yamagishi, J., Sako, S., Toda, T., Masuko, T., Black, A.W., Tokuda, K.: Recent development of the HMM-based speech synthesis system (HTS). In: Proceedings of APSIPA Annual Summit and Conference, Sapporo, Japan, pp. 121–130 (2009)
3. Zen, H., Senior, A., Schuster, M.: Statistical parametric speech synthesis using deep neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7962–7966 (2013)
4. Hunt, A., Black, A.: Unit selection in a concatenative speech synthesis system using a large speech database. In: Proceedings of ICASSP, Atlanta, Georgia, vol. 1, pp. 373–376 (1996)
5. Savargiv, M., Bastanfard, A.: Study on unit-selection and statistical parametric speech synthesis techniques. J. Comput. Robot. **2**(7–1), 19–25 (2014)
6. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
7. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
8. Deng, L., Dong, Y.: Deep Learning: Methods and Applications. Microsoft Research/NOW Publishers, UK (2014)
9. Ekpenyong, M.E., Inyang, U.G., Ekong, V.E.: A DNN framework for robust speech synthesis systems evaluation. In: Zygmunt, V., Mariani, H. (eds.) Proceedings of 7th Language and Technology Conference (LTC), Poznan, Poland, pp. 256261. Fundacja Uniwersytetu im. A. Mickiewicza (2015)
10. Cambell, N.: Evaluation of speech synthesis. In: Dybkjaer, L., Hamsen, H., Minker, W. (eds.) Evaluation of Text and Speech Systems. Text, Speech and Language Technology, vol. 37, pp. 29–64. Springer, The Netherlands (2007). https://doi.org/10.1007/978-1-4020-5817-2_2
11. Francis, A.L., Nusbaum, H.C.: Evaluating the quality of synthetic speech. In: Gardner-Bonneau, D. (ed.) Human Factors and Voice Interactive systems, pp. 63–97. Kluwer Academic, Boston (1999)
12. Morton, K.: Expectations for assessment techniques applied to speech synthesis. Proc. Inst. Acoust. **13**(2), 1–10 (1991)
13. Klatt, D.: Review of text-to-speech conversion for English. J. Acoust. Soc. Am. JASA **82**(3), 737–793 (1987)
14. Mariniak, A.: Global framework for the assessment of synthetic speech without subjects. In: Proceedings of Eurospeech, vol. 93, no. 3, pp. 1683–1686 (1993)
15. Goldstein, M.: Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener. Speech Commun. **16**, 225–244 (1995)
16. Logan, J., Greene, B., Pisoni, D.: Segmental intelligibility of synthetic speech produced by rule. J. Acoust. Soc. Am. JASA. **86**(2), 566–581 (1989)
17. Pisoni, D., Hunnicutt, S.: Perceptual evaluation of MITalk: the MIT unrestricted text-to-speech system. In: Proceedings of ICASSP, vol. 80, no. 3, pp. 572–575 (1980)
18. Bernstein, J., Pisoni, D.: Unlimited text-to-speech system: description and evaluation of a self organized maps. In: International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), pp. 215–222 (1980)

19. Duffy, S.A., Pisoni, D.B.: Comprehension of synthetic speech produced by rule: a review and theoretical interpretation. Lang. Speech **35**, 351–389 (1992)
20. Kraft, V., Portele, T.: Quality evaluation of five German speech synthesis systems. Acta Acust. **3**(1995), 351–365 (1995)
21. Pavlovic, C., Rossi, M., Espesser, R.: Use of the magnitude estimation technique for assessing the performance of text-to-speech synthesis system. J. Acoust. Soc. Am. JASA **87**(1), 373–382 (1990)
22. Mannell, R.: Evaluation of speech synthesis systems. Macquarie University, Australia (2009). http://clas.mq.edu.au/speech/synthesis/synth_evaluation/. Accessed 26 June 2017
23. Clark, R.A., Dusterhoff, K.E.: Objective methods for evaluating synthetic intonation. In Proceedings of Eurospeech, vol. 4, pp. 1623–1626 (1999)
24. Kohonen, T.: Essential of self organizing maps. Neural Netw. **37**, 52–65 (2013)
25. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
26. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: the RPROP algorithm. In: Proceedings of IEEE International Conference on Neural Networks, San Francisco, CA, USA, pp. 586–591 (1993)
27. Vasan, K., Surendiran, B.: Dimensionality reduction using Principal Component Analysis for network intrusion detection. Perspect. Sci. **8**, 510–512 (2016)

Neural Networks Revisited for Proper Name Retrieval from Diachronic Documents

Irina Illina[1,2,3](✉) and Dominique Fohr[1,2,3]

[1] Université de Lorraine, LORIA, UMR 7503, 54506 Vandoeuvre-lès-Nancy, France
irina.illina@loria.fr
[2] Inria, 54600 Villers-lès-Nancy, France
[3] CNRS, LORIA, UMR 7503, 54506 Vandoeuvre-lès-Nancy, France

Abstract. Developing high-quality transcription systems for very large vocabulary corpora is a challenging task. Proper names are usually key to understanding the information contained in a document. To increase the vocabulary coverage, a huge amount of text data should be used. In this paper, we extend the previously proposed neural networks for word embedding models: word vector representation proposed by Mikolov is enriched by an additional non-linear transformation. This model allows to better take into account lexical and semantic word relationships. In the context of broadcast news transcription and in terms of recall, experimental results show a good ability of the proposed model to select new relevant proper names.

Keywords: Speech recognition · Neural networks · Vocabulary extension
Out-of-vocabulary words · Proper names

1 Introduction

In the context of Large-Vocabulary Continuous Speech Recognition (LVCSR) systems, accurate recognition of proper names (PNs) is important because proper names are essential for understanding the content of speech (for example, for voice search, spoken dialog systems, broadcast news transcription, etc.). No vocabulary can contain all existing PNs [7]. By searching new proper names and by adding them to the standard vocabulary of LVCSR, we want to face the problem of out-of-vocabulary words (OOV, words that are not in LVCSR system vocabulary).

In word similarity or analogy tasks, count-based distribution models [1, 3, 22] and word embedding models have been successfully used [2, 4]. These approaches are based on the idea that words in similar contexts have similar meanings.

Recently, several new word embedding approaches have been proposed and have given very good performance. Mikolov et al. ([16–18]) have proposed continuous word representations in vector space based on Neural Networks (NN): semantic and syntactic word relationships are taken into account using huge amounts of unstructured text data. Another popular approach is GloVe (Global Vectors) that is based on a log-bilinear regression and tries to keep meaningful structure of the word space [19]. Hyperparameter optimization is a crucial factor for performance gain for embedding systems [14].

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 19–30, 2018.
https://doi.org/10.1007/978-3-319-93782-3_2

Compared to other methods, Mikolov's word embedding model gives very good accuracy on different tasks while minimizing computational complexity [14]. Today, Mikolov's system represents a state-of-the-art framework.

In [6], Mikolov's word embedding methods have been proposed for increasing the vocabulary of the ASR system with new PNs. The system uses lexical and temporal features. PNs evolve through time. For a given date, the same PNs would occur in documents that belong to the same time period [5]. In the present paper, the same problem of PN retrieval using lexical and temporal context is considered. We extend this work and propose to better model the word dependencies in Mikolov's word embedding model.

The scientific contributions of this paper are:

- We extend Mikolov's neural network by adding an extra non-linear transformation and we study different word projections;
- We present a comparison of standard (Mikolov) and our proposed approach in the context of French broadcast news speech transcription.

The paper is organized as follows. Section 2 introduces the proposed approach. Sects. 3 and 4 describe the experimental sets and the results of the evaluation of this model.

2 Proposed Methodology

In this paper, we use the same general framework as in [6, 9]: we want to use the relationships between co-occurring PNs for better OOV PN retrieval. For this, we want to take into account temporal, lexical and semantic context of PNs. We use text documents from a diachronic corpus that are contemporaneous with the test documents to be transcribed. We assume that, for a certain date, a PN from the test corpus will co-occur with other PNs in diachronic documents of the same time period [12]. Consequently, we have a test audio document to transcribe which contains OOV words, and we have a diachronic text corpus used to retrieve OOV proper names. An augmented vocabulary is dynamically built for each test document to avoid an excessive increase of vocabulary size.

We chose the high-quality vector representation of words proposed by Mikolov et al. [17] for OOV PN retrieval. This approach allows to build semantic context dependencies of OOV PNs.

2.1 OOV Retrieval Method

Our OOV retrieval method consists of 5 steps as in [6]:

(A) In-vocabulary (IV) PN extraction from each test document: For each test document, we extract IV PNs from the automatic transcription performed using our standard vocabulary. The goal is to use these PNs as anchors to collect linked new proper names.

(B) Selection of diachronic documents and extraction of new PNs from them: only diachronic documents (DDs) that correspond to the same time period as the test document are considered. After POS-tagging of these DDs, meaningful words are kept: verbs, adjectives, nouns and PNs. Among these PNs, we create a list of those that do not belong to our standard vocabulary (OOV PN).

(C) Temporal and lexical context extraction from diachronic documents (DD): After extracting the list of the IV PNs from the test document (step A), and the list of the OOV PNs from DDs (step B), we build their temporal and lexical contexts. For this, a high-dimensionality word representation space is used (see description below). We hope that in this space semantically and lexically related words will be in the same region of the space.

(D) Ranking of new PNs: The cosine-similarity metric is calculated between the projected vector of IV PNs found in the test document and the projected vector of each OOV PN occurring in the selected diachronic documents.

(E) Vocabulary augmentation: To reduce the vocabulary growth, only the top-N OOV PNs according to the cosine-similarity metric are added to our vocabulary. OOV PN pronunciations are generated using a phonetic dictionary or an automatic phonetic transcription tool.

Using this methodology, we expect to extract a reduced list of the potentially missing PNs.

Figure 1 presents an example of the cosine-similarity computation for one OOV PN (step D).

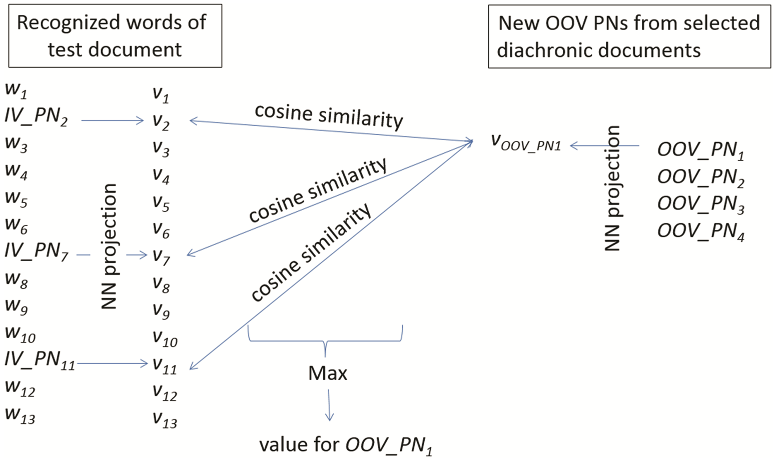


Fig. 1. Example of computation of the cosine-similarity metric of one OOV_PN.

2.2 Neural Networks for Word Representation

We propose to model the word space (step C) using Mikolov's neural network. In this network, each word is represented by a continuous vector in a high-dimensionality space.

We hope that this space will take into account semantic and lexical relationships between words.

We use Mikolov's Skip-gram model that tries to predict surrounding words of one input word. This is performed by maximizing the classification rate of nearby words given the input word. More formally, given a sequence of training words w_1, $w_2 \ldots$, w_T, it maximizes the average log probability:

$$\frac{1}{T}\sum\nolimits_{t=1}^{T}\sum\nolimits_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

where c is the context size and T the number of training words. Compared to classical NN, the non-linear hidden layer is removed and the projection layer is shared for all words. This model assumes that semantically similar words will be projected in the same region of the semantic space. An important property of this model is that the word representations learned by the Skip-gram model exhibit a linear structure.

Standard Case (Mikolov's Model). Let $X = \{x_1, \ldots, x_v\}$ denotes the input vector of the neural network. For a given input word, only one element of X will be 1, all other elements will be 0. V is the vocabulary size and N is the size of the hidden layer. W_{ih} is the VxN matrix of weights between the input layer and the hidden layer of the network. W_{ho} is the NxV matrix of weights between the hidden layer and the output layer. The goal of the neural network training is to estimate these two matrices.

Given a word and assuming that $x_k = 1$ and $x_j = 0$ for $j \neq k$, we have:

$$h = X^T W_{ih} \tag{2}$$

Thus, the hidden layer is obtained by a linear transformation of the input vector.

By deriving the error function on the output of the hidden layer, we obtain the updating equation for the input to hidden weights

$$\frac{\partial E}{\partial h} = \sum\nolimits_{j=1}^{L(w)-1}\left(\sigma\left(v_j'^{T} h\right) - t_j\right) \bullet v_j' \tag{3}$$

where σ is a sigmoid function, t_j is 1 if the j-th node is the actual output word and v_j' is the j-th column of the weight matrix W_{ho}. $L(w)$ is the length of the path from the leaf w to the root in the tree for the hierarchical softmax.

For the case of updating the hidden to output weights, we obtain:

$$\frac{\partial E}{\partial v_j' h} = \sigma\left(v_j'^{T} h\right) - t_j \tag{4}$$

Figure 2 shows a diagram of Mikolov's model on the standard case.

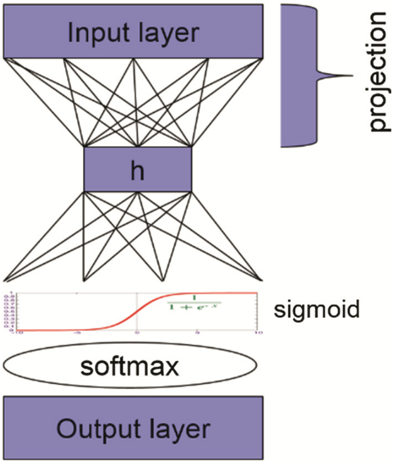


Fig. 2. Mikolov's skip-gram neural network (standard case).

Modified Case

In the framework of neural networks, the sigmoid function allows to distort the representation space, thus NNs can learn powerful non-linear transformations: in fact, with enough hidden units they can represent arbitrarily complex functions. “By transforming the data non-linearly into a new space, a classification problem that was not linearly separable (not solvable by a linear classifier) can become separable.” [2].

In this article, we propose to reinforce the non-linearity of the standard Mikolov model. For this we add a sigmoid transformation between input layer and hidden layer (as in a classical MLP).

We chose the sigmoid function because it is a classical non-linear function used in neural network frameworks and its derivative is easy to compute:

$$(x) = \frac{1}{1 + \exp(-x)} \tag{5}$$

We add a non-linear transformation (sigmoid) to compute the hidden layer: the Eq. (2) is replaced by Eq. (6).

$$h = \sigma(X^T W_{ih}) \tag{6}$$

We denote this case as modified: the hidden layer is obtained by a non-linear transformation of the input vector. As the model is more complex than the standard one, we hope that the new network is able to better model the relationships between words and thus the vector representation of words will be more accurate.

Figure 3 presents the proposed architecture. The sigmoid function is used to compute the hidden layer.

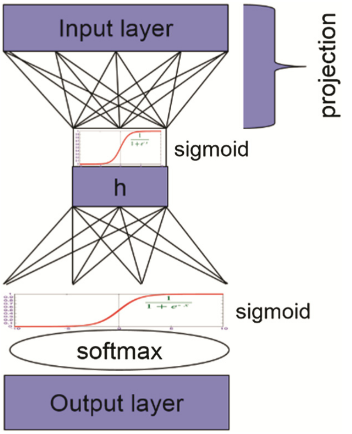


Fig. 3. Mikolov's skip-gram neural network (modified case).

In the standard case (Fig. 2), at the end of the training, the hidden layer computed for a word will be used as projection (vector representation) for this word. In the modified case, we do not use the hidden layer as projection. Instead, we use the values before applying the sigmoid ($X^T W_{ih}$). So, the vector representation space keeps the linear structure as in the standard case (Mikolov's model). For this non-linear transformation, the derivatives of weights for the input to hidden layer are modified (Eq. (3)):

$$\frac{\partial E}{\partial h} = \sum_{j=1}^{L(w)-1} \left(\sigma\left(v_j'^{T} h\right) - t_j\right) \bullet v_j' \bullet h \bullet (1-h) \tag{7}$$

3 Experiments

In this article, ***selected PNs*** are the new proper names that we were able to find in DDs using our methods. ***Retrieved OOV PNs*** are the *selected PNs* that are present in the test documents. Using the DDs, we build a specific augmented lexicon for each test document according to the chosen period. Figure 4 presents the OOV retrieval method architecture.

Results are presented in terms of Recall (%): the number of *retrieved OOV PNs* versus the number of OOV PNs. For the recognition experiments, PN Error Rate (PNER) is given. PNER is calculated like WER but taking into account only proper names. The best results are highlighted in bold in Tables.

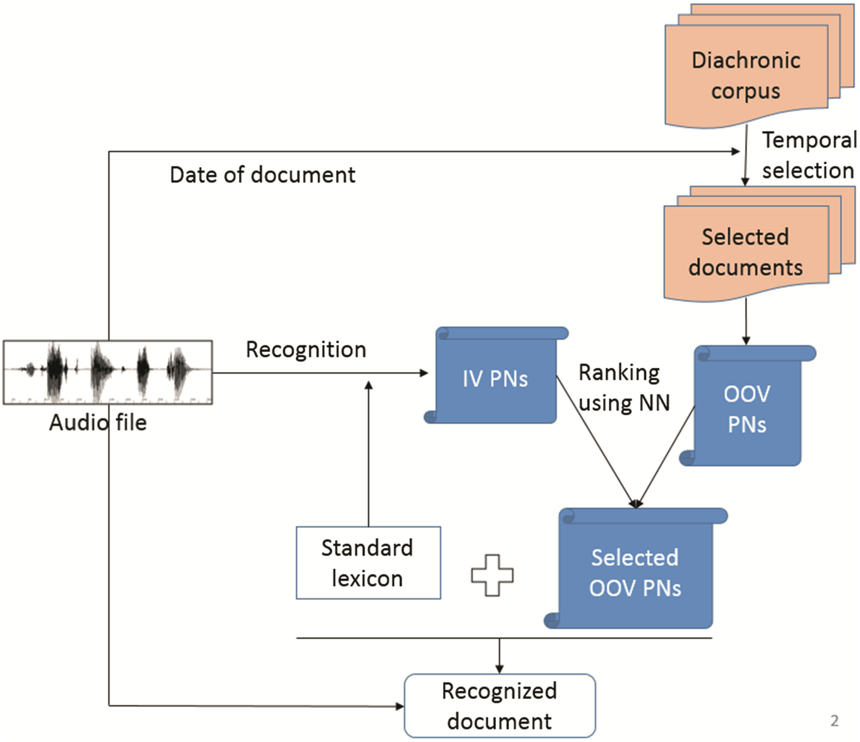


Fig. 4. OOV retrieval method architecture.

3.1 Audio Corpus

As audio corpus, seven audio documents of development part of ESTER2 (between 07/07/2007 and 07/23/2007) are used [8]. To artificially increase OOV rate, we have randomly removed 223 PNs occurring in the audio documents from our 122k ASR vocabulary.

Table 1 summarizes the average occurrences of all PNs (IV and OOV) in audio documents with respect to 122k-word ASR vocabulary. Finally, the OOV PN rate is about 1.2%.

Table 1. Average proper name coverage for audio corpus per file.

Corpus	Word occ	IV PNs	IV PN #occ	OOV PNs	OOV PN #occ
Audio	4525.9	99.1	164.0	30.7	57.3

3.2 Diachronic Corpus

The French *GigaWord* corpus is used as the diachronic corpus: newswire text data from *Agence France Presse* (AFP) and *Associated Press Worldstream* (APW) from 1994 to 2008. The choice of *GigaWord* and ESTER corpora was driven by the fact that one is contemporary with the other, their temporal granularity is one day and they have the same textual genre (journalistic) and domain (politics, sports, etc.).

3.3 Transcription System

ANTS [11] is based on Context Dependent HMM phone models trained on 200-hour broadcast news audio files. The recognition engine is Julius [13]. The baseline phonetic lexicon contains 260k pronunciations for the 122k words. Using the SRILM toolkit [21], the language model is estimated on text corpora of about 1800 million words. The language model is re-estimated for each augmented vocabulary using the whole text corpora. The best way to incorporate the new PNs in the language model is beyond the scope of this paper.

4 Experimental Results

4.1 Baseline Results

We extracted a list of all the OOV PNs occurring in the selected diachronic documents corresponding to the time period of the document to be transcribed. This period can be, for example, a day, a week or a month. After this, our vocabulary is augmented with the list of all extracted OOV PNs. If the diachronic corpus is large, a bad tradeoff between the lexical coverage and the increase of the lexicon size is obtained.

Using *TreeTagger* [20], we extracted 160k PNs from 1 year of the diachronic corpus. Among these 160k PNs, 119k are not in our lexicon. Among these 119k, only 151 PNs are present in the audio corpus. It shows that it is necessary to filter this list of PNs to have a better tradeoff between the PN lexical coverage and the increase of the lexicon size.

Table 2 shows that using the DDs of 1 year, we extract, on average, 118797.0 PNs per file. Among these PNs, we retrieve on average 24.0 OOV PNs per audio file (compared to 30.7 in Table 1). This represents a recall of 78.1%.

Table 2. Baseline results for audio corpus according to time periods. Values averaged on the 7 audio files.

Time period	Average of selected PNs per file	Average of retrieved OOV PNs per file	Recall (%)
1 day	532.9	10.0	32.6
1 week	2928.4	11.4	37.2
1 month	13131.0	17.6	57.2
1 year	118797.0	24.0	**78.1**

4.2 NN-Based Results

We used Mikolov's open-source NN toolkit available on the web. The NN is trained on the diachronic corpus described in Sect. 3.2 using only meaningful words. After preliminary experiments, we defined the best parameter set that will be used here: 400 for the size of the hidden layer, 20 for the context size and 5 for the number of negative samples. We performed 5 training epochs. Moreover, for the month time period, in order to select more relevant PNs, a frequency threshold is used (OOV PNs occurring less than 3 times in the selected diachronic documents are excluded). We selected the Skip-gram model because it achieved very good results on semantic tasks [14]. The target word is at the input layer and the context words are at the output layer.

An operating point of 15% of the average number of selected PNs per audio file seems to be a good compromise: 80 for a day, 440 for a week and 2000 for a month [6]. This operating point is chosen to obtain a good recall with a reasonable number of selected PNs. This operating point will be used in our experiments.

First, we analyzed the effect of hierarchical softmax and negative sampling in the standard and modified cases. Table 3 shows the recall according to time periods. From the results it can be observed that for the standard case, hierarchical softmax gives better results for all periods (24.2% versus 22.3%, 32.1% versus 28.8% and 47.0% versus 44.2%). For the proposed modified case, negative sampling performs slightly better compared to hierarchical softmax. Comparing standard and modified cases, for a day and a month period, the modified model is more powerful. For a week period this is not true.

Table 3. Recall (%) for standard and modified cases according to time period for audio corpus. Values averaged on the 7 audio files.

Methods	1 day 80 selected PNs		1 week 440 selected PNs		1 month 2000 selected PNs	
	Std.	Modif.	Std.	Modif.	Std.	Modif.
Hierarch. softmax	24.2	25.1	**32.1**	31.6	47.0	47.4
Negative sampling	22.3	**25.6**	28.8	31.2	44.2	**48.8**

At the end of the training, we obtain two weight matrices: word matrix W_{ih} and context matrix W_{ho}, according to the notations of Levy et al. [15]. In Table 4, results are given for different time periods and different ways to calculate the projections using these two matrices. We want to analyze the importance of different projections. Negative sampling is only used because it obtains good results for modified cases according to Table 3.

- ***ih only***: this method refers to Mikolov's word projection: for the word k it is the k-th row of the matrix W_{ih};
- ***ho only***: the projection of the word k is the k-th column of the matrix W_{ho};
- ***Concat***: the projection of the word k is the concatenation of the k-th row of the matrix W_{ih} and the k-th column of the matrix W_{ho};
- ***Sum***: in the standard and modified case, the representation space has a linear property: word vectors can be combined using vector addition. In this case, the projection of

word k is the sum of the k-th row of the matrix W_{ih} and the k-th column of the matrix W_{ho}. Levy et al. [14] have shown that using this kind of addition is equivalent to using first and second order similarities: "The second order similarity measures the extent to which two words are replaceable based to their tendencies to appear in similar contexts."

Table 4. Recall (%) for standard case according to time period for audio corpus. Negative sampling. Values averaged on the 7 audio files.

Methods	1 day 80 selected PNs	1 week 440 selected PNs	1 month 2000 selected PNs
ih only	**25.6**	31.2	**48.8**
ho only	17.7	21.9	38.6
Concat (ih, ho)	25.1	31.6	**48.8**
Sum (ih + ho)	24.2	**32.6**	47.4

As shown in Table 4, these different projections only give tiny improvements over Mikolov's case or a degradation (*ho only*). *ho only* configuration uses only the context matrix and so the word representation is less accurate, as expected. Sum configuration seems to be neither good nor bad for our task and our corpus.

It confirms the results of Levy et al. [14]: on eight datasets, results improved in four cases and degraded in other four cases. So this behavior is perhaps corpus dependent or task dependent.

Table 5, extracted from Table 3, summarizes the best results for standard and modified cases. It can be seen that the recall improvement is obtained with modified cases for day and month periods. But, we notice that for a week period, the degradation is about 0.9%.

Table 5. Recall (%) for standard and modified cases according to time period for audio corpus. Hierarchical softmax (HS) for standard cases and negative sampling (NS) for modified cases. Values averaged on the 7 audio files.

Time period	Method	Selected PNs	Recall (%)
1 day	Std. HS	80	24.2
	Modif. NS	80	25.6
1 week	Std. HS	440	32.1
	Modif. NS	440	31.2
1 month	Std. HS	2000	47.0
	Modif. NS	2000	**48.8**

4.3 Automatic Speech Recognition Results for the Audio Corpus

We performed automatic transcription of the 7 audio documents using augmented lexicons (generating one lexicon per audio file, per period and per case). For generating the pronunciations of the added PNs, we used the G2P CRF approach [10], trained on phonetic lexicon containing about 12000 PNs.

In order to incorporate the new PNs in the language model, we re-estimated it for each augmented vocabulary using the large text corpus described in Sect. 3.3. The number of selected PNs per period is the same as previously: 80 for a day, 440 for a week and 2000 for a month.

In terms of word error rate, no significant improvement is observed using the augmented lexicon. In this work, we are interested in the proper name recognition, so we also compute proper name error rate. Table 6 shows that, compared to standard lexicon, a significant improvement is obtained for the two NN systems (standard and modified cases) in terms of PN error rate (38.2, 38.3% versus 43.6%). There is no significant PNER difference between standard and modified cases. It is possible that, when using larger dataset, the difference will increase.

Table 6. PNER (%) for standard and modified cases according to time period. Skip-gram model. Hierarchical softmax for standard case and negative sampling for modified case. Values averaged on the 7 audio files.

Stand. lexicon	Augmented lexicon			
43.6	*Method*	*1 day*	*1 week*	*1 month*
		80 selected PNs	*440* selected PNs	*2000* selected PNs
	Std HS	40.9	40.4	38.3
	Modif. NS	40.5	40.4	**38.2**

5 Conclusion

In this paper, the problem of OOV proper names and the vocabulary extension of a speech recognition system were investigated. Diachronic documents were used to retrieve new proper names and to enrich the vocabulary. We are interested in the continuous space word representation using the neural networks proposed by Mikolov. One of the key contributions of this paper is to extend Mikolov's network by adding a non-linear transformation to better model the lexical and semantic context of proper names.

Experimental analysis on French corpus of broadcast news suggests the proposed modified configuration slightly outperforms the standard one in terms of recall. In terms of PNER, the results of both methods are comparable and show a significant decrease of the proper name error rate.

Acknowledgements. This work is funded by the *ContNomina* project supported by the French national Research Agency (ANR) under contract ANR-12-BS02-0009.

References

1. Baroni, M., Lenci, A.: Distributional memory: a general framework for corpus-based semantics. Comput. Linguist. **36**(4), 673–721 (2010)
2. Bengio, Y., Goodfellow, I., Courville, A.: Deep Learning. MIT Press, Cambridge (2015)

3. Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16**(1), 22–29 (1990)
4. Deng, L., et al.: Recent advances in deep learning for speech research at Microsoft. In: Proceedings of ICASSP (2013)
5. Federico, M., Bertoldi, N.: Broadcast news LM adaptation using contemporary texts. In: Proceedings of Interspeech, pp. 239–242 (2001)
6. Fohr, D., Illina, I.: Word space representations and their combination for proper name retrieval from diachronic documents. In: Proceedings of Interspeech (2015)
7. Friburger, N., Maurel, D.: Textual similarity based on proper names. In: Proceedings of the Workshop Mathematical/Formal Methods in Information Retrieval, pp. 155–167 (2002)
8. Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.-F., Gravier, G.: The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. In: Proceedings of Interspeech (2005)
9. Illina, I., Fohr, D., Linares, G.: Proper name retrieval from diachronic documents for automatic transcription using lexical and temporal context. In: Proceedings of SLAM (2014)
10. Illina, I., Fohr, D., Jouvet, D.: Grapheme-to-phoneme conversion using conditional random fields. In: Proceedings of Interspeech (2011)
11. Illina, I., Fohr, D., Mella, O., Cerisara, C.: The automatic news transcription system: ANTS, some real time experiments. In: Proceedings of ICSLP (2004)
12. Kobayashi, A., Onoe, K., Imai, T., Ando, A.: Time dependent language model for broadcast news transcription and its post-correction. In: Proceedings of ICSPL (1998)
13. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine julius. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (2009)
14. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Trans. Assoc. Comput. Linguist. **3**, 211–225 (2015)
15. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, pp. 2177–2185 (2015)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at ICLR (2013)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceedings of NIPS (2013)
18. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of NAACL:HLT (2013)
19. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: Proceedings of EMNLP (2014)
20. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of ICNMLP (1994)
21. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proceedings of ICSLP (2002)
22. Turney, P., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1), 141–188 (2010)

Cross-Lingual Adaptation of Broadcast Transcription System to Polish Language Using Public Data Sources

Jan Nouza, Petr Cerva, and Radek Safarik(✉)

Institute of Information Technology and Electronics,
Technical University of Liberec, Studentska 2, Liberec, Czech Republic
{jan.nouza, petr.cerva, radek.safarik}@tul.cz

Abstract. We present methods and procedures designed for cost-efficient adaptation of an existing speech recognition system to Polish. The system (originally built for Czech language) is adapted using common texts and speech recordings accessible from Polish web-pages. The most critical part, an acoustic model (AM) for Polish, is built in several steps, which include: (a) an initial bootstrapping phase that utilizes existing Czech AM, (b) a lightly-supervised iterative scheme for automatic collection and annotation of Polish speech data, and finally (c) acquisition of a large amount of broadcast data in an unsupervised way. The developed system has been evaluated in the task of automatic content monitoring of major Polish TV and Radio stations. Its transcription accuracy (measured on a set of 4 complete TV news shows with total duration of 105 min) is 79,2%. For clean studio speech, its accuracy gets over 92%.

Keywords: Speech recognition of polish · Broadcast monitoring
Acoustic model training · Cross-lingual adaptation

1 Introduction

Within the last 15 years we have been working on the development of a robust automatic speech recognition (ASR) system for Czech. Its recent version is capable of fairly accurate real-time speech recognition even if the lexicon size exceeds 500,000 words. It has been used in several applications, e.g. voice dictation programs, broadcast monitoring systems, or automatic transcription of a huge historical audio archive [10]. It has been a natural idea to utilize the existing modules and acquired experience to port the system to other Slavic languages.

A few years ago we began to work on Slovak language, which is the most similar one. A prototype was presented in 2008 when it achieved 75% word recognition rate (WRR) on a Slovak broadcast news task. The most recent version operates with WRR value around 86% and it has been already deployed in several practical applications.

The next language we decided to focus on has been Polish. It is partly understandable to Czech people though it has completely different lexicon and phonology. We have developed a set of procedures that allowed us to utilize the existing text and audio processing tools and even the Czech acoustic model to build a Polish ASR

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 31–41, 2018.
https://doi.org/10.1007/978-3-319-93782-3_3

system within a relatively short period of one year. We have saved much human labor by automating the most tedious works, such as speech data collection, phonetic annotation and acoustic model training. Moreover, during the development we have used only data (texts and recordings) that are freely available on the Internet, which also reduced the costs. In this paper, we present our approach and methods in more detail.

2 State-of-the-Art and Related Work

With its approx. 40 million native speakers, Polish is the second largest Slavic language (after Russian). Yet, there is not much literature concerned with Polish speech recognition. Major scientific databases offer a rather small number of research papers published on that topic, including those dealing with small vocabulary tasks, such as [17] or [5]. A large-vocabulary continuous speech recognition (LVCSR) system for Polish is presented in [7]. The author used the open-source HTK toolkit to build an experimental system with a 20 k-word lexicon and tested it on read speech recordings provided by 12 speakers with average WRR around 87%. Another LVCSR system, named Skrybot, is briefly described in [12]. Its decoder is based on open-source Julius system and the authors state that its WRR was 73% in a 5-h test. (No information about the lexicon size and the test is provided.) A more recent approach to the development of a Polish dictation system for legal texts is described in [4].

Polish ASR has been investigated also by research teams from abroad. It is one of the 20 languages whose spoken data have been collected within project called Globalphone [13]. These data were later used to test a method for rapid development of language models (LM) in 5 Slavic languages, including Polish [16]. However, the most critical task in the development of any ASR system is the creation of an acoustic model (AM). Lööf et al. [6] proposed a method for cross-lingual adaptation and unsupervised iterative training of a Polish AM. Their work was part of a project focused on automatic transcription of EU parliament talks. They used recordings of Polish representatives and interpreters together with official text documents to adapt an ASR system originally designed for Spanish to Polish. With a 60 k word lexicon they were able to get close to 82% WRR on that given task.

Our approach described in this paper has a similar idea. We want to utilize the existing Czech ASR as a starting point from which the target Polish system is built in an iterative and almost fully automated way. Our goal is more ambitious because we want to create a system for transcription of TV and radio programs, where many speakers, various speaking styles and different topics can occur. The lexicon must be much larger (at least 250 k words) and also the AM and LM need to be more flexible and robust.

3 Modular ASR System

The LVCSR system we have built for Czech has a modular structure where the language specific modules (lexicon, LM and AM) are separated from the rest of the system (a signal processing front-end and a decoder).

The signal parameterization unit can accept many major audio format (e.g. WAV or MP3), which is internally converted into a stream of 16 kHz, 16 bit sampled data. They are converted into 39 Mel-frequency cepstral coefficients (MFCC) and then floating-window Cepstral Mean Subtraction (CMS) is applied. The acoustic model uses triphone multi-gaussian HMMs to represent all phonemes and 8 types of noise. The most recent version employs also a deep neural network (DNN) with 5 to 7 hidden layers which proved to be more robust especially for lower-quality signals. The decoder is based on highly optimized implementation of Viterbi algorithms. On recent CPUs, most applications can run in real time.

When porting the system to Polish, we need to replace the Czech lexicon, AM and LM by the Polish ones.

4 Lexicon and Language Model for Polish

The first step consists in the acquisition of a sufficiently large text corpus, which is necessary for creating a representative lexicon and an LM.

4.1 Text Corpus

Nowadays, the best source of multi-domain texts are web-pages of major newspapers and broadcasters. We have developed a web parser that can be easily adjusted to any web source type. It is based on an SGML parser that transfers an HTML file into an XML structure, from which we can distill the content we are interested in. In this way we have collected and processed almost 3 GB of texts from major Polish newspapers (*Gazeta Wyborcza, Rzeczpospolita, Dziennik Gazeta Prawna, Fakt*, etc.) and TV/radio stations oriented on news (*TVP, TVN24, TV-Nowa, Polsat*).

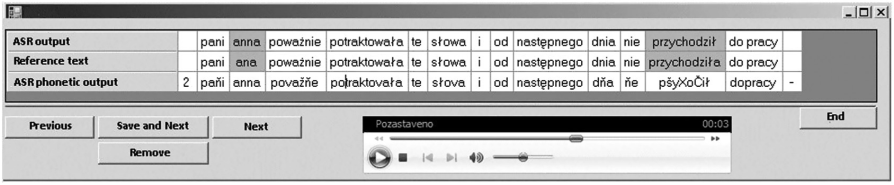
The downloaded data were cleaned and pre-processed. The remaining HTML artifacts as well as non-literal symbols, strings, formatting marks, etc., were removed. Next we tried to replace digits by their text equivalents. This is a challenging tasks in all Slavic languages because a digit (or a string of digits), when spoken, can get various morphological forms that depend on long context. We used the algorithms and tools mentioned in [1], which allowed us to convert at least some types of digit strings, namely those containing dates, years, currencies or distances.

4.2 Lexicon

As Polish is a highly inflected language with many word-forms derived from a lemma, we had to limit the lexicon to the most frequent words. We selected all that were seen in the corpus at least 10 times and got a lexicon with 303 k entries.

4.3 Pronunciation

Polish, similarly to other Slavic languages, has a rather straightforward relation between orthography and pronunciation. We used the basic rules mentioned in [2] to make a grapheme-to-phoneme (G2P) converter. It was applied to all items to get a pronunciation vocabulary needed for an ASR system. A special care was put to abbreviations (where we used a spelled-letter converter), terms with digits (e.g. 'A1') and to loanwords. For the latter, we borrowed their pronunciation from the Czech lexicon. Some words were assigned multiple phonetic variants, e.g. in case of 'NHL', 'ABC' or 'Jacka', where Polish as well as English pronunciation can occur. The recent version of the lexicon has 303,321 entries with 318,888 pronunciations. We use a set of 36 phonemes, each represented by a single-letter symbol (an example is in Fig. 1).



ASR output		pani	anna	poważnie	potraktowała	te	słowa	i	od	następnego	dnia	nie	przychodził	do pracy	
Reference text		pani	ana	poważnie	potraktowała	te	słowa	i	od	następnego	dnia	nie	przychodziła	do pracy	
ASR phonetic output	2	paňi	anna	považňe	potraktovała	te	słova	i	od	następnego	dňa	ňe	pšyXoČił	dopracy	-

Fig. 1. Program to check transcribed sentences. One can easily compare ASR output, reference text and ASR produced phonetic transcription (incl. silence denoted as '-' and noises indicated by digits). Differences are highlighted. In this example, the first one is due to wrongly typed word 'ana' (error in reference), the second was made by the ASR system (omitted 'a').

4.4 Language Model

The LM is probabilistic, based on N-grams. From practical reasons (mainly with respect to the very large vocabulary size), we prefer bigrams. In the 3 GB corpus of Polish texts we found 65 million different word-pairs.

5 Acoustic Model Building - Methods and Procedures

Building a robust AM requires that at least 50 h of speech from hundreds of speakers must be collected. Each recording need to be annotated on the acoustic-phonetic level (as a sequence of phonemes and noises). This is the most tedious and time consuming part of the development. There exist projects that focus on the data collection, e.g. [3]. However, these data are not freely available and we had to search for alternative resources. The most suitable ones seem to be archives of broadcast stations or national parliaments. They contain both audio and text documents that can be used as source data for automatically annotated speech. We created several procedures and schemes for this purpose, that are described in the following text.

5.1 Basic Procedures

The procedures employ an existing ASR system (i.e. a system available at that phase) to do most works that would be otherwise done by a human expert. They cut long audio documents (at proper instants) to get shorter and manageable files, transcribe them on the orthographic and phonetic level and decide which files could be automatically added to a training set and which should be possibly corrected by a human annotator.

Automatic Transcription of Audio Signal. Here we employ the basic operation mode of the ASR decoder. It takes a parameterized signal, decodes it (using the given lexicon, AM and LM) and translates it into text. Our decoder can reveal also detailed acoustic-phonetic transcription of the signal with pronunciation of each recognized word, and detected noises. This type of the output can be utilized for annotations that are necessary for AM training. Moreover, the decoder can provide also start/end times (so called time stamps) for each word and noise. We call the procedure `DoTanscription`.

Automatic Segmentation of Audio Signal. The above mentioned time stamps are useful if we work with long audio documents and need to split them into shorter segments that are better suited for further processing. It is done by procedure `DoSegmentation` that reads the detailed ASR output and cuts the signal at convenient instants - usually during silence, noise or breath, so that the speech itself is not disrupted. We can set the limits for the segment length.

Segmentation Matched to Text. This is the most essential procedure. It is used if we have an audio signal and a text that more or less corresponds to the content of the signal. In the optimal case, the text is verbatim transcription, but it can be just a brief summary. In any case, we want to find those parts of the signal that match (as well as possible) the provided text. These are searched by aligning the ASR output to the text via an algorithm proposed in [9]. The found segments are cut off (as in previous paragraph) and stored (together with the matched text fragments) in a `StackList`. The match score is computed via Eq. (1) mentioned in Sect. 6.4. At this stage, we do not insist on perfect (100%) match, as the segments will pass repeated decoding with a gradually improving AM later. Instead, we keep all the segments whose score is higher than a threshold (usually 70–80%). The procedure is called `DoMatchedSegmentation`.

Automatic Check and Optional Correction. This procedure takes the matched segments and classifies them into 2 sets: In the first, there are the segments that achieved 100% score. Their phonetic transcriptions are considered correct and hence they are moved to the AM training list (`TrainList`). The other are ordered according to their scores and prepared for optional manual inspection. This is the only instant where a human may (but does not need to) enter the automated process.

In order to minimize human work, we have developed a program whose interface is shown in Fig. 1. It utilizes the ordered list of imperfectly matched segments, and shows and plays them to the annotator. The words where the ASR output and the reference text differ are highlighted. The annotator just decides which is correct and clicks on it to fix the error. When needed, he/she types the correct word or modifies the pronunciation. If a segment contains speech which is not clear, it can be skipped or definitely removed from the list. The correction process is easy and fast. Moreover, it does not require a person who knows the target language. Within an hour it is possible to check and correct several hundreds of speech segments, because most contain just 1 or 2 errors. The corrected segments are automatically added to the `TrainList`. The other remain in the `StackList`. In our schemes we name this procedure `CheckAndCorrect`.

Acoustic Model Retraining. When the number of newly acquired (and annotated) segments in the `TrainList` is sufficiently large, we add them to the previously collected speech data and run a procedure that retrains the AM using the standard HMM training tool. We denote this step as `Retrain`.

Switching Between Phoneme Sets. As one of our schemes utilizes the cross-lingual part, we need auxiliary procedures that make switching between two phoneme sets, one of the source language (SL) and another for the target one (TL). Usually, they are applied at the beginning and end of the bootstrapping phase. In the former case, we need to map all the phonemes from the TL (Polish in this case) to those of the SL with an existing AM (e.g. Czech). This approximation is only temporal and it is not much critical. We use the phoneme map proposed in [8]. After its application, we get the Polish lexicon represented by Czech phonemes.

When the bootstrapping phase is finished, we switch back to the original lexicon. All the phonetic annotations made within the phase are changed to the original Polish phonetic set, using the lexicon as a reverse look-up table.

The two procedures are denoted as `MapPhonemes` and `RemapPhonemes`.

5.2 Data Annotation and AM Training Schemes

Here we present 3 schemes. The first one is used when we already have an AM for the target language, the second is applicable in the case when an AM is available for other than the target language, and the third finds its usage when the target language AM has become mature enough to allow for unsupervised annotations and training.

Iterative Data Annotation and AM Training. This scheme is applied in a situation when we have a large number of speech documents and each of them is associated with some text. The goal is to find the speech segments that match parts of the text, annotate them and use them for AM retraining. The scheme combines the basic procedures in an iterative loop. We suppose that at the start we already have an AM for the target language. At the end of each iteration, new annotated data are added to the training list and a new (better) AM is trained. With this AM we repeat the scheme either from the start (step 1, i.e. a new segmentation) or for the already segmented files (step 2). The former is useful when the initial AM was trained on a small amount of data. The scheme is finished when the number of newly annotated segments is too small to run another iteration.

```
IterativeRetraining:
1 For each Document
    DoMatchedSegmentation
2 For each Segment from StackList
    DoTranscription
    CheckAndCorrect
3 Retrain
4 Repeat from step 1 or 2
```

Cross-Lingual Iterative Training. This scheme is a modification of the previous one. It is used for initial bootstrapping when no AM for the TL is available. Here, we utilize an AM from an SL and do the temporary phoneme mapping. Moreover, we need to add a certain amount of training data (e.g. 10 h) from the SL to the `TrainList` to ensure proper performance of the HMM training procedure. After that the standard iterative scheme is started. When finished, all the annotations are remapped, the SL data are removed, and the target language AM is retrained once again.

```
CrosslingualTraining:
1 MapPhonemes
2 Add SL Data to Trainlist
3 IterativeRetraining
4 RemapPhonemes
5 Remove SL Data from Trainlist
6 Retrain
```

Unsupervised Data Acquisition and Training. This scheme is used when only audio data is available. In this case, the previous two schemes cannot be applied because they have no text to be matched. Here we utilize an idea which is similar to that proposed in [16]. A segment is transcribed by several different recognizers and if all the transcriptions are same, we consider them as correct and add these segments (with their annotations the `TrainList`. In fact, all the used recognizers have the same structure, but they have different AMs (trained on different data subsets), or different operating parameters. Usually, this scheme is used when we already have a mature AM for the target language. Though, it can be used also with recognizers that operate with AMs borrowed from different languages as shown in the above mentioned paper.

```
UnsupervisedTraining:
1 For each Document
    DoSegmentation
2 For each segment
3   For each recognizer
      DoTranscription
    If all ASR outputs are same
      Add to TrainList
4 Retrain
```

6 Practical Implementation and Evaluation

When building a robust AM for Polish we combined all the three schemes as described further.

6.1 Bootstrapping

For the initial phase we used the large archive of Polish Sejm, namely the video files and stenograms available at http://www.sejm.gov.pl. The video files contain speech of good quality provided by several hundreds of speakers. The stenograms are almost verbatim transcriptions of the talks, yet sometimes slightly smoothed or reformulated, e.g. without repeated words or phrases, sometimes using synonyms instead of actually spoken words, etc. They also contain some non-verbal information e.g. about reactions from the auditorium, which can be easily detected and removed.

We have chosen 20 random sessions from period 2013–2014. The stenograms were converted to plain text files and added to the corpus. Some frequent OOV names and specific words were inserted to the lexicon and the LM was recomputed. After that we started the cross-lingual training scheme described in Sect. 5.2. To initialize the process, we used the Czech AM and put 10 h of annotated Czech speech to the train list. After the first iteration, we got 1450 segments with an average duration 3.5 s, i.e. 84 min of annotated Polish speech data. A new (Czech-Polish mixed AM) was trained and used for improved resegmentation. In several subsequent loops we gained around 2000 new segments per iteration, from which about one third passed the manual check and correction. The scheme was stopped when the number of newly acquired segments dropped below 100. At that time 16,127 segments (17.9 h) were available. They were used to train the first genuine Polish AM.

6.2 Standard Iterative Retraining Process

As the next step, we took other 10 Sejm sessions and used them in the standard (monolingual) scheme as in Sect. 5.2. In this way, we acquired an additional amount of 15.6 h of speech. It would be possible to get much more, however we did not want to saturate the AM by one type of data. Instead, we searched for another source. We found several radio programs that have both audio and text (approximate transcription) on their web, e.g. http://www.polskieradio.pl/Rozmowy-Jedynki. We processed them in the same way and got 8.2 h.

6.3 Unsupervised Retraining Process

To increase the variety of the training data, we searched for other large sources of speech. Since our target application domain is broadcasting, we focused on major TV stations and their news programs. Unfortunately, these have no accompanying text and therefore we had to use the unsupervised scheme proposed in Sect. 5.2. We employed 4 recognizers, each trained on a different subset of the training data available at that time. The scheme processed some 120 news programs (each about 30 min long) from major Polish TV channels and eventually produced 16.4 h of annotated data. We checked manually a small subset of them and found that the transcriptions (when approved by the 4 different recognizers) were fully correct for 9 of 10 segments. In most cases, the errors were marginal) with a minimal impact on the trained AM. At the end of this phase we obtained an AM trained on 58 h of speech.

6.4 Evaluation

To evaluate the quality of the Polish ASR system and to document the progress after each phase, we have prepared a large test set. It is made of 4 news shows from Polish major TV stations (*TVN-Fakty, TVP2-Panorama, TVP1-Wiadomosci and Polsat-Wydarzenia*). The shows are complete; from the opening jingles to the closing ones. They include all types of speech occurring in news programs: clean speech read in studio, speech with background music or noise, spontaneous utterances recorded in streets, or a dubbed speech with a talk in a foreign language in background. A Polish native speaker has provided their verbatim transcriptions that were used as reference texts in evaluations.

Because we wanted to learn how the system performs under ideal conditions, we extracted a smaller test subset which contained only the clean speech from studio. The main parameters of the two sets are listed in Table 1.

Table 1. Parameters of test data (4 full TV shows and their studio speech extracts)

	Full shows	Studio speech only
Total duration [min]	105	23
Number of words	14,742	3,984
Out-of-vocabulary words [%]	0.92	1.03

During each development phase we run tests to evaluate the progress of AM training. We measured transcription accuracy using the standard formula for word recognition rate

$$WRR = (H - I)/N.100 \tag{1}$$

where N, H and I are the numbers of words in the reference text, and the numbers of hits and insertions, respectively.

The most relevant results are summarized in Table 2, where the amount of automatically collected training data and WRR values for the two test sets are listed. The first row shows the initial situation where no Polish data were available, in the second one there are the results after the bootstrapping phase (Sect. 6.1), next are those after the phase described in Sect. 6.2. When the last phase (Sect. 6.3) was finished, we got 58.1 h and trained two AMs: one based on standard Gaussian HMM and the other on deep neural networks. It can be observed that the DNN acoustic model performs significantly better, particularly for noisy and low quality parts of the TV shows, as it was demonstrated in several previous papers, e.g. [14].

Table 2. Word recognition rates (WRR) achieved with iteratively improving AMs

		WRR [%]	
Acoustic model	Hours of speech	Full shows	Studio speech
Czech (before bootstrapping)	–	50.4	63.2
Polish after bootstrapping	17.9	61.2	78.3
Polish after further retraining	41.7	68.7	84.6
Polish final GMM model	58.1	74.1	91.3
Polish final DNN model	58.1	79.2	92.1

7 Conclusions

In this paper, we present a series of methods and procedures that allowed us to build a Polish LVCSR system applicable to the automatic broadcast transcription task. We were able to adapt the existing modular ASR platform to a new language within a relatively short time without the need for a dedicated and expensive speech database. The lexicon and the language model were built from public texts available on the Internet and also the acoustic model training used audio data entirely from Polish web pages. The latter was possible due to the iterative schemes that automatically collected, processed, annotated and checked audio data, and trained the AM.

The accuracy we achieved with the best model is fairly good for the target application, which is automatic transcription of broadcast programs. Most errors that occur (namely in clean speech) are just confusions between similarly sounding word-forms of the same lemma, or omitted very short words (prepositions and conjunctions).

The proposed methods are language independent and we have already used them in the rapid development of ASR systems for several other Slavic languages [11].

Acknowledgment. This work was supported by the Technology Agency of the Czech Republic (project MultiLinMedia, no. TA04010199) and by the Student Grant Scheme at the Technical University of Liberec.

References

1. Chaloupka, J.: Digits to words converter for slavic languages in systems of automatic speech recognition. In: Karpov, A., Potapova, R., Mporas, I. (eds.) SPECOM 2017. LNCS (LNAI), vol. 10458, pp. 312–321. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66429-3_30
2. Demenko, G., Wypych, M., Baranowska, E.: Implementation of grapheme-to-phoneme rules and extended SAMPA alphabet in Polish text-to-speech synthesis. Speech Lang. Technol. **7** (17), 79–97 (2003)
3. Demenko, G., Grocholewski, S., Klessa, K., Ogorkiewicz, J., Wagner, A., Lange, M., Sledzinski, D., Cylwik, N.: JURISDIC: polish speech database for taking dictation of legal texts. In: Proceedings of LREC, pp. 1280–1287 (2008)
4. Demenko, G., et al.: Development of large vocabulary continuous speech recognition for polish. Acta Phys. Pol. A **1**(121), A-86 (2012)
5. Koržinek, D., Brocki, L.: Grammar based automatic speech recognition system for the Polish language. In: Jabłoński, R., Turkowski, M., Szewczyk, R. (eds.) Recent Advances in Mechatronics, pp. 87–91. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73956-2_18
6. Lööf, J., Gollan, C., Ney, H.: Cross-language bootstrapping for unsupervised acoustic model training: rapid development of a Polish speech recognition system. In: Proceedings of Interspeech, pp. 88–91 (2009)
7. Marasek, K.: Large vocabulary continuous speech recognition system for Polish. Arch. Acoust. **28**(4), 119–126 (2003)
8. Nouza, J., Boháč, M.: Using TTS for fast prototyping of cross-lingual ASR applications. In: Esposito, A., Vinciarelli, A., Vicsi, K., Pelachaud, C., Nijholt, A. (eds.) Analysis of Verbal and Nonverbal Communication and Enactment. The Processing Issues. LNCS, vol. 6800, pp. 154–162. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25775-9_15
9. Nouza, J., Cerva, P., Kucharova, M.: Cost-efficient development of acoustic models for speech recognition of related languages. Radioengineering **22**(3), 866–873 (2013)
10. Nouza, J., et al.: Speech-to-text technology to transcribe and disclose 100,000 + hours of bilingual documents from historical czech and czechoslovak radio archive. In: Proceedings of Interspeech, pp. 964–968 (2014)
11. Nouza, J., Safarik, R., Cerva, P.: ASR for south slavic languages developed in almost automated way. In: Proceedings of Interspeech, pp. 3868–3872 (2016)
12. Pawlaczyk, L., Bosky, P.: Skrybot–a System for Automatic Speech Recognition of Polish Language. Man-Machine Interactions, pp. 381–387. Springer, Heidelberg (2009)
13. Schultz, T.: GlobalPhone: a multilingual speech and text database developed at karlsruhe university. In: Proceedings of ICSLP, pp. 345–348 (2002)
14. Seps, L., Malek, J., Cerva, P., Nouza, J.: Investigation of deep neural networks for robust recognition of nonlinearly distorted speech. In: Proceedings of Interspeech, pp. 363–367 (2014)
15. Vu, N.T., et al.: Rapid bootstrapping of five eastern European languages using the rapid language adaptation toolkit. In: Proceedings of Interspeech, pp. 865–868 (2010)
16. Vu, N.T., Kraus, F., Schultz, T.: Multilingual A-stabil: a new confidence score for multilingual unsupervised training. In: Proceedings of Spoken Language Technology Workshop (SLT), pp. 183–188. IEEE (2010)
17. Ziółko, M., et al.: Automatic speech recognition system dedicated for Polish. In: Proceedings of Interspeech, pp. 3315–3315 (2011)

Automatic Transcription and Subtitling of Slovak Multi-genre Audiovisual Recordings

Ján Staš(✉), Peter Viszlay, Martin Lojka, Tomáš Koctúr, Daniel Hládek, and Jozef Juhár

Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Košice,
Park Komenského 13, 042 00 Košice, Slovak Republic
{jan.stas,peter.viszlay,martin.lojka,tomas.koctur,
daniel.hladek,jozef.juhar}@tuke.sk

Abstract. This paper summarizes a recent progress in the development of the automatic transcription system for subtitling of the Slovak multi-genre audiovisual recordings, such as lectures, talks, discussions, broadcast news or TV/radio shows. The main concept is based on application of current and innovative principles and methods oriented towards speech and language processing, automatic speech segmentation, speech recognition, statistical modeling and adaptation of acoustic and language models to a specific topic, gender and speaking style of the speaker. We have developed a working prototype of automatic transcription system for the Slovak language, mainly designed for subtitling of various types of single- or multi-channel audiovisual recordings. Preliminary results show a significant decrease in word error rate relatively from 2.40% to 47.10% for an individual speaker in fully automatic transcription and subtitling of Slovak parliament speech, broadcast news or TEDx talks.

Keywords: Automatic subtitling · Broadcast news · Lecture speech
Parliament speech · Speech recognition · Speech segmentation
User modeling

1 Introduction

The development of the automatic transcription systems for subtitling of various multi-genre audiovisual recordings is very popular research area, when human-machine interaction (HCI) through speech recognition technologies is becoming more and more integrated into the everyday use.

The TV broadcasters are interested in automated transcription of audiovisual recordings, such as academic lectures, talks, discussions [1], or broadcast news (BN) [5], because of the valid EU governments regulations watching the amount of TV shows with closed captions for hearing impaired in each member state.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 42–56, 2018.
https://doi.org/10.1007/978-3-319-93782-3_4

The European Federation of Hard of Hearing People (EFHOH) in its annual report "*Creating a barrier-free Europe for all hard of hearing citizens*" from year 2011 [3], addressed to the European Parliament, pushes ahead the idea to increase the ratio of programs accompanied by open or closed captions to 100% until year 2020 in each member state of the EU. This goal has already been reached in countries like Great Britain, Netherlands or France.

For Slovakia, the minimum amount is 10% for commercial and 50% for public broadcasters and this is usually achieved by manual transcriptions. That is why the global companies try to present products in this area, which usually work satisfactory for major languages, but they are insufficient for the minor ones as is Slovak[1]. As a result, broadcasters and subtitling companies are seeking for subtitling alternatives more productive than the traditional manual process [2].

Current trend in the area of the creation of closed captions is utilization of the automatic speech recognition and application of the modern principles and methods of the speech technologies in automatic transcription of speech in real time. The FP7 project SAVAS[2] provide such applications for major EU languages, as is English, French, German, Italian, Spanish, or Portuguese. Similar project of applied R&D for automatic subtitling was solved by the University of Western Bohemia in Pilsen for the Czech Television during research project ELJABR[3] solved in years 2006–2011 and the project ELJABR II that started in year 2011 and is still running. Several significant results in automatic subtitling of broadcast news were reported for the Slovenian [14] or Hungarian [22] language. Therefore, we have oriented our research on the development of a working prototype of the automatic subtitling system for the Slovak language.

The Slovak transcription system allows multi-channel automatic segmentation of the speech recordings at several levels based on the decomposition using principal component analysis with possible gender and speaker detection. It can be used for subtitling of various types of single- or multi-channel audiovisual recordings. The system makes use of multi-pass sequential speech recognition running on a computational server for each channel using a combination of multiple parallel speech recognition systems with different settings for general and specifically adapted acoustic and language models with hypothesis combination.

In the following sections we will describe the main components of the proposed automatic transcription system for subtitling of Slovak multi-genre audiovisual recordings in more detail.

2 Web-Based User Interface

The web-based user interface offers to upload and automatically process multimedia files. The user can select one of two methods for digital audio signal processing: single- or multi-channel automatic speech segmentation (see Fig. 1).

[1] http://googleblog.blogspot.sk/2009/11/automatic-captions-in-youtube.html.

[2] http://www.fp7-savas.eu/.

[3] http://www.kky.zcu.cz/en/research-fields/eljabr.

Multi-channel input module uses multimedia container Matroska (MKV), which is capable to store multiple audio/video streams. N-channel audio/video stream can be inserted into the MKV container. In the proposed system, streams are ordered as follows: stream #0 is video stream without audio track; streams #1 to #N are mono audio tracks; and #$N + 1$ is joined audio track.

Multi-channel MKV file inserted into the system is converted in to three audio or video formats: MP4, OGG, WebM, supported by HTML5 without any need to install external plugins to a web browser. For speech recognition, the system extracts audio tracks from the inserted media and converts them to the 16 kHz 16bit PCM mono audio format.

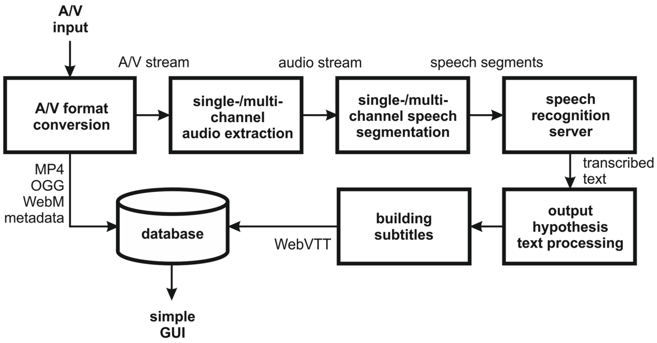
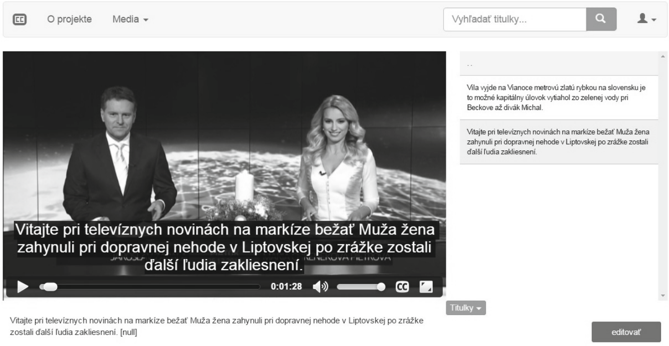


Fig. 1. Block diagram of the proposed automatic transcription and subtitling system



Vila vyjde na Vianoce metrovú zlatú rybkou na slovensku je to možné kapitálny úlovok vytiahol zo zelenej vody pri Beckove až divák Michal.

Vitajte pri televíznych novinách na markíze bežať Muža žena zahynuli pri dopravnej nehode v Liptovskej po zrážke zostali ďalší ľudia zakliesnení.

Vitajte pri televíznych novinách na markíze bežať Muža žena zahynuli pri dopravnej nehode v Liptovskej po zrážke zostali ďalší ľudia zakliesnení.

Vitajte pri televíznych novinách na markíze bežať Muža žena zahynuli pri dopravnej nehode v Liptovskej po zrážke zostali ďalší ľudia zakliesnení. [null]

Fig. 2. Graphical user interface

Extracted audio for each channel is split into short speech segments using SoX package[4] according to the time stamps obtained from the step of acoustic speech segmentation that will be discussed in the next section (see Sect. 3).

[4] http://sox.sourceforge.net/.

Speech segments are sent to the multi-threaded parallel speech recognition server according to actual server's capacity. The advantage of using speech recognition server is simultaneous speech recognition of multiple speech segments. Parallel speech recognition of each media is monitored automatically and servers load is dynamically allocated. To reduce the load on recognition server, audio segments are sent only when the server has enough capacity to recognize segment. Individual segments are then transcribed with appropriate configuration for speech recognition with a different setup for male and female gender, or individual speaker. If speaker's gender is not recognized by the gender detection (see Sect. 3), the general configuration for speech recognition with the background acoustic model (AM) and language model (LM) is used.

Speech recognition text output obtained from recognition server is time aligned to segmentation timestamps, post-processed to generate subtitles in WebVTT format, supported by HTML5. All subtitles are stored in WebVTT file and also in MySQL database for future processing, correction, and indexing.

Since HTML5 does not support MKV format, MKV video files are converted to MP4 format with 2 streams (audio/video), for further subtitling, archiving and indexing. The first stream is an original video stream without audio track and the second one is joined audio stream converted to AAC format. Selected media can be played with standard subtitles or subtitles as "karaoke". A simple web-based graphical user interface[5] (see Fig. 2) has been designed for this purpose [9].

3 Automatic Speech Segmentation

The accurate automatic speech transcription of an unknown audio stream depends on several processing steps that have to be performed before the main recognition. In the presented system, these steps are covered by speaker diarization, robust and discriminative feature extraction and finally, by building gender- and speaker-dependent AMs for gender detection and speaker identification purposes. In the recognition phase, a precise automatic speech segmentation is employed to identify voice/speech active segments that are additionally labelled with gender- and speaker-specific labels.

3.1 Speaker Diarization

The speaker diarization is an automated annotation of speech recordings with labels that represent speakers. This task is performed without any prior information, neither the number of speakers, nor their identities, nor samples of their voices are available. The speaker diarization can be also employed for helping speech recognition, facilitating the searching and indexing of audio archives and increasing the richness of automatic transcriptions [17].

The diarization usually consists of voice activity detection (VAD), in better case speech activity detection (SAD) and feature extraction. Speech segments

[5] http://isada.kemt.fei.tuke.sk/.

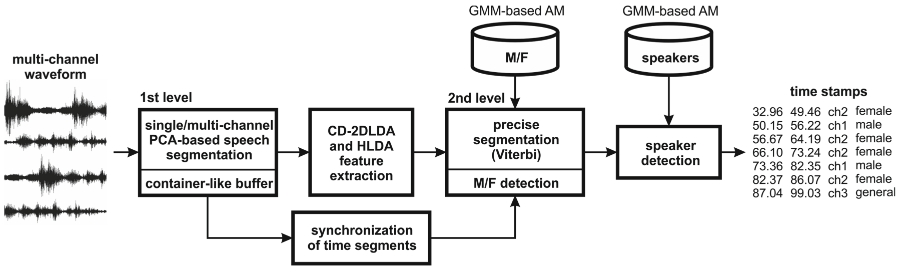


Fig. 3. Block diagram of the proposed automatic speech segmentation

belonging to the same speaker are clustered together by hierarchical agglomerative clustering. The Viterbi decoding (re-segmentation) is performed to generate a new segmentation realigned on the speaker boundaries.

In our framework, the LIUM SpkDiarization Toolkit [8,17] was successfully applied for the initial segmentation and speaker clustering. The diarization produced homogeneous segments belonging to the specific gender or speaker. According to the diarization outputs, the clustered and labelled data of the analyzed audio channel were employed for building gender-dependent (GD) or speaker-dependent (SD) AMs needed for gender detection and speaker identification. Note that these models are independent from the speech recognition process.

If annotated acoustic data are available, more precise GD or SD AMs can be built using the audio segments automatically extracted according to the manually generated transcriptions. The data from the diarization may be replaced by the extracted audio segments. If we suppose that the diarization toolkit provides a reliable clustering with low diarization error rate then the generated speech segments can be appropriately used for building AMs for detection.

The presented transcription system was primarily adapted to the BN data that were used for the offline diarization process and subsequently for building AMs for detection and identification. But generally, there is no limitation of using other type of acoustic data. In that case, the diarization followed by GD and SD modeling should be carried out again so that the acoustic conditions of the training data and the audio data to be transcribed will match each other.

3.2 Acoustic Feature Extraction

We found that the optimal segmentation performance can be achieved, when each module is optimized for the specific task separately. Therefore, the gender detection and speaker identification modules contain different advanced front-ends compared to our previous work [19]. More specifically, the AMs for the speaker identification task were trained on features generated by our own method called as class-dependent two-dimensional linear discriminant analysis (CD-2DLDA) that was proposed in [24] for speech recognition.

CD-2DLDA extends the classical 2DLDA to the class-dependent approach in order to improve the discriminability between the defined classes. In this setup, the speaker labels are treated as classes that are needed for estimation of the left and right transformation matrices. This method employs two-pass recognition strategy where the first pass utilizes the state-of-the-art 13-dimensional mel-frequency cepstral (MFC) coefficients to obtain the label of the class being identified. During the second pass, the class-dependent 2D transformation is performed and the final speaker identification step is carried out.

The second extended front-end utilizes the heteroscedastic linear discriminant analysis (HLDA) features [10] that are optimized to improve the gender detection performance. In this case, only three classes are available (male, female, or background). They are intended to estimate the classical LDA matrices that are further improved by the heteroscedastic iterative optimization criteria.

3.3 Gender- and Speaker-Dependent Modeling

The acoustic representations and the corresponding cluster information needed for the acoustic modeling were prepared previously in the diarization task (see Sect. 3.1). They were further transformed by CD-2DLDA and HLDA, depending on detection/identification task. All GD and SD models were trained as multivariate one-state left-to-right Gaussian mixture models (GMMs) with up to 1,024 PDFs (probability density functions) using the HTK [27]. The GD and SD recognizers employed a simple vocabulary, grammar and word net composed from the modeled units (genders and speaker IDs). Basically, the mentioned detectors worked as a simple phone-based recognizers where the phone units were replaced by the gender or speaker IDs. The GD and SD acoustic modeling frameworks have a number of interesting features. They allow to:

- train new GD AMs from scratch, if a transcription of new data has to be performed and new acoustic data are available;
- retrain an existing GD AM, if additional acoustic data are collected for the genders, whereas the robustness of AMs may be increased;
- train a new SD AM, if a new speaker appears in the speaker inventory;
- retrain easily an existing SD model, if additional SD data are collected.

3.4 Speech Segmentation

In general, it is possible to transcribe a continuous audio stream without any segmentation, but the computation time and decoding may take a very long time [5]. Automatic speech segmentation is usually applied to speed up the recognition process and to improve the overall performance by identifying and handling the specific parts in the recognized speech (speaker change boundaries, gender- or speaker-specific segments, non-speech events, different acoustic conditions, etc.).

The presented transcription system supports two independent modes of the speech segmentation: single- and multi-channel mode. The single-channel segmentation module was designed to process any kind of a standard single-channel audio stream (broadcast news, discussion, lecture, etc.). The multi-channel mode was primarily designed for BN transcription, where multiple audio streams are present at the same time, containing different types of audio (studio streams, external streams from reporters or interviewees, jingles, commercials, etc.) [25]. In this case, each channel employs independent processing with different configuration and the single-channel stream is transcribed by an independent recognizer.

In order to employ gender- or speaker-dependent speech recognition and thus improve the transcription performance, gender detection or speaker identification have to be carried out on the single or multi-channel audio stream. The gender detection can be performed using the default GD AMs (the detection rate will be satisfactory, because the models were trained on a sufficient amount of acoustic representations for each gender).

As was mentioned earlier, the speech processing supports new GD AM training directly from the recognized audio that does not match exactly the acoustic conditions of the default AMs. This operation is meaningful only if there are available enough training examples for both genders. The SD segmentation is not supported implicitly, if the single-channel waveform contains voices of unknown speakers that were not included in the SD training data. However, there is a possibility to train a new SD AM, if the recognized audio provides a sufficient amount of speaker examples that originate from the diarization. In the presented system, the multi-channel mode supports speaker identification because the speakers in the BN are known and the SD AMs can be at ease prepared in the previous training phase. Note that the inner principles of segmentation, GD and SD task for both single- or multi-channel mode are basically the same.

The proposed speech segmentation is a two-level sequential process described in Fig. 3. The first level is represented by our VAD [23] that is responsible for accurate speech or silence discrimination. In order to determine the VAD labels, the waveform is processed in the time domain by overlapping blocks extracted by rectangular window with standard length of 25 ms and 10 ms frame step. After re-arranging the samples of the current block into matrix, segmental time domain principal component analysis (PCA) [7] is applied to the sample matrix. After that, N eigenvalues are computed for each block, where N is the dimension of PCA space. The eigenvalues are used to determine the nature of the i-th segment (voice/silence) using a thresholding criterion [23]. In this way, the whole waveform is described by VAD coefficients that are further smoothed by moving average window. In our setup, we set N to 4 or 2 for signals with $f_s = 16\,\text{kHz}$ or 8 kHz, respectively. For detailed description of the described VAD, see [23].

Compared to our previous work [19], the segmentation module was considerably improved due to the fact that very short speech segments occurred at the segmentation output ($< 2\,\text{s}$). This effect caused producing one-word (and similar) segments that did not have good influence on the recognizer.In order to eliminate this phenomenon, we designed a container-like buffer that is applied to all generated speech-active segments. They are sequentially accumulated in the buffer until the predefined container length is reached. After that, the accumulated segments are joined together with respect to the original silence gaps.

We set the length of the container to approx. 20 s, thus the lengths of segments that are fed to the second segmentation level lie around the predefined value.

The second segmentation level employs the Viterbi algorithm and it is responsible for precise GD or SD segmentation. In other words, GD and SD recognizers are run to detect and locate gender- and speaker-change points that split these regions into gender- or speaker-homogeneous segments. At this stage, time stamps are generated with gender labels and if needed, they can be extended with speaker labels. The overall segmentation requires final time synchronization between the first and second level due to eliminated silent parts at the first level.

4 Speech Recognition Server

To support wide variety of applications the server-based speech recognition was adopted. The main advantage of the server-based solution is that the software and algorithms used can be optimized for one hardware configuration and under supervision of an expert, while the service can be provided to the users without any expert knowledge. This way any kind of devices, especially devices with low computation resources can be equipped with speech recognition capability. In our case the advantage is that this solution separates more complex system from speech recognition making it modular, scalable, easy to maintain and reusable for other purposes beyond this paper.

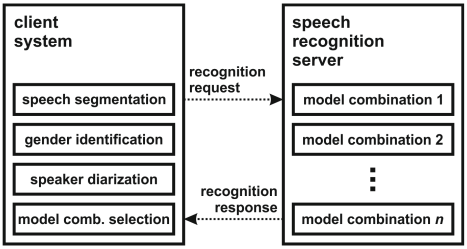


Fig. 4. Multi-threaded speech recognition server

According to the thought the whole system is broken into client and server part. Client system is responsible for extracting as much information as possible from input audio recording including speech segmentation, speaker diarization and gender identification while the server is responsible for speech recognition (see Fig. 4). The main idea is to use as much information as possible to identify acoustic and language model combination for subsequent speech recognition. Logically can be assumed that more matching AM to the speaker and channel characteristics and LM to the topic can increase the speech recognition accuracy. Another approach is to use multiple combination of AMs and LMs and combine output hypotheses to further increase the accuracy [4]. In our system we have used both approaches represented by a rule based identification for the selection of multiple best model combinations.

4.1 Speech recognition server

We developed a speech recognition server application on top of the Julius speech recognition engine [11] that was extensively modified to support multi-thread parallel speech recognition allowing to share resources among the threads to minimize memory footprint [13]. The instances of the engine (each in its own thread) are controlled by configuration manager that creates, destroys and caches idle ones for later use (Fig. 5). Connection manager receives requests from the users through TCP port and makes the data available to the engine instance via buffer. Buffer is created for each opened connection and immediately starts recording acoustic data even when they are not currently consumed by any engine instance. Buffer has fixed length, thus in case of low space further transfer is postponed until it is available again. The data are transfered using simple communication protocol that consists from header indicating the size of the following data chunk. Header with zero length data chunk is used to indicate end of speech and that user is waiting for final recognition result. Service manager requests engine instances from configuration manager on each new connection and connects them together, while it also collects closed connections and returns the buffers and the engine instances to their respective managers.

The server is configurable, the maximum number of parallel engine instances can be controlled along with the maximum allowed connections. In our case, the maximum connection number can be higher than the instances depending on the hardware resources. Smaller number of engine instances can process larger number of input connections as the speech recognition can be faster than real time and the engine instances are shared among the waiting connections. Pre-created number of instances can be also set for faster response on new connections.

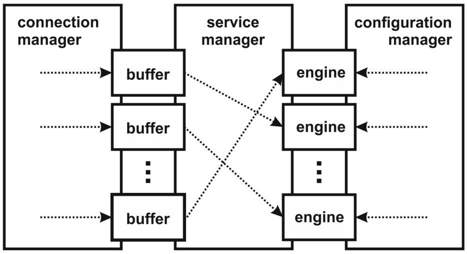


Fig. 5. Structure of the speech recognition server for one system configuration

4.2 Client system

As was presented above the client's responsibility is to extract as much information as possible and to make choice about AM/LM combination. In order to not increase complexity of the server implementation nor communication protocol, multiple server instances in parallel with different model combinations and running on different ports are used. The client then uses TCP port of speech recognition server with desired combination of the models. To select the right combination of models a set of rules are used depending on completeness of information extracted. If the gender identification is available then gender-dependent model combination is used along with gender-independent and their outputs are later combined. If no gender identification is available then only gender-independent model combination is used. In case of selecting more than one model combinations, the outputs are merged to single hypothesis using the ROVER algorithm [4]. In fact a modified version of the ROVER is used with additional preprocessing of input hypothesis that consists from score smoothing [12].

The ROVER algorithms is following. First it incrementally creates word transition network from input hypothesis while it aligns only two of them at a time. Substitutions and deletions are handled using NULL transition score $C(@)$ that can be estimated on a development set. In the next step it uses voting mechanism based on confidence score for each alternative word $C(w_i)$ and its occurrence $N(w_i)$ normalized by count of the combined systems according Formula 1 to select the resulting word alternative.

$$V(w_i) = \alpha \frac{N(w_i)}{N_S} + (1 - \alpha)C(w_i). \tag{1}$$

Confidence score $C(w_i)$ for each word alternative can be computed as average or maximum score found in input hypotheses [12]. In our case, we empirically estimated the $C(@)$ score to 0.5 and α to 0.7 in order to control the balance between the number of word occurrences and its confidence scores. The mentioned smoothing of input confidence scores is done according Formula 2.

$$C(w_t) = \beta C(w_{t-1}) + (1 - \beta)C(w_t). \tag{2}$$

The new score $C(w_t)$ for word w_t is computed from previous word in time w_{t-1} and β is smoothing factor controlling amount of transfered score.

4.3 Acoustic Modeling for Speech Recognition

In presented system we have used gender-dependent and gender-independent AMs. Both AMs are based on triphone context-dependent 3-state hidden Markov models (HMMs) with 32 probability density functions (PDFs) on a state and on a feature vector type MFC coefficients with zeros, delta and acceleration coefficients and cepstral mean normalization removed (MFCC_0DAZ). The training process was modified, tree-based state tying was replaced by the effective triphone mapping algorithm [18]. The audio format of recordings for training were 16 kHz 16 bit PCM mono. The gender-independent AM was trained on manually annotated speech recordings from the database of judicial readings (100h); read phonetically rich sentences and newspaper articles (150h); broadcast news (250h); parliament speech (90h); and the Court TV shows (80h) [15,18,25]. Gender-dependent AMs were created by splitting the training database for each gender and building two separate AMs for females and males.

4.4 Language Modeling for Speech Recognition

The background LM was created by using the SRILM Toolkit [21]. It was restricted to the vocabulary size of 500 thousand unique words and smoothed with the Witten-Bell back-off algorithm. The trigram model was trained on the background web-based corpus of Slovak written texts.

The background corpus was created by the web-crawling, gathering and processing agent proposed earlier [6]. The agent explores Slovak web-sites and analyzes content of each web-page and stores it in the database. The collected text is then processed by our tools for tokenization, sentence boundary detection, automatic correction, and duplicity filtering, and segmented into small paragraphs for better representation in the vector space. We have also created a set of rules for removing duplicates and grammatically incorrect segments from the text corpora, either on the document, paragraph, or sentence levels. The corpus size of about 1.89 billion tokens and 110.75 million sentences was then segmented into 5,929,843 paragraphs with approximately 315 words on average.

Moreover, we proposed a semantic indexing and document retrieval approach for user-specific modeling to improve speech recognition accuracy for individual speakers [20]. The latent semantic indexing (LSI) and vector space modeling [26] were implemented to retrieve a subset of documents from the background corpus relevant to the topic and speaking style of a speaker. We select a subset of documents semantically similar to the output hypotheses from recognized speech segments in the first decoding stage. After that, a small user-specific LM is created from the relevant documents, interpolated with the background LM, adapted to the current topic and speaking style of a the speaker and applied during the second decoding stage.

5 Speech Recognition Results

The experiments have been oriented on evaluation of the LM perplexity and performance of the proposed transcription system on the test data using word error rate. Test data were represented by real speech recordings obtained from 10 native Slovak speakers (5 females and 5 males) for each of the 3 different domains: parliament speech, broadcast news TV shows and TEDx talks. Speakers from each domain were chosen carefully taking different topics, acoustic conditions, out-of-vocabulary (OOV) word rates, speaking rates and speaking styles into account to be simulated the real-life conditions as much as possible.

In this context, perplexity (PPL) is defined as reciprocal value of the (geometric) average probability assigned by the language model to each word in the test data and is related to cross-entropy $H(W)$ by the equation:

$$PPL = 2^{H(W)} = \frac{1}{\sqrt[n]{P(W)}} = \frac{1}{\sqrt[n]{P(w_1 w_2 \dots w_n)}}, \tag{3}$$

where $P(w_1 w_2 \dots w_n)$ is the probability of sequence of n words in a history.

Word error rate (WER) was used to evaluate overall performance of the proposed automatic transcription system on the test data. WER is computed by comparing reference annotations against recognized result as follows:

$$WER = \frac{N_{SUB} + N_{DEL} + N_{INS}}{N_{REF}} \times 100\ [\%], \tag{4}$$

where N_{SUB} refers to the number of substituted words, N_{DEL} is related to words, which are missed out, N_{INS} indicates the number of words incorrectly added by the recognizer, and N_{REF} is the total number of words in the reference.

In order to evaluate the performance of the proposed system, each recording has been recognized in configuration setup with both background acoustic and language models and models adapted to the specific topic, gender and speaking style of the speaker. The output hypotheses were combined using a modified ROVER algorithm and evaluated for each speaker separately. After that, labelled speech transcriptions has been created and compared with the reference.

Table 1 summarizes the average state-of-the-art out-of-vocabulary (OOV) word rates [%], language model perplexity (PPL) and word error rates (WER) [%] evaluated separately for each speaker in the context of Slovak multi-genre automatic speech recognition and transcription.

As we can see from the Table 1, the speech recognition results are promising and comparable with other European languages considering the fact that the Slovak language is a highly inflectional and more complex than the other European languages. The overall performance of the proposed automatic transcription and subtitling system was increased relatively by about 8.27% WER in the context of broadcast news transcription task, about 9.62% WER in transcription of Slovak TEDx talks and about 25.98% for parliament speech.

It can be concluded that using multiple speech recognizers in various of configurations and application of gender-dependent acoustic modeling, adaptation of language models to the specific topic and speaking style, and utilization of N-best rescoring on the word recognition level we achieved further improvement in transcription of semi-spontaneous and spontaneous Slovak speech.

6 Discussion and Future Work

Automatic transcription works well without any known bugs, but system is still under the development. Several new features are gradually added to the current prototype. Although the initial experimental evaluation reports that the word error rates are promising, there is still room for some new innovations, system tuning and improvements of the underlying speech technology. There is a number of challenging tasks associated with the further development of the current speech transcription system leading to more promising performance. Moreover, the system's modularity, its interesting features and modern approaches makes it highly applicable in many other domain-oriented tasks.

If we look at the server-based speech recognition, there are several possibilities of improvements. One of the most challenging tasks is to apply the discriminative feature transformations in the block of feature extraction and acoustic

Table 1. Overall speech recognition results evaluated on speech recordings obtained from 10 native Slovak speakers for each of the 3 different domains: parliament speech, broadcast news and TEDx talks

Speaker ID	Gender	Segments	Reg. words	*OOV* [%]	Backgr. models		Adapt. models		*Δ* Relative
					PPL	WER [%]	PPL	WER [%]	WER [%]
	Parliament speech								
jva11ba	F	143	1319	1.34	428.70	14.56	227.40	10.99	−24.52
lzi11ba	F	23	240	0.97	391.22	14.17	158.24	8.33	−41.21
mgi11ba	F	64	684	1.27	731.49	7.89	112.54	4.24	−46.26
rzm11ba	F	37	386	0.67	405.68	13.73	89.15	8.55	−37.73
vto11ba	F	204	1733	1.42	316.27	11.89	74.56	6.29	−47.10
lka11ba	M	123	1130	0.79	569.04	16.02	327.69	12.04	−24.84
phr11ba	M	498	4304	0.83	154.50	7.62	91.10	6.02	−21.00
pka11ba	M	105	931	1.32	426.83	18.15	293.30	17.40	−4.13
ppa11ba	M	249	1921	1.47	611.83	11.61	510.69	10.88	−6.29
rsu11ba	M	274	2309	0.96	179.69	14.81	111.03	11.82	−20.19
	broadcast news								
bde15ba	F	101	1020	0.49	837.46	9.61	683.97	8.14	−15.30
dbr15ba	F	34	374	0.54	566.76	7.22	392.91	6.42	−11.08
jki15ba	F	92	1219	1.23	398.23	10.58	333.17	10.25	−3.12
kte15ba	F	92	1036	1.83	713.90	20.75	590.37	19.31	−6.94
smi15ba	F	227	2764	2.24	595.15	15.92	472.76	14.94	−6.16
doc15ba	M	205	2684	7.23	270.76	20.01	165.28	18.93	−5.40
mba15ba	M	35	354	3.96	790.33	18.93	587.38	15.82	−16.43
mko15ba	M	56	631	3.96	402.23	29.48	341.29	27.89	−5.39
msi15ba	M	88	848	6.89	430.89	8.37	332.38	7.78	−7.05
msp15ba	M	314	3834	6.26	359.90	22.38	230.68	20.34	−9.12
	TEDx talks								
dfa13ba	F	180	1937	3.82	568.52	36.24	432.22	31.80	−12.25
jma13ke	F	129	1395	2.29	361.86	54.34	285.35	45.59	−16.10
mbe12ba	F	112	1196	3.01	466.48	45.32	371.65	42.47	−6.29
mvi14ke	F	383	2433	1.48	484.71	50.31	389.30	42.75	−15.03
zwi14ke	F	168	1627	3.69	617.24	22.19	523.96	18.68	−15.82
aku12ba	M	229	1738	1.84	617.19	17.09	478.17	15.54	−9.07
dve13ke	M	307	3198	2.06	426.40	58.22	322.85	56.10	−3.64
mka14tn	M	136	1487	5.92	531.44	35.51	393.41	34.23	−3.61
rse12ba	M	99	871	4.36	493.83	31.80	354.99	27.78	−12.64
tro13ke	M	377	3721	4.27	390.29	32.57	298.09	31.79	−2.40

modeling. Regarding to the automatic speech segmentation, we are planning to find an optimal configuration of features for gender- and speaker-dependent modeling. Also, more sophisticated decoding based on finite state transducers such as Kaldi for speech recognition [16] is necessary.

Acknowledgement. The research in this paper was supported by the Faculty of Electrical Engineering and Informatics at the Technical University of Košice under the research project FEI-2015-30, by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the research project VEGA 1/0511/17 and the Slovak Research and Development Agency under the research project APVV-15-0517.

References

1. Akita, Y., Watanabe, M., Kawahara, T.: Automatic transcription of lecture speech using language model based on speaking-style transformation of proceeding texts. In: Proceedings of the INTERSPEECH 2012, Portland, Oregon, USA, pp. 2326–2329 (2012)
2. Alvarez, A., Mendes, C., Raffaelli, M., Luís, T., Paulo, S., Piccinini, N., Arzelus, H., Neto, J., Aliprandi, C., del Pozo, A.: Automating live and batch subtitling of multimedia contents for several European languages. Multimed. Tools Appl. **75**, 10823–10853 (2015)
3. Bobeldijk, M., Ellisen, K.M., Lamby, J., Schaeding, R., Best-Smolarek, L.: Creating a barrier-free Europe for all hard of hearing citizens: State of subtitling access in EU. Technical report 2011, European Federation of Hard of Hearing People, Stockholm, Sweden (2011)
4. Fiscus, J.G.: A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER). In: Proceedings of the ASRU 1997, Santa Barbara, California, USA, pp. 347–354 (1997)
5. Gauvain, J.L., Lamel, L., Adda, G.: The LIMSI broadcast news transcription system. Speech Commun. **37**, 89–108 (2002)
6. Hládek, D., Ondáš, S., Staš, J.: Online natural language processing of the Slovak language. In: Proceedings of the CogInfoCom 2014, Vietri sul Mare, Italy, pp. 315–316 (2014)
7. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, New York (1986). https://doi.org/10.1007/978-1-4757-1904-8
8. Kiktová, E., Juhár, J.: Comparison of diarization tools for building speaker database. Adv. Electr. Electron. Eng. **13**(4), 314–319 (2015)
9. Koctúr, T., Pleva, M., Juhár, J.: Interface for smart audiovisual data archive. In: Proceedings of the RADIOELEKTRONIKA 2015, Pardubice, Czech Republic, pp. 292–294 (2015)
10. Kumar, N.: Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition. Ph.D. thesis, Johns Hopkins Universtiy, Baltimore, Maryland (1997)
11. Lee, A., Kawahara, T., Shikano, K.: Julius - an open source real-time large vocabulary recognition engine. In: Proceedings of the EUROSPEECH 2001, Aalborg, Denmark, pp. 1691–1694 (2001)
12. Lojka, M., Juhár, J.: Hypothesis combination for Slovak dictation speech recognition. In: Proceedings of the 56th International Symposium ELMAR 2014, Zadar, Croatia, pp. 43–46 (2014)
13. Lojka, M., Ondáš, S., Pleva, M., Juhár, J.: Multi-thread parallel speech recognition for mobile applications. J. Electr. Electron. Eng. **7**(1), 81–86 (2014)
14. Maučec, M.S., Žgank, A.: Speech recognition system for Slovenian broadcast news. In: Ipsic, I. (ed.) Speech Technologies, pp. 105–112. InTech, Rijeka (2012)
15. Pleva, M., Juhár, J.: TUKE-BNews-SK: Slovak broadcast news corpus construction and evaluation. In: Proceedings of the LREC 2014, Reykjavik, Iceland, pp. 1709–1713 (2014)
16. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K.: The Kaldi speech recognition toolkit. In: Proceedins of the ASRU 2011, Waikoloa, Hawaii, US, pp. 1–4 (2011)

17. Rouvier, M., Dupuy, G., Gay, P., Khoury, E., Merlin, T., Meignier, S.: An open-source state-of-the-art toolbox for broadcast news diarization. In: Proceedings of the INTERSPEECH 2013, Lyon, France, pp. 1477–1481 (2013)
18. Rusko, M., Juhár, J., Trnka, M., Staš, J., Darjaa, S., Hládek, D., Sabo, R., Pleva, M., Ritomský, M., Ondáš, S.: Recent advances in the Slovak dictation system for judicial domain. In: Proceedings of the LTC 2013, Poznań, Poland, pp. 555–560 (2013)
19. Staš, J., Viszlay, P., Lojka, M., Koctúr, T., Hládek, D., Kiktová, E., Pleva, M., Juhár, J.: Automatic subtitling system for transcription, archiving and indexing of Slovak audiovisual recordings. In: Proceedings of the LTC 2015, Poznań, Poland, pp. 186–191 (2015)
20. Staš, J., Zlacký, D., Hládek, D.: Semantically similar document retrieval framework for language model speaker adaptation. In: Proceedings of the RADIOELEKTRONIKA 2016, Košice, Slovakia, pp. 403–407 (2016)
21. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proceedings of the ICSLP 2002, Denver, USA, pp. 901–904 (2002)
22. Varga, Á., Tarján, B., Tobler, Z., Szaszák, G., Fegyó, T., Bordás, C., Mihajlik, P.: Automatic close captioning for live hungarian television broadcast speech: a fast and resource-efficient approach. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) SPECOM 2015. LNCS (LNAI), vol. 9319, pp. 105–112. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23132-7_13
23. Vavrek, J., Viszlay, P., Kiktová, E., Lojka, M., Juhár, J., Čižmár, A.: Query-by-example retrieval via fast sequential dynamic time warping algorithm. In: Proceedings of the TSP 2014, Berlin, Germany, pp. 469–473 (2014)
24. Viszlay, P., Lojka, M., Juhár, J.: Class-dependent two-dimensional linear discriminant analysis using two-pass recognition strategy. In: Proceedings of the EUSIPCO 2014, Lisbon, Portugal, pp. 1796–1800 (2014)
25. Viszlay, P., Staš, J., Koctúr, T., Lojka, M., Juhár, J.: An extension of the Slovak broadcast news corpus based on semi-automatic annotation. In: Proceedings of the LREC 2016, Portorož, Slovenia, pp. 4684–4687 (2016)
26. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the 7th LREC 2010 Workshop: New Challenges for NLP Frameworks, Valleta, Malta, pp. 46–50 (2010)
27. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.4). Cambridge University, Cambridge (2006)

Multiword Expressions

SEJF - A Grammatical Lexicon of Polish Multiword Expressions

Monika Czerepowicka[1(✉)] and Agata Savary[2]

[1] Faculty of Humanities, University of Warmia and Mazury in Olsztyn, Olsztyn, Poland
czerepowicka@gmail.com
[2] Université François Rabelais Tours, Tours, France
agata.savary@univ-tours.fr

Abstract. We present SEJF, a lexical resource of Polish nominal, adjectival and adverbial multiword expressions. It consists of an intensional module with about 4,700 multiword lemmas assigned to 160 inflection graphs, and an extensional module with 88,000 automatically generated inflected forms annotated with grammatical tags. We show the results of its coverage evaluation against an annotated corpus. The resource is freely available under the Creative Commons BY-SA license.

1 Introduction

Multiword expressions (MWEs) are linguistic objects containing two or more words and showing some degree of non-compositionality. For instance, the meaning of *to kick the bucket* (i.e. to die) cannot be predicted from the meanings of its components, while the singular number of a *cross-roads* is not inherited from the component which should normally be its headword (*roads*). MWEs encompass versatile objects: compounds (*all of a sudden*, *air brake*), complex terms (*random access memory*), multiword named entities (*European Union*), light-verb constructions (*to take a decision*), idioms (*to kick the bucket*), proverbs (*fortune favors the bold*), etc. Basic facts about MWEs are that they are pervasive in natural language texts, they show idiosyncratic behavior at the level of segmentation, morphology, syntax, semantics or pragmatics, they are concerned by sparseness problems and they are underrepresented in language resources and tools. In morphologically rich, e.g. Slavic, languages MWEs pose additional challenges due to the high number of morphosyntactic variants under which they occur in texts.

In this paper we focus on Polish compounds. We present SEJF (pl. Słownik Elektroniczny Jednostek Frazeologicznych 'Electronic Dictionary of Phraseological Units'), a grammatical lexicon of Polish MWEs containing over 4,700 compound nouns, adjectives and adverbs, where inflectional and word-order variation is described via fine-grained graph-based rules. It is provided under two forms – intensional (lemmas and inflection rules) and extensional (list of morphologically

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 59–73, 2018.
https://doi.org/10.1007/978-3-319-93782-3_5

annotated variants) – and is available[1] under the terms of the Creative Commons BY-SA license[2].

2 Data Sources

One of the major data sources for the SEJF lexicon was the National Corpus of Polish[3] (NKJP, Narodowy Korpus Języka Polskiego) [19]. The tagsets of both resources are compliant, which should facilitate the future use of SEJF in corpus studies.

The NKJP corpus was also used as a source of illustration and verification of research hypotheses. On the basis of concordance lists we verified the forms of the paradigms of almost each MWE included in the lexicon. We also used the corpus to find new, previously undescribed, MWEs thanks to automatic MWE extraction methods developed by the Wrocław University of Technology [5]. Each of the extracted MWE candidates was manually validated by the lexicographer.

Phraseological units were also acquired from theoretical and lexicographical studies of contemporary Polish. A group of about 1,500 nominal compounds, analyzed by [12], was the first to be encoded in the dictionary. Some adjectival units were drawn from a dictionary of comparisons [3]. Adverbial units were acquired from two other monographs: [6,31].

3 Formalism and Tool

The grammatical description of MWEs in SEJF was done within Toposław [16], a lexicographic framework offering a user-friendly graphical interface over three core components:

- Morfeusz [32] – a morphological analyzer and generator of Polish simple words, containing full paradigms of over 250 thousand lemmas.
- Multiflex [25] – a formal language and a tool based on graphs, which describes each inflected form of a MWE as a specific combination of its components. The relation from MWEs to graphs is one-to-many: each MWE (no matter how complex it is) has one particular graph assigned to it, while one graph can describe any number of MWEs.
- A graph editor stemming from Unitex[4], a multilingual corpus processor.

While Morfeusz is Polish-specific, the two other components have also been applied to Serbian, Greek and Macedonian, as mentioned in Sect. 8. Thus, Toposław as a whole is adaptable to another language, provided that a morphological module for simple words in this language exists and that some interface constraints between this module and Multiflex are fulfilled – cf. [25].

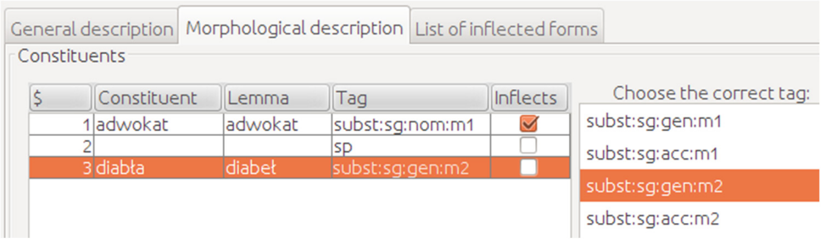
[1] http://zil.ipipan.waw.pl/SEJF.

[2] http://creativecommons.org/licenses/by-sa/3.0/.

[3] http://clip.ipipan.waw.pl/NationalCorpusOfPolish.

[4] http://www-igm.univ-mlv.fr/~unitex/.

The description of a MWE in Toposław is a multistage procedure. Firstly, the lexicographer assigns the MWE to the appropriate morphosyntactic class equivalent to one of the 33 *flexemes* (inflectionally motivated POSs) used in the NKJP corpus. Secondly, the MWE is segmented into words and separators, whereas the latter are considered full-fledged components that can further be referred to in inflection graphs. Thirdly, each component word is automatically assigned a list of all lemmas and morphological tags stemming from Morfeusz, thus all possible homonyms are distinguished. The lexicographer manually disambiguates each word by choosing the right interpretation. Figure 1 shows the nominal MWE *adwocat diabła* ‘devil’s advocate’, which has been segmented into three components, including a space. The first component is marked by the lexicographer as admitting inflection. The last one obtains four morphological interpretations, the third of which is correct.



$	Constituent	Lemma	Tag	Inflects
1	adwokat	adwokat	subst:sg:nom:m1	☑
2			sp	□
3	diabła	diabeł	subst:sg:gen:m2	□

subst:sg:gen:m1
subst:sg:acc:m1
subst:sg:gen:m2
subst:sg:acc:m2

Fig. 1. Segmentation and morphosyntactic annotation of the nominal MWE *adwokat diabła* ‘devil’s advocate’ in Toposław. The following codes are used: accusative case (`acc`), genitive case (`gen`), masculine animate gender (`m2`), masculine human gender (`m1`), singular (`sg`), space (`sp`), and substantive (`subst`).

In the last step, the lexicographer manually chooses an existing inflection graph (or creates a new one if needed) describing inflected forms of the current MWE entry. Figure 2 shows the inflection graph `NC-O_N` (cf. Table 2 for the meaning of the `NC`, `O` and `N` codes) for the entry from Fig. 1. Graph paths are applied from left to right and the numbered boxes in them correspond to constituents. The formulae inside boxes consist of constituents’ indexes and equations on morphological constants and variables. These equations impose constraints on the inflection, variation and agreement of constituents. Here, the formula ⟨*$1:Case=$c;Nb=$n*⟩ says that the first component (here: *adwokat*) inflects freely for case and number. The formulae appearing below paths determine the features of the inflected forms of the whole MWE as a function of the features of its constituents. Here, each form resulting from the unique path inherits its gender from the first constituent and has the conforming case and number (⟨*$1:Gen=$1.Gen;Case=$c;Nb=$n*⟩). Variables like *$c* or *$n* are freely defined by the user and subject to unification, i.e. if they reoccur on the same path the respective components must agree (cf. Sect. 5 and Fig. 4).

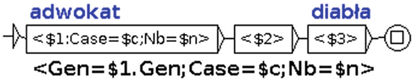
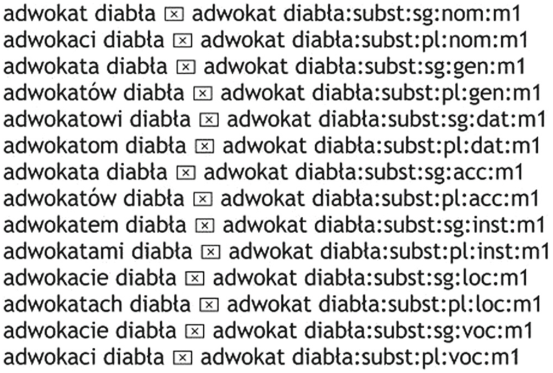


Fig. 2. Inflection graph NC-O_N for the nominal MWE *adwokat diabła* ‘devil’s advocate’.

When applying the graph in Fig. 2 to the entry in Fig. 1, we automatically obtain the list of all inflected forms and their morphological tags, as shown in Fig. 3.



```
adwokat diabła ☒ adwokat diabła:subst:sg:nom:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:nom:m1
adwokata diabła ☒ adwokat diabła:subst:sg:gen:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:gen:m1
adwokatowi diabła ☒ adwokat diabła:subst:sg:dat:m1
adwokatom diabła ☒ adwokat diabła:subst:pl:dat:m1
adwokata diabła ☒ adwokat diabła:subst:sg:acc:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:acc:m1
adwokatem diabła ☒ adwokat diabła:subst:sg:inst:m1
adwokatami diabła ☒ adwokat diabła:subst:pl:inst:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:loc:m1
adwokatach diabła ☒ adwokat diabła:subst:pl:loc:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:voc:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:voc:m1
```

Fig. 3. Inflection paradigm for the nominal MWE *adwokat diabła* ‘devil’s advocate’.

4 Contents of the Lexicon

Table 1 shows the current state of SEJF. Complete entries are those whose components’ inflection is fully handled by Morfeusz and Multiflex, thus the generation of the inflected forms for these entries could be fully performed. Problematic entries are those containing components which are unknown or wrongly handled.

Table 1. Contents of the lexicon.

	MWE lemmas		Inflected forms	Graphs
	Complete	Problematic		
Nouns	3,705	188	46,021	115
Adjectives	422	33	41,984	30
Adverbs	608	0	608	8
Others	40	1	113	5
ALL	4,775	222	88,726	158

On average, compound nouns have over 12 inflected forms – most of them inflect for case (with 7 case values) and some inflect for number (2 values). Compound adjectives are much more productive, with as many as almost 100 inflected forms on average, due to the case, number and gender inflection (with 9 gender values – 3 masculine, 1 feminine, 2 neuter and 3 plurale tantum ones – according to the Morfeusz tagset). Compound adverbs do not inflect, while among other compounds – selected conjunctions, particles and numerals – only the last ones inflect. The inflection graphs are mostly rather simple: 152 of them contain only one path representing inflection and, possibly, agreement of components. Eight remaining graphs (assigned to 154 MWEs in total) contain two paths, which account mainly for a flexible word order. Table 2 shows the most frequently assigned inflection graphs, the corresponding syntactic structures and examples of the assigned entries. A large majority of them consists of a noun and an agreeing adjective in both orders.

Table 2. Distribution of the most frequently assigned inflection graphs. The following codes are used: nominal compound (`NC`), variable component (`O`), invariable component (`N`), substantive (`S`), substantive in genitive (`Sgen`), and adjective (`Adj`).

Graph	Syntactic structure	Comment	MWE examples	Assigned MWEs
NC-O_O-1+	S Adj	Inflection for number	*koń trojański* 'Trojan horse'	1,153
NC-O_O-1	Adj S	Inflection for number	*aksamitna rewolucja* 'velvet revolution'	556
NC-O_O-2t	S Adj	Fixed number	*dobra osobiste* 'personal belongings'	426
NC-O_O-1t	Adj S	Fixed number	*czarna magia* 'black magic'	396
NC-O_N	S Sgen	Inflection for number	*adwokat diabła* 'devil's advocate'	351

5 Interesting Problems

The Toposław suite allows to successfully encode most of the nominal Polish MWEs however not all of them. For instance masculine human gender nouns are challenging in the sense that they exhibit not only the regular case and gender inflection but also have alternative depreciative forms in plural which are stylistically marked and show the speaker's pejorative attitude to the persons named by the multiword noun. Grammatically speaking, depreciative forms differ from the regular ones in plural nominative and vocative, namely they take the masculine animate gender `m2` (e.g. *adwokaty* instead of *adwokaci* 'advocates'). Because of the unusual gender, these forms constitute a separate flexeme (of type *depr*, cf. the NKJP tagset[5]). Since Toposław does not currently allow to gather several flexemes of the same lemma in one lexeme, generating depreciative forms for masculine human nominal compounds (e.g. *adwokaty diabła* 'devil's advocates') is blocked.

Another reason of a deficient description of the inflection paradigms is the (inevitable) incompleteness of Morfeusz. Neologisms such as *rozporkowy* (relative adjective for a trousers' fly) are not encoded, therefore compounds such as *afera rozporkowa* (lit. *fly affair*) 'a sexual scandal' cannot be automatically inflected.

Challenging examples which Toposław allows us to cover include variable word order, as in *automatyczna sekretarka, sekretarka automatyczna* (lit. *automatic secretary*) 'answering machine', or fluctuation of the grammatical gender. For instance, the nominal unit *czerwony pająk* (lit. *red spider*) 'communist' is exocentric in that its noun component *pająk* 'spider' is in masculine human animate gender (m2), while the whole compound, denoting a person, has the masculine human (m1) behavior. As shown on the upper path in Fig. 4, while the case and number of the whole MWE are conforming to the ones of the (inflected) noun and adjective, it's gender is not inherited from component 3 but given by the constant value m1. The major difference in inflection paradigms of masculine human and animate nouns is in the plural accusative form. It is equal to the plural genitive for m1 (*czerwonych pająków*) and to the plural nominative for m2 (*czerwone pająki*). The second path in Fig. 4 accounts for the m2-to-m1 shift: the accusative plural masculine human form of the whole compound is obtained by combining the genitive rather than the accusative forms of the two components. The inflection paradigm generated by the graph in Fig. 4 for *czerwony pająk* is shown in Fig. 5.

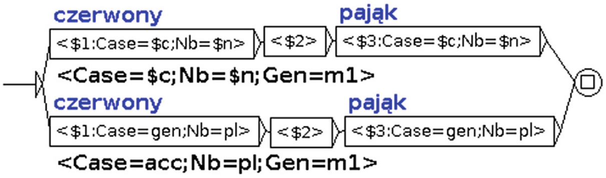
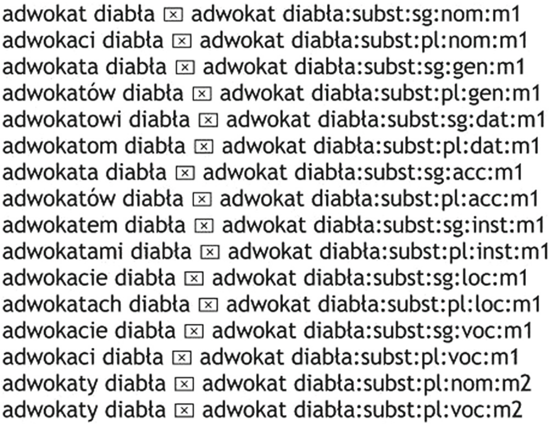


Fig. 4. Inflection graph NC-O_N describing a masculine gender fluctuation in *czerwony pająk* (lit. *red spider*) 'communist'.

6 Evaluation

In order to perform an evaluation of the lexicon we prepared a corpus of general Polish language texts manually annotated with contiguous MWEs. It consists of documents extracted from the manually annotated subcorpus of the National Corpus of Polish. This subcorpus does not contain full texts but only randomly selected paragraphs thereof, and for the sake of our evaluation we chose the

[5] http://nkjp.pl/poliqarp/help/ense2.html.



```
adwokat diabła ☒ adwokat diabła:subst:sg:nom:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:nom:m1
adwokata diabła ☒ adwokat diabła:subst:sg:gen:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:gen:m1
adwokatowi diabła ☒ adwokat diabła:subst:sg:dat:m1
adwokatom diabła ☒ adwokat diabła:subst:pl:dat:m1
adwokata diabła ☒ adwokat diabła:subst:sg:acc:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:acc:m1
adwokatem diabła ☒ adwokat diabła:subst:sg:inst:m1
adwokatami diabła ☒ adwokat diabła:subst:pl:inst:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:loc:m1
adwokatach diabła ☒ adwokat diabła:subst:pl:loc:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:voc:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:voc:m1
adwokaty diabła ☒ adwokat diabła:subst:pl:nom:m2
adwokaty diabła ☒ adwokat diabła:subst:pl:voc:m2
```

Fig. 5. Inflection paradigm for the nominal MWE *czerwony pająk* (lit. *red spider*) ‘communist’.

Table 3. Contents of the evaluation corpus.

Document extracts	Tokens	Annotated MWEs					
		Occurrences					Unique forms
		Nouns	Adjectives	Adverbs	Others	All	
125	234,891	9,468	174	1,087	303	11,032	9,580

125 longest extracts of different press genres: newspapers, magazines, periodicals, popular science, etc. The annotation schema was rather simple: contiguous sequences of words judged as multiword expressions of the general Polish language were to be tagged as belonging to one of the following categories: compound noun (CN), foreign compound noun (CNF), compound adjective (CA), foreign compound adjective (CAF), compound adverb (CADV), foreign compound adverb (CADVF) or other MWE (Polish, foreign, erroneously spelled – OTH)[6]. The annotator was a native Polish speaker, expert in linguistics, neutral with respect to the project, i.e. uninvolved in the creation of the lexicon. Table 3 shows the contents of the resulting evaluation corpus. For the purpose of the evaluation, some categories were merged or eliminated so as to obtain the three final categories to which the lexicon was dedicated: nouns (CN and CNF), adjectives (CA and CAF) and adverbs (CADV and CADVF).

The evaluation results are presented in Table 4. Note that only about 10% (455 out of 4,775) of all lemmas contained in the lexicon have their inflected forms in the corpus, which confirms the sparseness issues typical for MWEs. The coverage of the evaluation corpus by the lexicon is reasonably high for adverbs (33%)

[6] Some economical sublanguage terms were also annotated but those judged as not belonging to the general Polish language were eliminated during the evaluation.

but rather low for nouns and adjectives. The total coverage attains 9%. Two main reasons may underlie this score. Firstly, the lexicon focuses mainly on the most idiomatic, semantically opaque or strongly institutionalized compounds, while the corpus annotator had a much broader understanding of a MWE and marked many relatively weakly lexicalized phrases and collocations (e.g. *wiejska droga* 'country road', *bliski śmierci* 'close to death'). Secondly, the lexicon size was delimited by the scope of the funding project and its development should clearly continue, given that similar resources for other languages easily attain several dozens of thousands of compound lemmas.

Table 4. Lexicon coverage evaluated against the corpus.

	Corpus MWEs found in the lexicon	
	Occurrences	Lemmas
Nouns	598 (6%)	353
Adjectives	7 (4%)	6
Adverbs	364 (33%)	96
All	969 (9%)	455

7 Application to Automatic Treebank Annotation

SEJF, as a high-quality grammatical resource, can be used in a variety of NLP applications. Notably, its utility for automatic treebank annotation was recently tested by [26]. The task was to project 3 available resources of Polish MWEs, including SEJF, on a Polish constituency treebank, Składnica [30], which contained no initial MWE annotations. This task is important since MWE-annotated treebanks are scarce and constitute bottlenecks in the MWE-oriented research.

The extensional version of SEJF, containing the 88,000 morphosyntactic variants of MWEs, as in Figs. 3 and 5, was used in the experiments. The SEJF entries were transformed into queries and evaluated against the treebank. As a result, the treebank subtrees containing continuous sequences of leaves corresponding to the SEJF entries, and respecting the relevant morphological constraints, were automatically marked. The automatic projection was followed by a manual validation. The SEJF-specific outcome of this process is shown in Table 5.

The true positives (TP) correspond to the MWEs from SEJF correctly identified in the treebank by the automatic projection. The most frequently repeated MWEs in this group are adverbials like *przede wszystkim* (lit. *before all*) 'mainly' (25 occ.), *na pewno* (lit. *on sure*) 'certainly' (12 occ.), *na miejscu* (lit. *on place*) 'instantaneously/relevant' (12 occ.), *po prostu* (lit. *on simple*) 'simply' (12 occ.), *od razu* (lit. *from time*) 'immediately' (9 occ.), etc. False positives (FP) are errors resulting from bugs in the mapping procedure. The compositional readings (CRead) are cases like those in examples (1)–(4), sometimes included in larger

Table 5. Results of projecting the SEJF entries on the Składnica treebank, including true positives (TP), false positives (FP), compositional readings (CRead), and idiomaticity rate (IRate).

	Nouns		Adverbs		Others		All categories	
	Occ.	Lemmas	Occ.	Lemmas	Occ.	Lemmas	Occ.	Lemmas
TP	209	154	153	67	6	4	368	225
FP	0	0	4	4	1	1	5	5
CRead	17	12	19	11	0	0	36	23
All occ.	226	165	176	78	7	3	409	248
IRate	0.92	n/a	0.89	n/a	1	n/a	0.91	n/a

MWEs, as in (3)–(4). The idiomaticity rate [8], i.e., the ratio of occurrences with idiomatic reading to all correctly recognized occurrences, is relatively high, especially for nominal MWEs. The MWE with the highest number of compositional readings is *na miejscu* (lit. *on place*) 'instantaneously/relevant' as in example (2) (6 occ.). Note that the same MWE also has a high number of idiomatic occurrences (12).

(1) ... w **drugiej połowie** XIX wieku
'... in the **second half** of the 19th century'
coinciding MWE: *druga połowa* (lit. *second half*) 'one's husband or wife'

(2) ... był **na miejscu** zdarzenia
'... he was **at the place** of the event'
coinciding MWE: *na miejscu* (lit. *on place*) 'instantaneously/relevant'

(3) ... od czasu **do czasu** zazdrościła przyjaciółkom
'... from time **to time** she envied her friends'
coinciding MWE: *do czasu* (lit. *to time*) 'temporarily'

(4) Zmiany dokonane w Oplu Fronterze wyszły mu **na dobre**
(lit. 'Changes operated in Opel Fronter went out **for the good**.')
'Changes operated in Opel Fronter turned to its advantage.
coinciding MWE: *na dobre* (lit. *for the good*) 'permanently'

These results show that SEJF can be successfully applied to automatic treebank annotation, due to the fine-grained grammatical descriptions contained in this resource, and to the high idiomaticity rate of Polish MWEs. Automatic disambiguation, i.e. distinguishing idiomatic from compositional readings, remains a challenge in cases when a MWE does not occur although it might (i.e. most morphosyntactic constraints it imposes are fulfilled). Note, however, that cases like (1)–(2) can be resolved if the MWE entry is enriched with information on its valency, i.e. its allowed or prohibited non-lexicalized modifiers. Efforts towards rich syntactic encoding of this kind have notably been undertaken in valence dictionaries with phraseological components [20], and synergies between such formalisms and SEJF-like e-dictionaries are being investigated.

8 Related Work

Although MWEs are still under-represented in language resources and tools, efforts have been put towards bridging this gap from the e-lexicographical point of view in many languages, as discussed in [15]. The community around Intex[7], NooJ[8] and Unitex has a long e-lexicographic tradition related to compounds, with dictionaries of compounds created for French [28], English [24], Greek [14] and others. Lexicons similar to SEJF, following the Multiflex paradigm, exist or are under construction for Serbian [13], Greek [9], and Macedonian [22]. Various e-lexicographic frameworks were developed for the creation of MWE e-lexicons notably in Turkish [18], Basque [2], Dutch [11], Serbian [29] and Hebrew [1], the last one also covers verbal MWEs.

On the Polish ground, SEJF is one of three grammatical lexicons of Polish multiword units built under Toposław. The two other resources are: (i) SAWA[9] [17], a grammatical lexicon of Warsaw urban proper names (streets, squares, bus stops, and other objects linked to the Warsaw communication network), (ii) SEJFEK[10] [27], a grammatical lexicon of Polish economic terminology containing over 11,000 specialized nominal compounds. Complementary formalisms for inflectional paradigms of Polish MWEs have been presented in [5,10].

9 Conclusions and Future Work

We have presented the construction of SEJF, an electronic grammatical lexicon of Polish nominal, adjectival and adverbial MWEs. It is one of the first steps towards a systematic and extensive description of such units, applicable to automatic text processing in Polish, including richly annotated corpora such as treebanks. While the coverage of compound adverbs offered by SEJF is reasonable, its contents in terms of compound nouns and adjectives should be extended, as shown by the evaluation results. Additional corpora can underlie this further work, including those available via Sketch Engine[11] with collocation support [21].

As mentioned in Sect. 5 the description of nominal MWEs in masculine human gender is not fully satisfactory with Toposław, due to the impossibility to generate the depreciative forms of these expressions. These problems can be overcome with a recent follow-up of Toposław, called Werbosław, which allows the user to gather several flexemes of the same lemma in one lexeme.

More precisely, according to [23], a lexeme is understood as an abstract unit of language containing all forms connected with the same lexical meaning. For instance, *adwokaci diabła* 'devil's advocates' in human masculine (`m1`) and *adwokaty diabła* 'devil's advocates (depr.)' in human animate (`m2`), belong to the

[7] http://intex.univ-fcomte.fr/.

[8] http://www.nooj4nlp.net/pages/nooj.html.

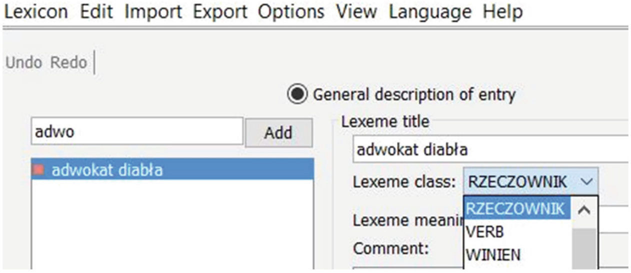
[9] http://zil.ipipan.waw.pl/SAWA.

[10] http://zil.ipipan.waw.pl/SEJFEK.

[11] https://www.sketchengine.co.uk/pltenten-corpus/.

same lexeme. A lexeme can further subdivide into several flexemes [4], i.e. morphosyntactically homogeneous sets of forms belonging to the same lexeme. Since a substantive (`subst`) is defined in the NKJP-Morfeusz tagset as a class which inflects for case an number and *has* (invariable) gender, *adwokaty/adwokaci diabła* in two different genders cannon belong to the same nominal flexeme. Thus, only the forms in `m1` are classified into the flexeme of class `subst`. The forms in `m2` are separated in another flexeme of class `depr` (depreciative form), defined as inflecting for case and having number and gender.

Toposław is flexeme-oriented, therefore these two flexemes would have to be described separately, which would be unnatural, since they both share the same lemma *adwokat diabła* 'devil' advocate'. Werbosław, conversely, is lexeme-oriented. Each of its individual entries is a lexeme whose class has to be selected by the lexicographer in the initial stage of the description, as shown in Fig. 6.



● General description of entry

adwo

adwokat diabła

Lexeme title: adwokat diabła

Lexeme class: RZECZOWNIK

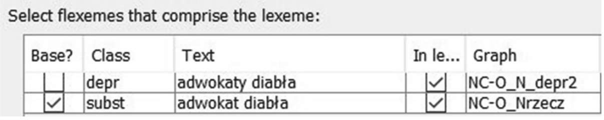
RZECZOWNIK
VERB
WINIEN

Lexeme meani

Comment:

Fig. 6. Selecting the class (RZECZOWNIK 'noun', VERB, etc.) of the lexeme *adwokat diabła* 'devil's advocate' in Werbosław.

A lexeme is a unit of a higher order as compared to a flexeme. Therefore, the next step is to define the list of flexemes associated with a given lexeme, as shown in Fig. 7.



Select flexemes that comprise the lexeme:

Base?	Class	Text	In le...	Graph
□	depr	adwokaty diabła	☑	NC-O_N_depr2
☑	subst	adwokat diabła	☑	NC-O_Nrzecz

Fig. 7. Selecting the flexemes (here: `depr` and `subst`) associated with the lexeme (here `RZECZOWNIK` 'noun') *adwokat diabła* 'devil's advocate' in Werbosław.

The description of each of the flexemes follows the same steps as in Toposław, i.e. consists of analyzing each component morphosyntactically and selecting the right inflection graph. Fig. 8 shows the graph for the depreciative flexeme of *adwokat diabła* 'devil's advocate'. Recall that the depreciative forms only show in the nominative and vocative case in plural, i.e. Morfeusz only generates these two forms for a depreciative noun. Therefore, the number in the graph can be fixed to plural (`Nb=pl`) and the case inflection can be unrestricted (`Case=$c`).

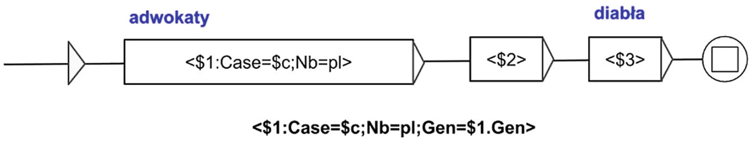
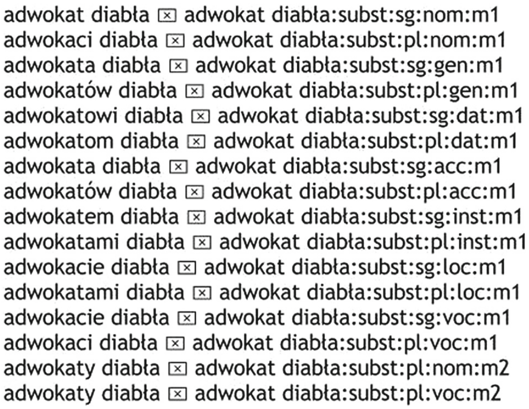


Fig. 8. Inflection graph `NC-O_N_depr2` describing the depreciative flexeme *adwokaty diabła* 'devil's advocates' of the nominal lexeme *adwokat diabła* 'devil's advocate'.

As a result, the full description of the lexeme yields an enhanced list of the inflected forms shown in Fig. 9. Note the occurrence of the two depreciative forms in the `m2` gender, as opposed to the paradigm obtained with Toposław in Fig. 3.



```
adwokat diabła ☒ adwokat diabła:subst:sg:nom:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:nom:m1
adwokata diabła ☒ adwokat diabła:subst:sg:gen:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:gen:m1
adwokatowi diabła ☒ adwokat diabła:subst:sg:dat:m1
adwokatom diabła ☒ adwokat diabła:subst:pl:dat:m1
adwokata diabła ☒ adwokat diabła:subst:sg:acc:m1
adwokatów diabła ☒ adwokat diabła:subst:pl:acc:m1
adwokatem diabła ☒ adwokat diabła:subst:sg:inst:m1
adwokatami diabła ☒ adwokat diabła:subst:pl:inst:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:loc:m1
adwokatami diabła ☒ adwokat diabła:subst:pl:loc:m1
adwokacie diabła ☒ adwokat diabła:subst:sg:voc:m1
adwokaci diabła ☒ adwokat diabła:subst:pl:voc:m1
adwokaty diabła ☒ adwokat diabła:subst:pl:nom:m2
adwokaty diabła ☒ adwokat diabła:subst:pl:voc:m2
```

Fig. 9. Inflection paradigm of the nominal lexeme *adwokat diabła* 'devil's advocate', containing regular and depreciative forms.

An even more challenging behavior is exhibited by Polish verbs, where a single lexeme consists of up to 12 different flexemes. For instance, the non-past flexemes (`fin`) like *robi* 'does' inflect for number and person, and have aspect; the past flexeme (`praet`) like *robił* 'did' inflect for number, gender and agglutination, and have aspect; the gerunds (`ger`) like `robienie` 'doing' inflect for number, case and negation, and have gender and aspect; etc. Thanks to flexeme-to-lexeme shift operated in Werbosław, verbal mutiword expressions, such as *odwracać kota ogonem* (lit. *to turn the cat with its tail to the front*) 'to distort the facts', can now be conveniently described. Such expressions are being currently addressed within the Verbel project[12] [7].

[12] http://uwm.edu.pl/verbel.

Note finally that the descriptive framework of Toposław and Werbosław does not account for more complex syntactic phenomena such as diathesis change and long-distance dependencies. Therefore, verbal MWEs can only be described from the point of view of their inflectional and word-order variants. Other syntactic variants (passivisation, nominalisation, internal modification, etc.) call for an expressive power close to full-fledged syntactic formalisms. The same applies to nominal, adjectival and adverbial MWEs with open slots, such as *[czyjaś] prawa ręka* (lit. *[someone's] right hand*) '[someone's] main assistant'. Despite these shortcomings, we hope to have shown that our proposals prove useful for the description of large classes of MWEs whose frequency in a corpus is usually rather high.

Acknowledgements. This work has been supported by three projects: (i) Nekst(http://www.ipipan.waw.pl/nekst), funded by the European Regional Development Fund and the Polish Ministry of Science and Higher Education, (ii) CESAR (http://clip.ipipan.waw.pl/CESAR) - a European project (CIP-ICT-PSP-271022), part of META-NET, (iii) IC1207 COST action PARSEME(http://www.parseme.eu).

References

1. Al-Haj, H., Itai, A., Wintner, S.: Lexical representation of multiword expressions in morphologically-complex languages. Int. J. Lexicogr. **27**(2), 130–170 (2014)
2. Alegria, I., Ansa, O., Artola, X., Ezeiza, N., Gojenola, K., Urizar, R.: Representation and treatment of multiword expressions in Basque. In: Proceedings of the ACL 2004 Workshop on Multiword Expressions, pp. 48–55 (2004)
3. Bańko, M.: Słownik porównań. Polish Scientific Publishers PWN, Warsaw (2004)
4. Bień, J.S.: Koncepcja słownikowej informacji morfologicznej i jej komputerowej weryfikacji. Rozprawy Uniwersytetu Warszawskiego 383 (1991)
5. Broda, B., Derwojedowa, M., Piasecki, M.: Recognition of structured collocations in an inflective language. In: Proceedings of the International Multiconference on Computer Science and Information Technology – 2nd International Symposium Advances in Artificial Intelligence and Applications (AAIA 2007), pp. 237–246 (2007)
6. Czerepowicka, M.: Opis powierzchniowoskładniowy wyrażeń niestandardowych typu «na lewo», «do dziś», «po trochu», «na zawsze» we współczesnym języku polskim. Akademicka Oficyna Wydawnicza EXIT, Warszawa (2006)
7. Czerepowicka, M., Kosek, I., Przybyszewski, S.: O projekcie elektronicznego słownika odmiany frazeologizmów czasownikowych. Polonica **34**, 115–123 (2014)
8. El Maarouf, I., Oakes, M.: Statistical measures for characterising MWEs. In: IC1207 COST PARSEME 5th General Meeting (2015). http://typo.uni-konstanz.de/parseme/index.php/2-general/138-admitted-posters-iasi-23-24-september-2015
9. Foufi, V.: Les noms composés A(A)N du Grec Moderne et leurs variantes. In: Kakoyianni Doa, F. (ed.) Penser Le Lexique-Grammaire : Perspectives Actuelles. Editions Honoré Champion, Paris (2013)
10. Graliński, F., Savary, A., Czerepowicka, M., Makowiecki, F.: Computational lexicography of multi-word units. How efficient can it be? In: Proceedings of the COLING-MWE 2010 Workshop, Beijing, China (2010)

11. Grégoire, N.: DuELME: a Dutch electronic lexicon of multiword expressions. Lang. Resour. Eval. **44**(1–2), 23–39 (2010)
12. Kosek, I.: Fleksja i składnia nieciągłych imiennych jednostek leksykalnych. Publishing House of the University of Warmia and Mazury, Olsztyn (2008)
13. Krstev, C., Stanković, R., Obradović, I., Vitas, D., Utvić, M.: Automatic construction of a morphological dictionary of multi-word units. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) NLP 2010. LNCS (LNAI), vol. 6233, pp. 226–237. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14770-8_26
14. Kyriacopoulou, T., Mrabti, S., Yannacopoulou, A.: Le dictionnaire électronique des noms composés en grec moderne. Lingvist. Investig. **25**(1), 7–28 (2002)
15. Losnegaard, G.S., Sangati, F., Escartín, C.P., Savary, A., Bargmann, S., Monti, J.: Parseme survey on MWE resources. In: Chair, N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, May 2016
16. Marciniak, M., Savary, A., Sikora, P., Woliński, M.: Toposław – a lexicographic framework for multi-word units. In: Vetulani, Z. (ed.) LTC 2009. LNCS (LNAI), vol. 6562, pp. 139–150. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20095-3_13
17. Marciniak, M., Rabiega-Wiśniewska, J., Savary, A., Woliński, M., Heliasz, C.: Constructing an electronic dictionary of polish urban proper names. In: Recent Advances in Intelligent Information Systems, pp. 233–246. Exit (2009)
18. Oflazer, K., Çetonoğlu, Özlem., Say, B.: Integrating morphology with multi-word expression processing in Turkish. In: Second ACL Workshop on Multiword Expressions, pp. 64–71 (2004)
19. Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.): Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw (2012)
20. Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M.: Extended phraseological information in a valence dictionary for NLP applications. In: Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014), pp. 83–91. Association for Computational Linguistics and Dublin City University, Dublin, Ireland (2014). http://www.aclweb.org/anthology/siglex.html#2014_0
21. Radziszewski, A., Kilgarriff, A., Lew, R.: Polish word sketches. In: Proceedings of the 5th Language and Technology Conference, Poznań, Poland, pp. 237–242, November 2011
22. Rafajlovska, A., Zdravkova, K.: Représentation des expressions composées en macédonien en tant qu'entrées lexicales en Unitex. In: Actes de la Traitement Automatique des Langues Slaves, pp. 1–8. Association pour le Traitement Automatique des Langues, Caen, France, June 2015. http://www.atala.org/taln_archives/TASLA/TASLA-2015/tasla-2015-court-001
23. Saloni, Z.: Klasyfikacja gramatyczna leksemów polskich. Język Polski **54**(1), 3–13 (1974)
24. Savary, A.: Recensement et description des mots composés - méthodes et applications, Ph.D. Thesis. Université de Marne-la-Vallée (2000)
25. Savary, A.: Multiflex: a multilingual finite-state tool for multi-word units. In: Maneth, S. (ed.) CIAA 2009. LNCS, vol. 5642, pp. 237–240. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02979-0_27

26. Savary, A., Waszczuk, J.: Projecting multiword expression resources on a polish treebank. In: Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, pp. 20–26. Association for Computational Linguistics, Valencia, Spain April 2017. http://www.aclweb.org/anthology/W17-1404
27. Savary, A., Zaborowski, B., Krawczyk-Wieczorek, A., Makowiecki, F.: SEJFEK - a lexicon and a shallow grammar of polish economic multi-word units. In: Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon, pp. 195–214. The COLING 2012 Organizing Committee, Mumbai, India, December 2012. http://www.aclweb.org/anthology/W12-5116
28. Silberztein, M.: Les groupes nominaux productifs et les noms composés lexicalisés. Lingvist. Investig. **17**(2), 405–425 (1993)
29. Stanković, R., Obradović, I., Krstev, C., Vitas, D.: Production of morphological dictionaries of multi-word units using a multipurpose tool. In: Proceedings of the Computational Linguistics-Applications Conference, Jachranka, Poland, pp. 77–84, October 2011
30. Świdziński, M., Woliński, M.: Towards a bank of constituent parse trees for polish. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS (LNAI), vol. 6231, pp. 197–204. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15760-8_26
31. Wojdak, P.: Przysłówki polisegmentalne w modelu składniowym polszczyzny. Publishing House of the University of Szczecin, Szczecin (2008)
32. Woliński, M.: Morfeusz - a practical tool for the morphological analysis of polish. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. AINSC, vol. 35. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-33521-8_55

Lemmatization of Multi-Word Entity Names for Polish Language Using Rules Automatically Generated Based on the Corpus Analysis

Jacek Małyszko, Witold Abramowicz, Agata Filipowska(✉), and Tomasz Wagner

Poznan Univeristy of Economics and Business,
al. Niepodległości 10, 61-875 Poznań, Poland
agata.filipowska@ue.poznan.pl

Abstract. The article concerns automatic lemmatization of Multi-Word Units for highly inflective languages. We present an approach, where the lemmatization is conducted using rules generated solely based on a corpus analysis. Conducted experiments revealed, that the accuracy of automatic lemmatization of MWUs for the Polish language according to the developed approach may reach up to 82%.

Keywords: Natural Language Processing · Multi-Word Units Lemmatization

1 Introduction

Multi-Word Units (MWUs), or Multi-Word Expressions (MWEs), are "idiosyncratic interpretations that cross word boundaries (or spaces)" [6]. Although they consist of many words (graphical units), for some application-dependent reasons they should be listed, described and processed as a single unit at some level of linguistic analysis [1,8]. One type of MWUs are multi-word entity names. MWUs pose a serious difficulty in many Natural Language Processing tasks [6]. One such difficulty is morphological analysis of such expressions, especially for languages with rich morphology, such as Slavic languages.

An example of task, which becomes difficult when dealing with MWUs, is their lemmatization. This is due to the fact, that the lemma of a MWU may contain words, which are not lemmas themselves [8]. Let's analyze a Polish multi-word entity name *Organizacji Narodów Zjednoczonych* (United Nations in genitive case). If we lemmatize each word separately and concatenate received lemmas, we obtain the following phrase: *Organizacja Naród Zjednoczyć*, which is an incorrect expression according to the grammar of Polish language (correct lemma is *Organizacja Narodów Zjednoczonych*). Therefore, trying to obtain the lemma of the phrase simply by performing lemmatization of each word separately, would result in generation of a grammatically incorrect phrase.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 74–84, 2018.
https://doi.org/10.1007/978-3-319-93782-3_6

In this paper, we analyze a problem of lemmatization of multi-word entity names for Polish language. As will be discussed in the Related Work section, a number of approaches exist towards this issue. It is commonly acknowledged that, to ensure high accuracy of the achieved results, the inflection of a phrase should be analyzed at a lexical rather than grammatical level. This usually requires a significant amount of manual work. Still, we have not found any evaluation on what accuracy can be obtained for highly inflective languages, like Polish, when the lemmatization is based only on grammatical rules, which ignore lexical information. We believe, that in some cases, such approach may be sufficient and much less labour intensive, especially when the inflection rules are automatically extracted from a corpus. Thus, the goal of this paper is to analyze what accuracy may be achieved for Polish language using only grammar-based inflection rules automatically extracted from a corpus.

The structure of this article is the following. First, in Sect. 2, we describe the problem of MWU lemmatization in greater detail. Next, in Sect. 3 a brief analysis of related work is presented. In Sects. 4 and 5 we first present a developed approach towards grammar-based MWU lemmatization and next analyze the obtained results of performed experiments. The article is concluded with a short summary.

2 Description of the Encountered Problem

We encountered the problem of multi-word names lemmatization for Polish language during our work on a search engine for legislative acts of Greater Poland Regional Assembly and Greater Poland Executive Board. We wanted to tag acts with names of entities, which were mentioned in the titles of acts. In many cases, some multi-word names were mentioned in the titles, usually in an inflected form. The entity names could be easily extracted, because each word in these names started with a capital letter. Having these names, we wanted to present users with tags representing these entities; if a user would click such tag, he would be presented with a list of all acts, in which this entity was mentioned in the title.

Two problems resulting from the inflection of multi-word entity names arise here:

1. linking differently inflected forms of the same names together,
2. presenting the users with lemmatized forms of the entity names.

The first problem can be solved using some text normalization techniques and string similarity measures, such as Levenshtein distance. Still, the other one poses a greater challenge, because, as was discussed in the Introduction, simple lemmatization of each constituent separately will usually result in a grammatically incorrect phrase and not the lemma of the MWU. There are three main types of decisions, which must be made to correctly generate a lemma for a given MWU:

1. Which words from the MWU should be inflected; in different MWUs a different number of words is being inflected. For example, having a three words

Table 1. Exemplary three-words long MWUs (in English these are Substance Abuse Treatment Facility, Regional Innovation Strategy and Poznań International Fair). In each of them, a different number of words must be inflected to produce a lemma from the inflected form. Below the phrases, their POS tags (using NKJP tagset) are presented

Inflected form $POS_{p',infl}$	Lemma $POS_{p,lemma}$
Zakładu Leczenia Uzależnień	Zakładu Leczenia Uzależnień

`subst:sg:gen:m3, subst:sg:gen:n,`
`subst:pl:gen:n`
	`subst:sg:nom:m3,subst:sg:gen:n,`
	`subst:pl:gen:n`

Regionalną Strategią Innowacji	Regionalna Strategia Innowacji

`adj:sg:acc:f:pos, subst:sg:inst:f,`
`subst:sg:gen:f`
	`adj:sg:nom:f:pos, subst:sg:nom:f,`
	`subst:sg:gen:f`

Międzynarodowych Targach Poznańskich	Międzynarodowe Targi Poznańskie

`adj:pl:gen:m3:pos, subst:pl:loc:m3,`
`adj:pl:gen:m3:pos`
	`adj:sg:nom:n:pos,`
	`subst:pl:nom:m3, adj:sg:nom:n:pos`

long phrase, in some cases its inflection may require that only one word must be inflected, while in other cases two or even all three word must be inflected (see Table 1).

2. If a given word is to be inflected, in the next step we must determine which form of a given word should be chosen, e.g. grammatical case, number and gender must be determined.
3. For some languages, inflection may change the order of constituents in the MWU [8,9]; still, for Polish language, this is generally not the case and we will skip this type of decisions in our work.

3 Related Work

Inflection of Multi Word Units is a well-established problem in Natural Language Processing [1]. Among others, it is often encountered when developing electronic dictionaries. Lemmatization of a phrase is one of the most important steps in this task [9]. A list of all inflected forms of a phrase, together with their inflectional description, is called an inflectional paradigm [8] and generation of such paradigm was the goal of a number of previous research.

3.1 Rule-Based Inflection of MWUs

A basic requirement, which has to be met to enable automatic inflection of MWUs, is acquisition of a comprehensive inflection module or an inflectional dictionary for single words, which are constituents of MWUs [9]. For Polish language, PoliMorf, an open morphological dictionary for Polish may be used for this purpose [10]. Still, lemmatization of single words is much more difficult when proper names are concerned, for example person names [4].

It is generally acknowledged that a high accuracy of automatic inflection of MWUs may be achieved only when lexical information is taken into account. In other words, inflection rules must be assigned on a per-phrase basis by the lexicon engineer, which is a labour intensive task [1]. A survey of such lexical approaches to the inflection of MWUs was published in [8].

An exemplary lexicalized approach towards inflection of MWUs was *Multiflex*, proposed in paper [7]. In this approach, to each phrase a so-called inflection graph is assigned, which is used to describe the inflectional behavior of a given MWU. The inflection graph is directed and acyclic and each node in it represents a single, possibly inflected, constituent. Each path in such graph corresponds to one or more inflected forms of a whole MWU. There may be many nodes corresponding to a single word in one graph and in each node there is information whether a given constituent should be inflected and, if so, how. A set of restrictions can be put on constituents, for example ensuring the agreement between specific attributes of several constituents, e.g. a grammatical case.

Similar approach was presented in paper [9]. In this paper, a tool designed to help linguists in developing, maintaining and exploiting e-dictionaries was presented, called LeXimir. LeXimir uses a set of rules manually produced by the expert, which deduce the basic structure of a given MWU, as well as its additional features. For each phrase, the software offers several lemmas (with assigned inflection rules), from which the user has to choose the correct one [9].

To summarize, there exists a number of approaches towards automatic inflection of MWUs, as well as. Analyzed approaches allow high accuracy, but require a lot of manual work to create inflection rules and assign them to individual phrases to subsequently allow conducting of automatic inflection. What is also worth mentioning is that the described approaches are, generally speaking, directed at lexicon construction task.

3.2 Wikipedia-Based Mappings for Lemmatization of Multi-Word Entity Names

An important resource for lemmatization-related data is Wikipedia. Due to its vast size and semi-structured contents, it is possible to automatically (or semi-automatically) obtain inflected forms of MWUs mapped to their lemmas. This can be done, for example, based on analysis of inter-wiki links, that is links in the content of a certain Wikipedia article mapping to another Wikipedia article. Usually, title of article on Wikipedia is a lemma and an anchor text of link in the contents often is in inflected form (depending on the context, in which it appears in the text) of a certain word or MWU. Such links, their targets and anchor texts may be extracted automatically from the HTML contents of the Wikipedia article or based on analysis of dump of Wikipedia database.

Still, in many cases, such mappings will consist not only of lemma-inflected form pairs. Many different words or phrases in the text may be used as anchor texts of links, not only inflected forms of the name of the given article. For example, many links on Polish Wikipedia pointing to Poznan University of Economics (Uniwersytet Ekonomiczny w Poznaniu) as their anchor texts have inflected form one of older names of the institution (Akademia Ekonomiczna w Poznaniu or Wyższa Szkoła Ekonomiczna). Such mappings also may be useful in some scenarios (we've used them for identification and disambiguation of maritime-related entity names in article [2]), but they cannot be used for lemmatization purposes. Some kind of filtering must be therefore conducted, in order to select only the correct mappings and, at the same time, do not erroneously reject correct mappings.

For Polish language, a resource containing Wikipedia-based lemmatization mappings was released as part of CLARIN project, called NeLexicon. Version 2.7, which was available as of time of writing this article, contained 143301 such mappings for many different types of entities, such as persons, organizations, locations etc. This resource was utilized for lemmatization for example in paper [3]. The described approach may be extremely useful for lemmatization of MWUs doe to its ease of use, but it is limited only to those mappings, which were found and correctly extracted from Wikipedia. Usually this means, that this resource cannot be used for lemmatization of names of less-known entities (e.g. organizations or persons), which did not have their own Wikipedia articles or which are not liked to from other articles. Also, in our case, names of departments of some institutions occur frequently in the analyzed dataset, and such types of entities are represented on Wikipedia even less frequently.

4 Proposed Approach

In our work, we decided to try to automatically retrieve a list of lemmatization rules based on a corpus analysis. The quality of such rules will be worse than of those prepared by the expert. Still, the accuracy of lemma identification performed this way may be sufficient for some tasks and it is much less labour intensive. Also, we did not find any evaluation on how such approach may work for morphology-rich languages like Polish and we hope to fill this gap with the method described below.

4.1 Available Corpus and Data Preparation

As was stated in Sect. 2, we were processing legislative acts of Greater Poland Regional Assembly and Greater Poland Executive Board. In the corpus, there were in total 5172 documents. From titles of these acts, using regular expressions, we extracted 3932 multi-word units, in which there were 942 unique phrases. The acts were well formatted and in most cases, phrases from the titles, in which several consecutive words were capitalized, were entity names (we extracted only MWUs at least three words long). The extracted entity names in many cases were inflected, but some of them were in their base form.

For each phrase, at the beginning we were determining if it is a lemma or some inflected form. We did that using a simple heuristic: if the first word of the MWU was in nominative case, we considered the phrase to be in its base form. Otherwise, the phrase was classified as inflected. For that purpose, we were using

WCRFT [5], a morpho-syntactic tagger for Polish language. We found, that such approach allowed us to identify MWUs in lemma forms with accuracy above 95%.

Identification of MWUs, which already are lemmas, immediately gave us two benefits. Firstly, obviously,we did not have to process lemmas anymore. Moreover, having a lemma of a phrase, we could search trough all extracted MWUs to find inflected forms of the same phrase. Thus, we would identify other phrases, for which we know their lemma.

To identify other MWUs, which are inflected forms of a given lemma, we were generating simplified forms of phrases, where as simplified form of a phrase we understand a form, where all words from that phrase were lemmatized separately and then concatenated. For lemmatization of single words, we used Hunspell tool[1]. An example of such simplified form was already given in the Introduction; for phrases *Organizacja Narodów Zjednoczonych* and *Organizacji Narodów Zjednoczonych*, the simplified form is *Organizacja Naród Zjednoczyć*. If both phrases had the same simplified form (as is the case in the presented example), we assumed, that they differ only because of the inflection. Thus, we could identify, that a lemma for a phrase *Organizacji Narodów Zjednoczonych* is *Organizacja Narodów Zjednoczonych* (because the first word of the latter phrase is in nominative case). We will refer to such identified pairs of phrases as (*lemma, inflected form*) pairs.

Analyzing phrases from titles of acts we found 67 such (*lemma, inflected form*) pairs. To find additional pairs, we searched trough whole documents (not only the titles) to find phrases with the same simplified form as some of MWUs extracted from the titles. We found in total 634 different (*lemma, inflected form*) pairs. Still, for 433 MWUs we did not find any corresponding lemma. For these MWUs, their lemmas had to be generated automatically.

4.2 Generation of Lemmatization Rules

As was stated, after some data preparation steps, we were identifying (*lemma, inflected form*) pairs in the corpus. For each phrase, we also had POS tags sequences, generated using WCRFT tagger. Thus, by analyzing tags sequences in such pairs we could identify, how POS tags sequences tend to change, when a phrase with a certain tag sequence is lemmatized. We will denote POS tags sequence for a phrase p for its inflected form as $POS_{p,infl}$, and for its base form as $POS_{p,lemma}$. Having such pairs of POS tags sequences, we were automatically generating four types of lemmatization rules, which are described below.

Each rule consists of two sides: a Left Hand Side (LHS) and a Right Hand Side (RHS), separated from each other with → sign. Each side of the rule is a sequence of tags. LSH was used to match a given phrase to a specific rule; that is, having an inflected phrase p' and its POS tags sequence $POS_{p',infl}$, we were comparing it with LHSs of all rules to find a match. If a match was found, the matched rule was applied to p', that is the constituents of the phrase were inflected as was stated on the RHS the rule.

[1] http://hunspell.sourceforge.net/.

Complete Rules. In this type of rules, we take POS tags sequences from lemma - inflected form pairs and consider those as lemmatization rules as shown on Eq. 1. Examples of such lemmatization rules are presented in Table 1. In each row of this table, below phrases, there are POS tags sequences $POS_{p,infl}$ in the first and $POS_{p,lemma}$ in the second column. Using such rules, for each phrase p', for which we do not know its lemma, we retrieve its POS tags sequence $POS_{p',infl}$ and we search through all complete rules for a rule, in which $POS_{p',infl}$ was equal to its LHS. We were assuming, that in such case, if we inflect the words in the MWU according to the RHS of the rule, we will receive a correct lemma for that phrase.

$$POS_{p,infl} \rightarrow POS_{p,lemma} \tag{1}$$

An example of application of this type of rule is the following. Lets assume, that we have the following inflected MWU: p' =*Miejskim Programem Rewitalizacji* (Urban Renewal Programme in genitive). Its POS tags sequence $POS_{p',infl}$ is exactly the same as for phrase *Regionalną Strategią Innowacji* in Table 1. A rule generated based on the second row in Table 1 would therefore have a LHS matching to $POS_{p',infl}$. Thus, the lemma for p' is generated based on the RHS of the rule, that is using tags from the lemma column of the same row in Table 1. For example, first word of p' (*Miejskim*) should be inflected to `adj:sg:nom:f:pos`. By inflecting all words from p' according to RHS, we receive phrase *Miejski Program Rewitalizacji*, which is a correct lemma for p'.

Partial Rules. Partial rules differ from complete rules in that, having a phrase p', for which we want to get its lemma, we go trough all $(POS_{p,infl}, POS_{p,lemma})$ pairs and we try to find the longest match between subsequences of $POS_{p',infl}$ and $POS_{p,infl}$, where such subsequences always start from the beginning of the sequence. If we denote subsequence starting at tag with index t_1 and ending at t_2 as $POS_{p,infl}[t_1, t_2]$, we look for the pair, in which $POS_{p,infl}[1, t_2] = POS_{p',infl}[1, t_2]$ and t_2 has the highest value. Then, we create a RHS of the rule as a concatenation of two sequences: subsequence of $POS_{p,lemma}$ ending at t_2 and subsequence of the $POS_{p',infl}$, starting at index $t_2 + 1$ and reaching the end of the sequence, as shown on Eq. 2. Please note, that $POS_{p,infl}$ and $POS_{p',infl}$ may have a different lenght (that is, the phrase being inflected may have a different number of words comparing to the phrase, which was used to generate the rule).

$$POS_{p,infl}[1, t_2] \rightarrow POS_{p,lemma}[1, t_2] + + POS_{p',infl}[t_2 + 1, ...] \tag{2}$$

Such rules are based on the fact, that in Polish language, when we inflect MWUs, in many cases some number of words at the end of the unit remain unchanged. This was shown in Table 1, where in the first row two final word, and in second row one final word, remained unchanged. Thus we assume, that in many cases we may skip the analysis of some number of POS tags at the end of the sequence and still get the proper lemma. On the other hand, it is unlikely, that inflection of the MWU will change some words at the end, without affecting the ones at the beginning.

Caseless Complete Rules. In this type of rules, having a $POS_{p',infl}$, that is a sequence of tags for phrase p', for which we wanted to generate a lemma, we were analyzing lemma - inflected form pairs in search for a pair $(POS_{p,infl}, POS_{p,lemma})$, in which for $POS_{p,infl}$ all tags were the same as in $POS_{p',infl}$ apart from the grammatical case. We assume here, that grammatical cases of words in these phrases are different only because phrases p' and p were used in the text in a different case and if they would be used in the same case, then $POS_{p,infl}$ and $POS_{p‘,infl}$ would be identical. The described type of rules is therefore identical to Complete Rules apart from the fact, that we ignore the information about the grammatical case on the LHS of the rule.

Caseless Partial Rules. This type of rules is a variant of Partial rules, in which information about grammatical cases on the LHS of the rule is ignored. For each $POS_{p',infl}$, that is a sequence of tags for phrase p', for which we wanted to generate a lemma, we were analyzing $(POS_{p,infl}, POS_{p,lemma})$ pairs in search of the longest match between subsequences of $POS_{p',infl}$ and $POS_{p,infl}$, while in both sequences we were ignoring information about the grammatical case.

4.3 Generation of Lemmas

Having some phrase in an inflected form, we were obtaining its lemma in the following manner. First, we were searching trough all lemmas found in the corpus to check, if the lemma of that phrase was found somewhere in the corpus. If the lemma was not found, we were applying rules described in the previous section in a cascade manner, in the same order as they were described above. Such order was set to ensure that rules, which we assumed would produce better results, were applied before the less reliable ones. If we found basis to apply rule of a certain type, we were generating the lemma for the phrase using a selected rule and we were ignoring rules of the subsequent types. In some cases, perhaps the analyzed phrase could match LHSs of two different rules of the same type; in such situation, we were choosing the rule to be applied randomly.

If we decided, that a certain rule should be applied, based on its RHS we knew, how words in the phrase should be inflected. For the inflection of single words, we used PoliMorf [10] dictionary.

5 Evaluation

We performed an experiment, in which we wanted to assess what accuracy of lemma identification may be achieved for the described approach. In the experiment, we were identifying lemmas for all MWUs identified as being inflected, according to the procedure described in Subsect. 4.3. Using the described approach, we were trying to identify lemmas for 1067 inflected MWUs extracted from the corpus.

The evaluation of accuracy of lemmatization was performed manually. A human annotator (a native speaker of Polish language) was presented with pairs, each consisting of an inflected phrase and a lemma generated (or identified) for that phrase. The annotator was to assign to each pair two annotations:

Table 2. Accuracy of automatic lemmatization of MWUs using different lemmatization rules and a percentage of correctly extracted phrases among all that phrases were lemmatized using a given lemmatization rule type

Rule type	Accuracy for		% of processable MWUs	# of phrases lemmatized
	All phrases	Processable ph.		
Lemma in corpus	.9328	.9407	99.16	634
Complete	.8182	.8421	86.36	22
Partial	.6852	.9167	66.67	150
Caseless complete	.6153	.875	61.54	26
Caseless partial	.5472	.6545	51.89	146
Not lemmatized	.0	.0	58.43	89
Total	**.7573**	**.8214**	**83,54**	**1063**

- annotation stating if the lemma for a given phrase is correct,
- annotation stating whether the phrase is processable; by processable we understand phrases which:
  - are correctly extracted, i.e. span across the whole entity name; incorrectly extracted phrases are for example phrases missing some words from the entity name (for example the first or the last word),
  - contain only words, that may be inflected using the available dictionary; many phrases may contain non-Polish words or some proper names, which are impossible to be correctly lemmatized without appropriate dictionaries; we decided to annotate such phrases as unprocessable.

The results of the annotation are presented in Table 2. There are four columns with statistics in the table. In column "accuracy for all phrases" we put accuracy for all phrases, regardless whether they were annotated as processable or not. In column "accuracy for processable phrases" we did not take into account phrases annotated as unprocessable. In the third column, we put information about the percentage of MWUs lemmatized using a given rule type, which were annotated as processable. Finally, in the last column, there is information about how many phrases were lemmatized using a given lemmatization rule type.

The total accuracy of the proposed approach, when only processable phrases are concerned, was above 82%. When taking all phrases into account (also those incorrectly extracted ones or MWUs containing words, which we were not able to inflect) the result was around 76%. For most of the inflected phrases (634 out of 1063), using the proposed approach, the lemma could be found in the corpus. In such case, more than 94% of lemmas were assigned correctly.

For the remaining inflected MWUs, lemmas had to be generated automatically using rules described in Subsect. 4.2 The accuracy of lemma generation for phrases annotated as processable was generally between 84% up to 92%, except for caseless partial rules, which performed much worse than the other types of rules. To some extent, this is probably due to the fact, that rules of this type were executed only when no other rule could lemmatize a given phrase. Because of that, the rules of this type were dealing with the most difficult MWUs. For 89 phrases, we were not able to generate the lemma using the developed approach at all (none of the generated rules was matching POS tags sequences of these phrases).

We have also experimented with lemmatizing MWUs in our dataset with Wikipedia-based mappings, which were described in Sect. 3.2. We have searched the NeLexicon mappings dataset for ocurrences of inflected entity names from our dataset. We have found a match in only 12 cases, out of 934 searched phrases (only prases assessed to be correctly extracted from the corpus were used). Still, all 12 lemmas obtained for these phrases from NeLexicon mappings were correct. This results correspond to our expectations, which were described in Sect. 3.2; Wikipedia mappings allow to obtain correct lemmas, but, in case of our corpus, these mappings are applicable to only a very limited number of inflected MWUs. An interesting utilization of NeLexicon dataset would be to use the provided mappings as additional examples for learning of lemmatization rules, as described in Sect. 4

6 Summary

In this paper, we presented an approach towards automatic lemmatization of Multi-Word Units for Polish language and an evaluation of lemmatization accuracy, which may be obtained using the proposed approach. The presented method utilizes rules automatically generated based on the corpus analysis. Conducted experiments revealed, that the accuracy of automatic lemmatization of MWUs for the Polish language may reach up to 82%. We believe, that such results prove, that the automatic lemmatization of MWUs may be used for some tasks. When high accuracy is a crucial factor, the proposed method may be followed by an additional step of verification by a human expert. In such case, the amount of manual work by the expert would be highly reduced comparing to situation, when he would have to assign lemmas to all phrases without any aid of a computer system.

References

1. Handl, J.: Computational inflection of contiguous multi-word units with JSLIM. Conf. Intell. Inf. Syst. **2013**, 113–126 (2013)
2. Małyszko, J., Abramowicz, W., Stróżyna, M.: Named entity disambiguation for maritime-related data retrieved from heterogenous sources. TransNav: Int. J. Mar. Navig. Saf. Sea Transp. **10**(3), 465–477 (2016)
3. Marcińczuk, M., Kocoń, J., Oleksy, M.: Liner2 - a generic framework for named entity recognition. In: Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing, Valencia, Spain, April 2017

4. Piskorski, J., Sydow, M., Kupść, A.: Lemmatization of Polish Person Names. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies, ACL 2007, pp. 27–34. Association for Computational Linguistics, Stroudsburg (2007). http://dl.acm.org/citation.cfm?id=1567545.1567551
5. Radziszewski, A.: A Tiered CRF Tagger for Polish. In: Bembenik, R., Skonieczny, L., Rybinski, H., Kryszkiewicz, M., Niezgodka, M. (eds.) Intelligent Tools for Building a Scientific Information Platform. SCI, vol. 467. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35647-6_16
6. Sag, I.A., Baldwin, T., Bond, F., Copestake, A., Flickinger, D.: Multiword expressions: a pain in the neck for NLP. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 1–15. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45715-1_1
7. Savary, A.: A formalism for the computational morphology of multi-word units. Arch. Control Sci. **15**(3), 437 (2005)
8. Savary, A.: Computational inflection of multi-word units, a contrastive study of lexical approaches. Linguist. Issues Lang. Tech. **1–2**, 1–53 (2008)
9. Stankovic, R., Obradovic, I., Krstev, C., Vitas, D.: Production of morphological dictionaries of multi-word units using a multipurpose tool. In: Proceedings of the Computational Linguistics-Applications Conference, Jachranka, Poland, 17–19 October 2011, pp. 77–84. Polish Information Processing Society (2011)
10. Woliński, M., Miłkowski, M., Ogrodniczuk, M., Przepiórkowski, A.: PoliMorf: a (not so) New Open Morphological Dictionary for Polish. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey. European Language Resources Association (ELRA), May 2012

Parsing

Parsing of Polish in Graph Database Environment

Jan Posiadała, Hubert Czaja(✉), Eliza Szczechla, and Paweł Susicki

Scott Tiger S.A., 15 Kolektorska Street, Warsaw, Poland
{janek, czajah, eliza, trzeci}@tiger.com.pl

Abstract. This paper describes the basic concepts and features of the Langusta system. Langusta is a natural language processing environment embedded in a graph database. The paper presents a rule-based syntactic parsing system for the Polish language using various linguistic resources, including those containing semantic information. The advantages of this approach are directly related to the deployment of the graph paradigm, in particular to the assumption, that rules describing the syntax of the Polish language are valid queries in a graph database query language (Cypher).

Keywords: NLP · Graph databases · Cypher · Deep parsing · Corpus analysis Written corpora · Stand-off annotation

1 Introduction

A number of papers have been published discussing various aspect of the synergy that exists between graph theory methods and the problems in natural language processing (NLP). Many of them (Ide and Suderman 2007; Pęzik 2013) focus on the issue of the multi-source and multi-layer annotation of natural language corpora. Application of the graph model to the problem of structuring linguistic information results in the abandon of inline annotations for more clear and flexible standoff annotations (Zeldes et al. 2009) without impact on annotation semantic.

All the publications mentioned above, along with this paper, emphasise the high level of generality of the graph model as well as the multiplicity and maturity of tools and algorithms used in the graph theory (Mihalcea and Radev 2011). Recent years have brought, along with the development of the NoSQL movement (Strauch 2011), a significant growth in the field of the database systems implementing the graph paradigm such as Neo4j[1], OrientDB[2] or Apache TinkerPop[3].

This area of research has evolved from theoretical models, providing simple and elegant solutions for the basic NLP problems such as morphosyntactic annotation and modelling of the word-sense ambiguity and semantic role labelling, towards those

[1] http://neo4j.com/.

[2] http://orientdb.com/orientdb/.

[3] Apache TinkerPop Project is most known for providing a set of interfaces that graph databases that database vendors can implement (Blueprints) to get all the features of the rest of the TinkerPop stack (Pipes, Gremlin, Frames, Rexster, Furnace) where each part of the stack provides a specific function in supporting graph−based application development; http://tinkerpop.apache.org/.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 87–98, 2018.
https://doi.org/10.1007/978-3-319-93782-3_7

solutions that lie on the borderline with the artificial intelligence domain, such as information extraction and QA (question & answer) systems support.

A persistent corpus representation featuring an underlying graph model and a high structural openness has resulted in a change in the language processing paradigm. The classic, pipeline based approach is usually implemented as a set of programs performing subsequent stages of linguistic processing (Graliński et al. 2012; Shi et al. 2014) of a text represented in one of the standard markup formats (Przepiórkowski and Bański 2009). The graph based representation sees this method replaced by a corpus-centric model of gradual enrichment of the graph representation by adding new layers of linguistic annotation. The solution has all the characteristics of the standoff annotation and emphasizes its advantages (Dipper 2005). Also, with this data representation, the deployment of a rule-based system in the architecture of the text processing components often proves advantageous (Negnevitsky 2001). We will show how this potential has been exploited in the Langusta system.

The approximate location of the used parsing method in the theoretical background will be facilitated when we notice that the main mechanism to calculate the result of rules applying is a pattern-matching, a mechanism which is less general and powerful then unification. Consequently, the presented method should be regarded as a rule-based parsing technique performed in a propitious executive environment. Therefore, the presented solution is not an implementation of formalism for unification grammar, nor a proposal for a new kind of formalism, in the type of Tree Adjoining Grammar (Joshi and Schabes 1997).

The authors of this paper are aware of the abundance of context discussed above. Therefore, in our work we concentrate on the application of the graph model to deep syntactic parsing – a key issue in NLP (Szpakowicz 1978; Świdziński 1992).

1.1 Langusta

Assumptions and Inspirations. Taking up the challenge of building a language processing environment was inspired by the need to build a rule-driven text processing system for the Polish language. One important design goal was to enable syntactic parsing of the Polish language without restrictions on parsing depth.

Our motivation has been strengthened by the analysis of the advantages and limitations of the SPEJD (Przepiórkowski 2008; Buczyński and Przepiórkowski 2008) system. SPEJD is a shallow parsing system for the Polish language, created by Linguistic Engineering Group in PAS and distributed under GNU General Public License.

Considering the identified needs, the inspiring capabilities of the SPEJD system include:

- clear and conceptually sophisticated formalism available for defining parsing rules (syntactic and semantic head, left and right surrounding)
- parser to tokenizer integration
- parser to morphosyntactic analyser integration and an extensible morphosyntactic tagset

Later, in the context of the identified needs, the inspiring limitations to the SPEJD system include:

- the inability to model the ambiguity of the results of the parsing process
- lack of an open, non-volatile representation of the final and intermediate results
- the inability to easily integrate with external language resources

Because of the profound inspiration taken from the SPEJD program, some of the examples of the syntactic parser rules quoted in this paper will be compared to the corresponding SPEJD formalism rules.

2 Data Model

The graph data model is derived directly from the concept of a graph used in graph theory (Wilson 1996). Langusta uses a directed property graph as its data model, adopting one of the most popular approaches to graph-based data modelling.

In this model, the basic concepts are:

- node - with labels and attributes,
- directed edge - with a label and attributes,
- path - a finite sequence of edges which connect a sequence of nodes.

The implementation supports simple Java compliant[4] types of attributes: boolean, byte, short, int, long, float, double, char, String and also heterogeneous arrays of values of those types. Among the numerous advantages of this model, structural capacity seems to be the most important in the foreseen applications. This strength offers a prospect of facilitating the incorporation and analysis of new resources within the designed system. This expectation can be further justified by the argument that the graph model is native to many linguistic resources e.g. plWordNet (Maziarz et al. 2012).

2.1 Query Language

The data model described above enables the design and implementation of a query language. One example of such a language is Cypher (Robinson et al. 2013), originally implemented as query language for the Neo4j graph database. Cypher is a declarative, pattern-matching, graph model compliant query language for a graph database.

The choice of a query language as a declarative way to access the data was an important decision, due to the fact, that in a sense, Langusta is intended as a data analysis system. Taking this into account and also the fact that the graph model is a paradigmatic model, there were an inclination to choose the native language to the property graph model - both in semantic and syntactic sense. The assumption that the author of the rules - but also the data analyst - will be a linguistic expert created intention to preserve intuitive character of graph model in the processing of (and in the access to) data. That meant the rejection of query languages which syntax or

[4] Java types description: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html.

computation model is derived from SQL, RDF or logic programming (Wood 2009). An additional advantage of Cypher was the syntactical simplicity of manipulating of the paths and node's properties. The rule-based clause order in Cypher queries is not without significance, and increases resemblance to the SPEJD formalism.

A cypher language query is composed of the three basic clauses:

MATCH Clause: The MATCH clause is the core element of a Cypher query. In this clause we describe the matching criteria for the sought subgraph. The primary way of setting out criteria for the subgraph is describing the structure of nodes connected by edges and tagged by labels.

```
MATCH
(p:PERSON)-[r:KNOWS]->(pp:PERSON)
RETURN
p, r, pp
```

Identifiers used in the MATCH clause to name nodes, edges and paths are bound with the corresponding matched objects in the database (nodes, edges, paths).

WHERE Clause: The WHERE clause contains a boolean expression that filters objects sought in MATCH clause:

```
MATCH
(p:PERSON)-[r:KNOWS]->(pp:PERSON)
WHERE
p.age > pp.age
RETURN
p, r, pp
```

RETURN Clause: The RETURN clause contains expressions returned as the result of a query for the subgraph meeting the selection (match and filter) criteria.

3 Parsing of Polish

3.1 Tokenization

Text tokenization implementation in Langusta does not go beyond the basic definition, i.e., its result is splitting text into tokens (words) and sentences (Mazur 2005). In particular, the method developed for the PWN Corpus of Polish has been implemented (Rudolf and Swidzinski 2004). In implementation there was no distinction between the layer of tokens and multi-token words, no less, the data model used in Langusta has a sufficient capacity to make the separation between tokens layer and words layer, in order to model the ambiguity of multi-token words.

3.2 Morphosyntactic Analysis

The morphosyntactic annotation is based on the morphosyntactic dictionary PoliMorf (Woliński et al. 2012). As a result of the process, a text structure is formed in the graph database. In this structure for each token there is a corresponding set of nodes representing a collection of interpretations from the morphosyntactic dictionary for which the inflected form is equal to the token (ignoring case). These nodes are labelled with the label Word. Each Word node has appropriate values of its grammatical class and its grammatical categories stored in its attributes.

The order of tokens in a sentence is represented by the relationship: follows. The relationship occurs between any two nodes Word representing consecutive tokens from the processed sentence. The process yields the following graph structure for the sentence: *"Młode dziewczyny biegły."*[5]. The sentence is tokenized to: "Młode", "dziewczyny", "biegły", ".".

Morphosyntactic Dictionary Compression. Langusta works with the content of the morphosyntactic dictionary in a compressed format. For this purpose the system uses a shortened representation of the grammatical interpretation. The shortened notation (Woliński and Przepiórkowski 2001) is a widely used method, because of the Polish language system syncretism. In Langusta the shortened notation mentioned above is used as the compression method.

This has resulted in a significant reduction of the number of nodes representing tokens and their grammatical interpretations which will undergo further rule-based processing.

For this purpose, the content of the dictionary PoliMorf was transformed to an atomized form, i.e. each entry containing the alternative values for a grammatical category has been split into atomic entries containing unambiguous values. Next, for all the positions sharing a common inflected form and a common basic form, the set of atomized grammatical interpretations has been compressed using the shortened notation. The sum of Cartesian products of compressed entries equals the original set containing atomized entries.

Let us consider selected dictionary entries corresponding the inflected form: *"młode"*.

[5] Translation to English: "Young girls run.".

młode młoda subst:pl:voc:f
młode młoda subst:pl:nom:f
młode młoda subst:pl:acc:f
młode młode subst:pl:voc:n2
młode młode subst:pl:nom:n2
młode młode subst:pl:acc:n2
młode młode subst:sg:acc:n2
młode młode subst:sg:voc:n2
młode młode subst:sg:nom:n2
młode młode depr:pl:voc:m2
młode młody depr:pl:nom:m2
młode młody adj:sg:nom.voc:n1.n2:pos
młode młody adj:sg:acc:n1.n2:pos
młode młody adj:pl:acc:m2.m3.f.n1.n2.p2.p3:pos
młode młody adj:pl:nom.voc:m2.m3.f.n1.n2.p2.p3:pos

The above shortened notation expands to 38 atomized dictionary entries with non-empty values for grammatical class (subst, depr, adj) and non-empty values for grammatical case, number and gender. As the result of compression we get 5 entries (see Fig. 1):

młode młoda subst:.pl:nom.acc.voc:f
młode młode subst:sg.pl:nom.acc.voc:n2
młode młody adj:pl:nom.acc.voc:m2.m3.f.n1.n2.p2.p3
młode młody depr.adj:pl:nom.voc:m2
młode młody adj:sg.pl:nom.acc.voc:n1.n2

The direct consequence of compression of the morphosyntactic dictionary is a change to the types of attribute values in nodes representing data derived from that dictionary, i.e. string attributes become string array attributes.

3.3 Parsing Rules

The Langusta rules performing the syntactic analysis are valid Cypher queries. Let us consider the parsing rule used for the phrase “młode dziewczyny”. The core of the rule looks as follows:

```
MATCH[6]
(adj)--(subst)
WHERE
subst.pos *= ['ger','pact', 'ppas', 'subst']
```

[6] The list intersection operator *= is not supported by the implementation of Cypher in the Neo4j database. The interpretation is: false if and only if the list is empty.

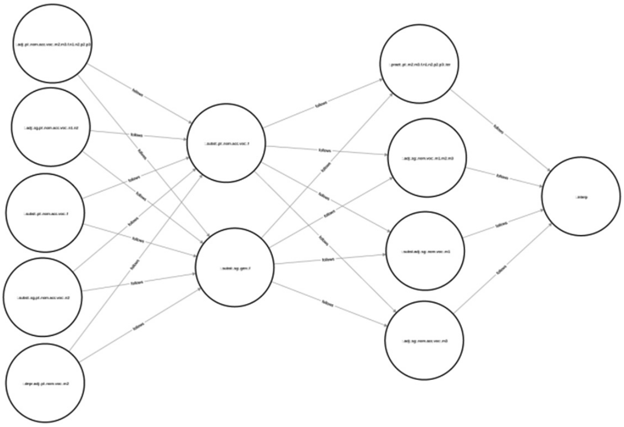


Fig. 1. Structure of sentence: "*Młode dziewczyny biegły.*" with compressed morphosyntactic interpretation in nodes caption.

```
and 'adj' in adj.pos
and adj.gender *= subst.gender
and adj.number *= subst.number
and adj.case *= subst.case
```

The corresponding SPEJD rule (Przepiórkowski and Buczyński 2007) accurate to a set of tags and set of syntactic groups used in the National Corpus of Polish (Przepiórkowski et al. 2012) is as follows[7,8]:

```
Match:([pos~"Adj|Pact|Ppas"]| [type="AdjG|AdjGk"])
([pos~"Noun"] | [type="NGg|NGs|NGb"]);
Eval: unify(case number gender,1,2);
group(NGa,2,2);
```

[7] Correspondence between WHERE expression in Langusta rule and unify operator in SPEJD rule is limited to condition component of unify operator. Application of Langusta rule rejects no interpretation.

[8] Correspondence between semantic of group action in SPEJD rule and consequence of Langusta rule application seems to be very strong, obviously excluding capability of ambiguity representation.

The full version of the rule in Langusta is as follows:

```
WITH
'R.B.Subst.01' as code,
100 as rate ,
90 as InversionRate ,
['znajomy kolega', 'znajomy krzywdzący', 'znajomy
pokrzywdzony', 'znane zagranie'] as examples
MATCH
(adj)--(subst)
WHERE
subst.pos *= ['ger', 'pact', 'ppas', 'subst'] and
'adj' in adj.pos and
adj.gender *= subst.gender and
adj.number *= subst.number and
adj.case *= subst.case
RETURN subst as synh
```

In the WITH clause, in which the computational environment for the query is predefined, the following values are passed in:

- Code – a mnemotechnic rule ID
- Rate – the weight of the rule
- InversionRate – the weight of the inverted rule[9]
- Examples – examples of expressions parsed by the rule.

In the RETURN clause the node matched under the criteria for the subst variable is aliased synh for further processing. The node will pass its syntactic features on to a new node representing the parsed phrase "*młode dziewczyny*".

The mechanism described above is insufficient to ensure accuracy of the values of attributes storing grammatical categories of the newly created node.

In the sample rule RBSubst.01 one could observe that values of the attributes storing grammatical categories should be consistent with the conditions of equalisation for grammatical attributes of the nodes adj and subst. And so, the values of attributes in the newly created node should compile with the listing beneath:

[9] Langusta supports the handling of word order inversion which is common in the Polish language which is a synthetic language. Through this mechanism the number of rules for parsing the corresponding expressions in normal and inverted order is not doubled. The use of mechanism is limited to rules which match 2 Word nodes. That means that in Langusta system, the expression "dziewczyny młode" will be parsed by the same rule (although certainly not by the same query). To apply the a given rule to the inverted word order it suffices to pass in the appropriate InversionRate value in the environment, i.e. the value of the weight for the rule which tries to perform matching using inverted order of of matching nodes.

Attribute	Expression	Value
pos	subst.pos *= [′ger′, ′pact′, ′ppas′, ′subst′]	[′subst′]
number	adj.number *= subst.number	[′pl′]
case	adj.case *= subst.case	[′nom′,′acc′,′voc′]
gender	adj.gender *= subst.gender	[′f′]

Extending of a rule with expressions for attribute values for each newly created node is carried out automatically on the basis of the analysis of the conditions of equalization.

Algorithm. The parsing algorithm applies parsing rules to the text corpus represented as a graph. The graph structure is the output from the process of tokenization and morphosyntactic analysis. When a rule is satisfied a new Word node is created to represent the correspondent piece of text. The new node inherits the syntactic features from the node designated as synh, i.e. from the syntactic head. The new node inherits all the incoming: follows relationships from the first node in the matched path. Likewise, the new node inherits all the outgoing: follows relationships from the last node in the matched path. Lastly, an: is_element_of relationship is being created between the new node and all the Word nodes of the path matched by the use of the rule.

The algorithm applies the rule set until no rule produces a new node. The algorithm guarantees that no rule will be successfully applied more than once to the same sequence of Word nodes, thus ensuring the uniqueness of their representation.

3.4 Additional Linguistic Resources

The structural capacity of the graph data model allows for a straightforward incorporation of additional linguistic resources that can be used to increase precision of parsing rules.

plWordNet. One such example is the lexical database for the Polish language, plWordNet. With this solution, the semantic dependencies can be applied at the stage of performing syntactic analysis. Let us consider the beneath core of rule designed to match phrases like "*butelka z benzyną*" or "*worki na liście*".[10]

```
MATCH[11] (cont:Word)--(prep)--(subst),
cont<-[:occurs]-(f:Form),
f-[:formof]->(b:Base),
b-[:means]->(lu:LexicalUnit),
lu<-[r:DNRS_hiponimia]-h
WHERE subst.pos *= [′ppron12′, ′ppron3′, ′subst′]
and ′subst′ in container.pos
```

[10] Phrases "bottle of gasoline", "sacks for leaves" as instances of prepositional phrases: "container of/for something". "Bottle" and "sack" are hyponyms of "container" and inherit its valency features.

[11] When the MATCH clause contains more than one path, Langusta selects the first one as the matching path by default. The unnamed and undirected relationships between the nodes on this path are labelled :follows and directed from left to right.

```
and prep.base in ['na', 'po', 'z']
and 'prep' in prep.pos and
and prep.case *= subst.case
and h.name in ['pojemnik','zbiornik']
```

In the above query we require that the noun cont be a form of a word that is a hyponyme[12] of one of the nouns: "pojemnik" or "zbiornik".

A query of this type can increase the precision of parsing of phrases like: "*Worek na ziemię został rzucony.*".

Walenty. It is only natural for a Langusta user to extend the basic set of rules by adding new rules automatically generated from the existing language resources. One such example is reaching out for the set of rules automatically generated from the valence dictionary Walenty (Przepiórkowski et al. 2014) and including them in the system.

The core of a parser rule in Langusta corresponding to a valence rule in the Walenty Dictionary:

buntować:pewny:_:imperf: prepnp(przeciw,dat)

has the form of:

```
WITH 'R.Wal.Prep.Za.Inst' as code, 40 as
rate,['buntować'] as verbBases
MATCH (Verb)--(przeciw)--(subst)
WHERE 'inst' in subst.case
and subst.pos *= ['ger', 'ppron12', 'ppron3', 'siebie',
'subst']
and 'przeciw' = przeciw.base
and 'inst' in przeciw.`case`
and 'prep' in przeciw.pos
and Verb.base in verbBases
and Verb.pos *= ['fin', 'ger', 'imps', 'impt', 'inf',
'pact', 'pant', 'pcon', 'ppas', 'praet']
```

4 Conclusions and Discussions

The distinctive features of the presented approach include:

- An open-structure, persistent and queryable corpora representation

[12] To increase ease of use of the plWordNet dictionary, the rules work with the transitive closure of the WordNet graph, traversing the hyponymy relation edges, taking into account transition through synset groups, i.e. if a lexical unit: lu1 is a hyponyme of a lexical unit lu2, then all the lexical units sharing the same synset group with lu1 are hyponymes of all lexical units sharing a synset group with lu2.

- A transparent way of dealing with ambiguity the grammatical interpretation of inflected forms and with multiplicity of syntactic trees constructed; this has been achieved by unleashing the potential of the large structural capacity of the graph data model
- Choosing an open, declarative query language as the formalism used for the description of parsing rules and the resulting ease of the automated generation of grammatical rules based on the primary linguistic resources (see **Walenty**)
- The ease of representation and use of the basic language resources for the Polish and the ease of incorporation of additional linguistic resources (see **plWordNet**)

Synergy of the above features builds up an advantage over other solutions, especially in the area of deep syntactic parsing of Polish. The advantage manifests itself in the ease of parse rule set management as well as quality of received results.

Most important works considered for the future include:

- Comprehensive analysis of the performance aspects of the proposed solution
- A comprehensive comparison of analytical capabilities for the presented solution with search engine Poliqarp[13,14]
- A more extensive use of the linguistic resources for the Polish language, currently used and those under development (i.e. semantic frames in Walenty)
- Application of the Langusta environment to more advanced topics from the field of natural language processing and understanding such as relation extraction or multi-text summarization
- Enhancing the solution by the introduction of a statistical component. The authors believe that the graph paradigm based approach present in the current solution may well be adopted in the future system featuring support for statistical methods.
- One of the identified areas of application of the statistical component is system tuning by choosing weights for rules based on statistical data.

As described above, the deployment of graph database environment has met our expectations sufficiently to allow for planning further development of the solution.

References

Buczyński, A., Przepiórkowski, A.: Demo: an open source tool for partial parsing and morphosyntactic disambiguation. In: Proceedings of LREC 2008 (2008)

Dipper, S.: Stand-off representation and exploitation of multi-level linguistic annotation. In: Proceedings of Berliner XML Tage 2005 (BXML 2005), pp. 39–50, Berlin (2005)

Graliński, F., Jassem, K., Junczys-Dowmunt, M.: PSI-Toolkit: Natural language processing pipeline. Computational Linguistics – Applications. Springer, Heidelberg (2012)

Ide, N., Suderman, K.: GrAF: a graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop, pp. 1–8. Czech Republic, Prague (2007)

[13] Poliqarp, similary to SPEJD, based its syntax on the formalism CQP derived from the project CWB − The IMS Open Corpus Workbench (http://cwb.sourceforge.net/).

[14] Poliqarp, similary to SPEJD, is was used as a part of NKJP project.

Joshi, A.K., Schabes, Y.: Tree-adjoining grammars. In: Handbook of Formal Languages, vol. 3, pp. 69–123. Springer-Verlag New York, Inc., New York (1997). ISBN:3–540-60649-1

Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems. Addison-Wesley Longman Publishing Co., Inc., Boston (2001)

Maziarz, M., Piasecki, M., Szpakowicz, S.: Approaching plWordNet 2.0. In: Proceedings of the 6th Global Wordnet Conference. Matsue, Japan (2012)

Mazur, P.: Text segmentation in polish. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 43–48, 8–10 September 2005, Wroclaw, Poland (2005)

Mihalcea, R., Radev, D.: Graph-Based Natural Language Processing and Information Retrieval. Cambridge University Press, Cambridge (2011)

Pęzik, P.: Indexed graph databases for querying rich TEI annotation (2013). http://digilab2.let.uniroma1.it/teiconf2013/wp-content/uploads/2013/09/Pezik.pdf

Przepiórkowski, A.: Powierzchniowe przetwarzanie języka polskiego. Akademicka Oficyna Wydawnicza EXIT, Warsaw (2008)

Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.): Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw (2012)

Przepiórkowski, A., Bański, P.: Which XML standards for multilevel corpus annotation? In: Proceedings of the 4th Language & Technology Conference, Poznań, Poland (2009)

Przepiórkowski, A., Buczyński, A.: Shallow parsing and disambiguation engine. In: Vetulani, Z. (ed.) Proceedings of the 3rd Language & Technology Conference, Poznań, Poland, pp. 340–344 (2007)

Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F., Świdziński M.: Walenty: Towards a comprehensive valence dictionary of Polish. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, pp. 2785–2792, Reykjavík, Iceland. ELRA (2014)

Robinson, I., Webber, J., Eifrem, E.: Graph Databases. O'Reilly Media, Massachusetts (2013)

Rudolf, M., Świdziński, M.: Automatic utterance boundaries recognition in large Polish text corpora. In: Kłopotek, M.A., Wierzchoń, S.T., Trojanowski, K. (eds.) Intelligent Information Processing and Web Mining. Advances in Soft Computing, vol. 25, pp. 247–256. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39985-8_26

Shi, C., Verhagen, M., Pustejovsky, M.: A conceptual framework of online natural language processing pipeline application. In: Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT, pp. 53–59, Dublin, Ireland, 23 August (2014)

Strauch, Ch.: NoSQL Databases (2011). http://www.christof-strauch.de/nosqldbs.pdf

Szpakowicz, S.: Automatyczna analiza składniowa polskich zdań pisanych. Praca doktorska (promotor Waligórski S.), Instytut Informatyki UW (1978)

Świdziński, M.: Gramatyka formalna języka polskiego, "Rozprawy Uniwersytetu Warszawskiego", t. 349, Warsaw (1992)

Wilson, J.R.: Introduction to Graph Theory, 4th edn. Addison Wesley, Reading (1996)

Woliński, M., Miłkowski, M., Ogrodniczuk, M., Przepiórkowski, A., Szałkiewicz, Ł.: PoliMorf: a (not so) new open morphological dictionary for Polish. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, pp. 860–864, Istanbul, Turkey. ELRA (2012)

Woliński, M., Przepiórkowski, A.: Projekt anotacji morfosynktaktycznej korpusu języka polskiego. Prace IPI PAN 938, grudzień 2001 (2001)

Wood, P.T.: Query languages for graph databases. ACM SIGMOD Rec. **41**(1), 50–60 (2012)

Zeldes, A., Ritz, J., Lüdeling, A., Chiarcos, C.: ANNIS: a search tool for multi-layer annotated corpora. In: Proceedings of Corpus Linguistics 2009, Liverpool, 20–23 July, 2009

Language Resources and Tools

RetroC – A Corpus for Evaluating Temporal Classifiers

Filip Graliński[1(✉)] and Piotr Wierzchoń[2]

[1] Faculty of Mathematics and Computer Science,
Adam Mickiewicz University in Poznań, Poznań, Poland
filipg@amu.edu.pl

[2] Institute of Linguistics, Adam Mickiewicz University in Poznań,
Poznań, Poland
wierzch@amu.edu.pl
http://amu.edu.pl
http://re-research.pl/en/

Abstract. We present a corpus for training and evaluating systems for the dating of Polish texts. A number of baselines (using year references, knowledge of spelling reforms and birth years) are given for the temporal classification task. We also show that the problem can be viewed as a regression problem and a standard supervised learning tool (Vowpal Wabbit) can be applied. So far, the best result has been achieved with supervised learning with word tokens and character 5-g as features. In addition, error analysis of the results obtained with the best solution are presented in this paper.

Keywords: Temporal classification · Optical character recognition
Vowpal Wabbit · Supervised learning

1 Introduction

In recent years, more and more historical material (such as old newspapers, books no longer under copyright and archival documents) has been digitised and made available online. Unfortunately, metadata, in particular creation/publication dates, is not always present. Moreover, old textual material is often made available on the Internet in an unstructured manner and mixed with contemporary Web texts.

The task of *automatic document dating* or *temporal text classification* consists in assigning a creation or publication date to a given text relying solely on its content – that is, without the need to use explicit metadata [4]. It can be viewed as a text classification problem (which period does it come from? – with, for instance, a yearly or decadal resolution) or as a regression problem (guess the time stamp as precisely as possible treating it as a continuous value). The task can be approached using either knowledge-based methods (through knowledge of the history of the orthography of a given language or using Wikipedia or

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 101–111, 2018.
https://doi.org/10.1007/978-3-319-93782-3_8

other external resources) or learning-based methods (supervised learning from a corpus of time-stamped texts).

In this paper, we present (1) two releases of *RetroC* – a publicly available corpus for evaluating and training systems for the automatic dating of Polish texts and (2) some baseline results obtained using the corpus.

In Sect. 2, we discuss previous work and the state of the art as regards temporal classification. Section 3 presents the rationale behind the RetroC(2) corpus, its source materials and scope. Section 4 discusses the availability of the corpus. Basic baselines and more advanced supervised methods are outlined in Sect. 5. Finally, an error analysis is presented in Sect. 6.

2 Previous Work

Although the problem of temporal classification is of significant importance for text processing and information retrieval, as well as in terms of its numerous applications (language chronologisation, support for the digitisation of cultural heritage), relevant literature is not abundant. This is probably a result of the lack of large, freely available, resources which might be used to train and test automatic dating systems.

The research problem was first raised by de Jong et al. [9]. Those authors presented an ambitious programme of using temporal unigram language models not only for the automatic dating of historic texts, but also for linking contemporary keywords with their historic variations. Since, unfortunately, there was no extensive diachronic corpus available, de Jong et al. carried out the experiment based on a fairly large but time-limited (1999–2005) corpus of Dutch-language press materials. Kanhabua and Nørvåg [10] developed further the method of de Jong et al. by applying semantic-based pre-processing (tagging parts of speech, excerpting collocations and filtering out words) and using statistical extensions of language models (word frequency interpolation, temporal entropy, the use of Google Zeitgeist). In order to learn and test the methods, a corpus of archival websites from an approximately 8-year period was used.

Evaluation of automatic dating methods was one of the objectives of the DEFT2010 workshop. To this end, a time-extensive (1800–1944), though relatively small (about 6300 texts) corpus of French newspaper texts was used. The best system obtained an F-measure of 0.338 [1]. Use was made of information about spelling reforms, birth dates of famous people and a module which learnt to chronologise vocabulary with conditional random fields.

The evaluation task was repeated during the DEFT2011 campaign. An advanced system based on information gained from external resources (birth dates, archaisms, neologisms, dates of spelling reforms) and on classification methods making use of a training corpus (classification based on the cosine distance, with modelling using support vector machines) was then constructed by Garcia-Fernandez et al. [5].

In order to improve the automatic dating results, Chambers [2] made use of a discriminant classifier, taking into account explicit temporal references in

Table 1. Summary of work on temporal classification

Paper	Language	Time span	Corpus/Size	Methods
de Jong et al. [9]	Dutch	1999–2005	2 GB raw text (train)/500 articles (test)	Unigram language models
Kanhabua and Nørvåg [10]	English	Web pages	?	POS tagging, collocations, filtering; word frequency interpolation, temporal entropy, Google Zeitgeist
Albert et al. [1]	French	1800–1944	DEFT2010 (6300 newspaper texts)	Spelling reforms, birth dates of famous people, CRF learning
Garcia-Fernandez et al. [5]	French	1801–1944	DEFT2011 (6050 newspaper texts)	External resources (birth dates, archaisms, neologisms, spelling reforms), SVM-based classification
Chambers [2]	English	1994–2002	Gigaword Corpus (New York Times section)	Discriminant classifier on temporal references and verb tenses
Ciobanu et al. [3]	Romanian	5 centuries	?	Learning-based methods
Guo et al. [8]	English	1502–2002	Hathi Trust (250K volumes)	
this paper	Polish	1814–2013	RetroC1, 59K texts, 212M	Linear regression
this paper	Polish	1814–2013	RetroC2, 153K texts, 537M	Linear regression

the dated text and parameters such as verb tense. Kumar et al. [11] applied language models learned from Wikipedia biographies to classify stories obtained from the Gutenberg Project, and Ciobanu et al. [3] trained a classifier based on a Romanian corpus, containing data from five centuries, to date contemporary historical novels. (This is a more difficult task than dating, for instance, press articles, which usually refer to events that are not distant in time from their publication dates.)

More recently, Guo et al. [8] applied various machine learning methods (e.g. SVMs) to a large dataset extracted from the HathiTrust digital library.

A summary of previous work is given in Table 1.

3 The RetroC Corpus

RetroC is a Polish-language diachronic corpus, spanning two centuries (1814–2013) and intended for training and testing automatic dating systems. It is mostly based on publications available in Polish digital libraries [7,13], plus some old textual material from other online sources.

There have been two releases of the corpus so far: the first one (RetroC1) in 2015 and the second one (RetroC2) in 2017. RetroC2 is not only larger (being a superset of RetroC1), but also contains extra features in the training set.

The corpus was designed with the following goals in mind:

- to be a collection of Polish texts;
- to be large enough to enable the use of statistical methods;
- to be time-extensive – not just modern Web-based texts, but also old printed materials;
- to cover relatively short fragments rather than whole books, for which the dating task is much easier.

In the second release of the corpus, some new objectives were considered:

- to treat time truly as a continuous variable
- and in the same time to take into account the fact that time granularity varies for publications (yearly for books, monthly for magazines, daily for newspapers, etc.);
- to make use of the fact that publications are usually clustered into collections, sources, etc.

Consequently, whereas the training set for RetroC1 contains just texts and years (given as integers) for each item, the training data in the RetroC2 corpus is a list of quintuples:

1. the beginning of the time span given as year with a fraction (e.g. 1933.7479 for a monthly published in October 1933),
2. the end of the time span given as year with a fraction (e.g. 1933.8328 for a monthly published in October 1933),
3. title of the publication,
4. identifier of the source of the publication (usually a digital library),
5. text fragment.

(3) and (4) are given only for the training set, so this information could not be used directly as a simple feature when testing. Motivation is that it could be used to detect unreliable features (e.g. words that occur only in one magazine or one source) while training. Also, the expected value for the test set is not a time span, but a single year with a fraction — mid-point of a given time span, e.g. 1933.7903 for a monthly published in October 1933, or 1921.5 for a book for which only the publication year (1921) is known.

RetroC corpora are divided into a training set, development sets and a test set. Their sizes are given in Table 2.

Table 2. Number of text fragments for each data set

	Train	dev-0	dev-1	Test
RetroC1	40,000	9,910	N/A	10,000
RetroC2	107,471	20,000	11,563	14,220

The RetroC1 dev-0 test set was incorporated into the RetroC2 training set and the RetroC1 test set became the RetroC2 dev-1 test set to avoid overfitting when switching to RetroC2 (the RetroC2 test set was formed using a completely new collection of digital libraries).

Each set is composed of 500-word fragments taken from random publications (500-word portions were also used in the DEFT corpus [5]). For instance, the following is a dev-set item taken from an 1855 publication from the e-library of Warsaw University[1] (which is the largest source of texts for the training and development sets):

> przeprawę. Szron ten zwiększa się w skutku przymrozków i śniegu, na czem w tych dniach zupełnie nam niebrak. Zapowiedziany Toro IV i ostatni dzieła p.n. Opisanie lasów Królestwa Polskiego i Gubernji Zachodnich CE- SARSTWA Rossyjskiego, już wyszedłz droku i znajduje [omitted for brevity] ubioru damskiego zastosowane, wyszły na r. 1855 nakładem i w litografji K. Romanowicza, przy ulicy Długiej Nr 578, przechodni dom na Bielańską. Nabyć ich także możua w składzie ryciu przy ulicy Senatorskiej, wdomuW 7 Neubauera. Nakładem Xięgarni Jana Breslauera, wyszła z druku powieść historyczna: Zamek Warszawski czjfli Rodzina Konrada, w 3ch tomach, przez J. N. Cżarnomskiego. Powieść ta opisuje w sposób nader zajmujący, ostatnie chwile Xięztwa Mazowieckiego i jego wcielenie do Korony. Cena exem: rs.2 k. 70 Rzadko takiego kursu sanek jak w dniu onegdajszyro, bo też dzień>był potemu, gdyż i dość mroźny, zatem pogodny i śnieg

As can be seen, a text in the RetroC corpus is given as it was found in the text layer of a DjVu/PDF file (with possible OCR noise and errors) – only minimal post-processing was applied (joining words separated with hyphens and new lines, removing end-of-lines and other non-printing characters, UTF-8 sanitisation). In contrast to the DEFT dataset [5], dates were not removed from texts (see *1855* in the example above); this was motivated by the fact that year references are obviously a useful (though not perfect) feature for temporal classification (and we aim to use classifiers trained with RetroC to find old textual material in large Web corpora where dates are available), although it is not as important as in [8], where whole volumes, including copyright and title pages, are taken into account.

The development and test sets are balanced with respect to publication year: 50 and 100 publications per year for, respectively, RetroC1 and RetroC2. We

[1] http://ebuw.uw.edu.pl.

were not able to find very many dev-set items for some years (in particular, during the early 19th century and World War II), hence the size of some sets is smaller than 200 (years) × 50/100 (texts). The development set and the test set are also balanced (as much as possible) with respect to their sources, in order to avoid the data set being overwhelmed by one large digital library. The training set is not balanced; the distribution of publication years therein is presented in Fig. 1.

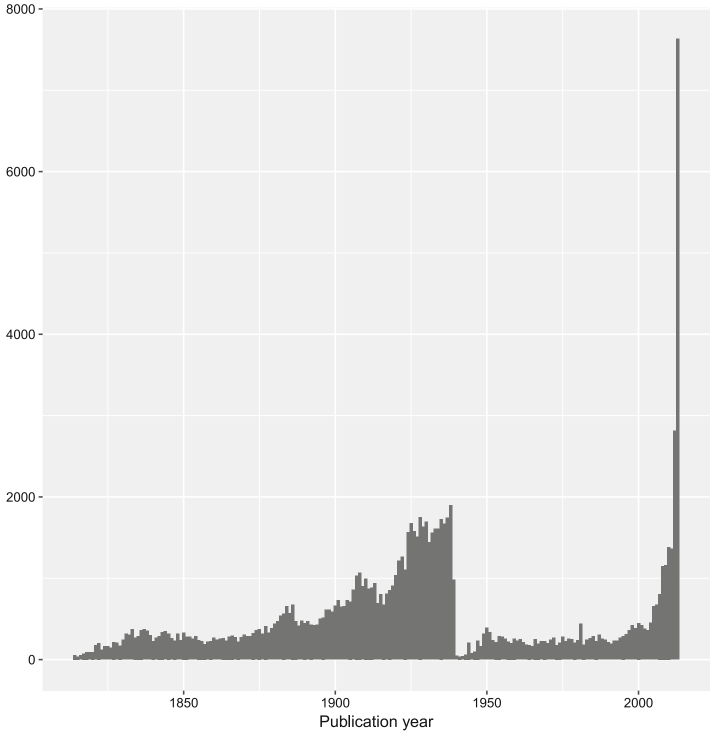


Fig. 1. Number of items in the training set

The training and dev-0 development sets are composed of texts from the same set of digital libraries. In order to make the challenge more difficult, the texts in the test set were taken from a separate set of sources (i.e. different digital libraries). This is a more realistic approach, as we would require that a temporal classifier work reliably on texts from new, unknown sources. To assess the quality of generalisation, an extra dev-1 development set taken from yet another set of sources was added in RetroC2.

The publication dates were extracted from the metadata from the digital libraries; no manual verification was performed, and there is no guarantee that all of the dates given in RetroC are correct (we also ignore whether it is in fact a publication date or creation date that is given).

4 RetroC as a Machine Learning Challenge

RetroC(2) data sets are available at Gonito.net (see http://gonito.net/challenge/retroc and http://gonito.net/challenge/retroc2). Gonito.net is an open source, web- and Git-based platform for hosting challenges for researchers in the field of machine learning (in particular: natural language processing) [6].

The key design feature of Gonito.net is using Git for managing challenges and solutions of the problems submitted by competitors. Thus, the corpus is freely available, simply from a Git repository (see repositories `git://gonito.net/retroc.git` and `git://gonito.net/retroc2.git`). In other words, there is no need to log in to Gonito.net web application to just download the data, Git command-line tool is enough.

Gonito.net web application is used to submit solutions and keep track of the effort of a given research community and progress in terms of clear evaluation metrics. Submitters are encouraged (but not forced) to upload source codes along with the test outputs as this allows for research transparency and reproducibility. For each solution described in this paper, a Gonito.net reference is given, for instance the null model that always returns 1913.5 (midpoint for the whole RetroC time span) is available at Gonito.net at {2ef3f0} (in case you are reading a physical copy of this paper, go to http://gonito.net/q and enter the reference number there). The Gonito.net reference is basically a Git commit ID, so even if the Gonito.net platform ceases to exist, the results and source codes may still be available as a regular Git repository to be cloned and inspected with standard Git tools, no external database is needed.

The Gonito.net machine learning task defined along with the RetroC(2) corpus is configured to use root-mean-square error (RMSE) as the evaluation metric, e.g. the null model yields RMSE = 52.5.

5 Baseline Solutions

5.1 Simple Baselines

A very simple baseline is to return the latest year reference found in the text (and back up to the null model if no year reference is found). This simple solution yields RMSE = 37.7 for RetroC2 (Gonito.net reference: {c9c6ce}), which is surprisingly much better than the null model.

Another simple method would be to use hand-crafted rules using the knowledge of spelling changes in Polish (*-dz*/*-c* ending for verbs, *-ya*/*-ja*/*-ia* ending for nouns, *-emi*/*-ymi* ending for adjectives, see Fig. 2). We obtained RMSE = 44.2 with this simple solution ({9f55cd}).

Both simple methods chained (first checking year references, then hand-crafted rules) yielded RMSE = 35.8 ({bd8665}), which could be treated as a simple rule-based baseline.

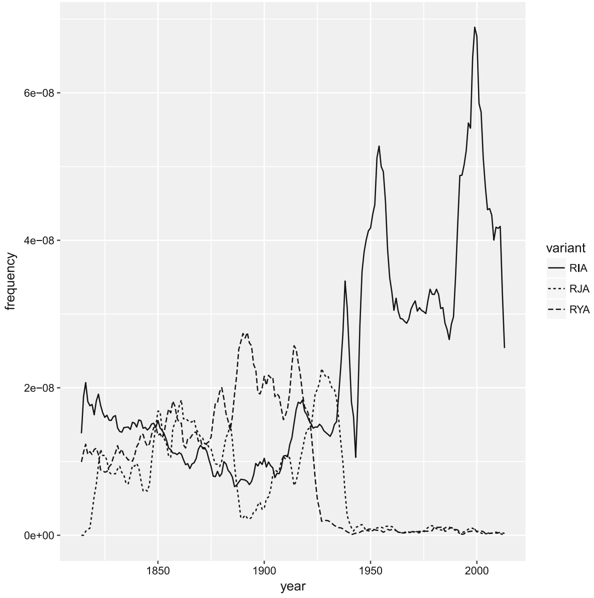


Fig. 2. Frequency of orthographic variants

5.2 Supervised Learning

As time could be treated as a continuous variable, temporal classification could be viewed as a regression problem. We trained a regressor using the Vowpal Wabbit open-source learning system [12] and RetroC training set. Both characters 5-g (as suggested in [5]) and word tokens were used as features. In addition, a small neural network (6 units) was used. This way, the best results so far were obtained: RMSE = 24.8 years for RetroC1 {9dcf6a} and RMSE = 19.5 years for RetroC2 {6ab497}. Better results for RetroC2, though not strictly comparable, might suggest that more training data might still improve the results.

The character n-grams with the highest weights (for RetroC1, with neural network switched off) are presented in Table 3. Some of the most informative features are quite obvious (e.g. year references), others less so – for instance, frequent words *ktoś* (*somebody*) and *czym* (*with what*) are informative as they were spelled as *któś* and *czem* during the 19th century, *tzw.* (*so-called*) is a relatively new abbreviation, *tal.* is a abbreviation for a monetary unit used in the 19th centry (*talar*), *aig* is a very frequent Polish word *się* mis-recognised by OCR.

In order to test the assumption that giving publication time with the highest resolution possible brings improvement when training a temporal classifier (an extra assumption introduced in RetroC2), the best solution was re-trained with publication time-stamps rounded to full years. As expected, the results were slightly, but significantly worse for the test set (RMSE = 19.7, {60e217}), which confirms the assumption.

Table 3. The features with the highest scores

	Positive	Negative
1	stori	tém
2	czym	aig
3	wtedy	»
4	*dash*	il
5	”	i5
6	tzw	)”
7	‘	téj
8	ktoś	tal
9	2009	1837
10	1985	storj

6 Error Analysis

In order to (1) learn of any defects in the corpus and (2) get insights how to improve the temporal classifier, we compared dates returned by the best solution obtained so far ({6ab497}) with the expected dates – see Fig. 3. As can be seen, whereas there is a clear tendency for the oldest texts to be misclassified as belonging to the later period, more noise can be observed for the texts from the late 20th century. In addition, there are a number of outliers. The top 100 outliers (the test cases with the highest discrepancy between publication date expected and the value returned) were inspected manually:

- 5 texts were assigned incorrect temporal metadata (e.g. 1983 instead of 1893) and the publication dates returned by the best temporal classifier were actually, more or less, correct;
- 19 text fragments were written (all or nearly all) in a language other than Polish – Russian, German, French and Latin texts were usually misclassified as earlier ones (as it was more common to found such texts among Polish publications in the 19th and early 20th centuries), whereas English texts – as later (this is partly also a metadata problem, as texts in foreign languages *are* filtered out using language metadata);
- 32 texts were misclassified due to high level of OCR noise (even though some heuristics had been used to remove such texts);
- 11 text fragments were lists of items (words, surnames, football clubs, book titles), which made them difficult to classify;
- 11 texts referred to earlier periods (e.g. excerpts from historical journals);
- for 22 texts no clear reason for the discrepancy was identified, in some cases it seemed that old texts were too “clean” (novels from the 19th century manually re-typed, no OCR noise).

The conclusion is that (1) temporal classifiers could be used to detect defects and anomalies in the temporal metadata (running the classifier on a text with known publication year and checking manually if the discrepancy is too high) and (2) there is still some room for improvement (e.g. by better filtering texts with high level of OCR noise) for RetroC data.

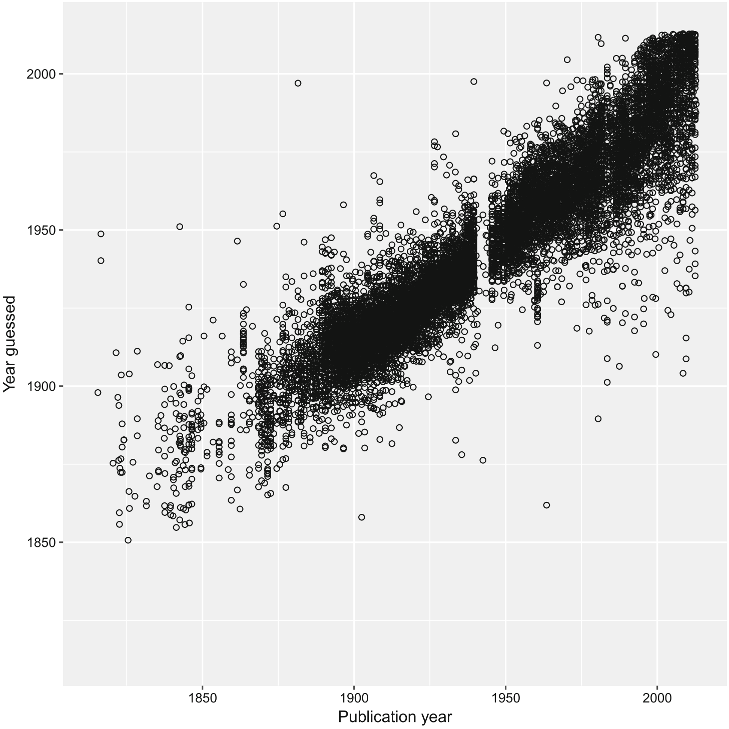


Fig. 3. Publication dates – years expected vs guessed (dev-1 set)

7 Conclusions and Further Work

We have presented two releases of RetroC, a Polish corpus for evaluating temporal classifiers, and reported initial results for certain methods. It has been shown that automatic dating can be treated as a regression problem, and that a standard machine learning tool (Vowpal Wabbit) can be used to obtain fairly good results.

For future work, we plan to implement all the advanced classification methods known in the literature for other languages, and compare and combine them with the regression methods. Also, we plan to use the temporal classifier trained on RetroC data for detecting old text fragments in large Web corpora.

References

1. Albert, P., Badin, F., Delorme, M., Devos, N., Papazoglou, S., Simard, J.: Décennie d'un article de journal par analyse statistique et lexicale. In: Proceedings of Traitement Automatique des Langues Naturelles (TALN), pp. 85–97 (2010)
2. Chambers, N.: Labeling documents with timestamps: learning from their time expressions. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 98–106. Association for Computational Linguistics (2012)
3. Ciobanu, A.M., Dinu, L.P., Sulea, O.M., Dinu, A., Niculae, V.: Temporal text classification for Romanian novels set in the past. In: RANLP, pp. 136–140 (2013)
4. Dalli, A., Wilks, Y.: Automatic dating of documents and temporal text classification. In: Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 17–22. Association for Computational Linguistics (2006)
5. Garcia-Fernandez, A., Ligozat, A.-L., Dinarelli, M., Bernhard, D.: When was it written? Automatically determining publication dates. In: Grossi, R., Sebastiani, F., Silvestri, F. (eds.) SPIRE 2011. LNCS, vol. 7024, pp. 221–236. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24583-1_22
6. Graliński, F., Jaworski, R., Borchmann, Ł., Wierzchoń, P.: Gonito.net - open platform for research competition, cooperation and reproducibility. In: Branco, A., Nicoletta, C., Khalid C. (eds.), Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language, pp. 13–20 (2016)
7. Graliński, F.: Polish digital libraries as a text corpus. In: Proceedings of 6th Language and Technology Conference, Poznań, pp. 509–513 (2013)
8. Guo, S., Edelblute, T., Dai, B., Chen, M., Liu, X.: Toward enhanced metadata quality of large-scale digital libraries: estimating volume time range. In: iConference 2015 Proceedings (2015)
9. Jong, d.F., Rode, H., Hiemstra, D.: Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences (2005)
10. Kanhabua, N., Nørvåg, K.: Using temporal language models for document dating. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5782, pp. 738–741. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04174-7_53
11. Kumar, A., Baldridge, J., Lease, M., Ghosh, J.: Dating texts without explicit temporal cues, CoRR abs/1211.2290 (2012). http://arxiv.org/abs/1211.2290
12. Langford, J., Li, L., Zhang, T.: Sparse online learning via truncated gradient. In: Advances in Neural Information Processing Systems, pp. 905–912 (2009)
13. Wierzchoń, P.: Fotodokumentacja 3.0. Language, Communication. Information **4**, 63–80 (2009)

Reinvestigating the Classification Approach to the Article and Preposition Error Correction

Roman Grundkiewicz(✉) and Marcin Junczys-Dowmunt

Adam Mickiewicz University, ul. Wieniawskiego 1, 61-712 Poznań, Poland
{romang,junczys}@amu.edu.pl

Abstract. In this work, we reinvestigate the classifier-based approach to article and preposition error correction going beyond linguistically motivated factors. We show that state-of-the-art results can be achieved without relying on a plethora of heuristic rules, complex feature engineering and advanced NLP tools. A proposed method for detecting spaces for article insertion is even more efficient than methods that use a parser. We examine automatically trained word classes acquired by unsupervised learning as a substitution for commonly used part-of-speech tags. Our best models significantly outperform the top systems from CoNLL-2014 Shared Task in terms of article and preposition error correction.

Keywords: Grammatical error correction · Article errors
Preposition errors · CoNLL-2014 shared task · Detecting omitted words

1 Introduction

In the field of grammatical error correction (GEC), a large effort is made to design models and algorithms that incorporate linguistic knowledge. Heuristic rules, advanced tools or resources for natural language processing that were not created specifically with grammatical error correction in mind are commonly used. This results in a high degree of complexity with modest gains in overall performance. Results are difficult to reproduce and the integration of different systems is complicated. We believe, in accordance with Occam's Razor, that between two models that solve the same problem on similar levels of quality the simpler one is to be preferred.

In this work, we reinvestigate the classifier-based grammatical error correction paradigm by reducing its dependence on heuristic rules and advanced natural language processing tools. We focus on two of the most frequent error types among English as a second language (ESL) learners: article and preposition errors.

The only features we allow ourselves to use are simple n-gram features of: surface level tokens, part-of-speech (POS) tags, and automatically trained word classes (AWC). Where possible we try to replace POS tags with AWC tags. The

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 112–122, 2018.
https://doi.org/10.1007/978-3-319-93782-3_9

latter are language-independent tags produced by clustering vector space representations of words which in turn are learnt on large unannotated text [14]. These have been applied to a variety of NLP problems, such as document classification [21], statistical machine translation [12] or named entity recognition, and chunking [24], usually with beneficial effects.

Our main contributions are the following: Firstly, a new contextual method for detecting omitted articles is introduced that in practice outperforms previous methods. Secondly, we apply unsupervised word classes to classifier-based GEC. Finally, we show that it is possible to achieve state-of-the-art results for article and preposition error correction with almost no linguistic knowledge.

The remainder of the paper is organized as follows: Sect. 2 reviews recent research. Data sets, classification algorithms, feature sets, and evaluation schemes are described in Sect. 4. Section 3 deals with detection of spaces for potential article insertions. In Sect. 4, we present our results and compare them with top systems from the CoNLL 2014 shared task [16]. Conclusions and future work are presented in Sect. 5.

2 Related Work

In this section we briefly discuss related work with predominantly classifier-based approaches which focused on correcting mistakes in article and preposition usage. For more comprehensive description of the field we refer the reader to the work of Leacock et al. [13] and the recent CoNLL shared tasks [16,17].

The majority of researchers use lexical word forms, POS tags and structural information from shallow parser when designing features for article error correction classifiers. Features that encode linguistic knowledge are extracted, for example combinations of words preceding the article and a head word of the identified noun phrase [8,10].

For instance, Rozovskaya et al. [18] design high-level features that encode POS tags and shallow parse properties. The authors show that adding rich features to the baseline system that uses only word n-grams is helpful. However, they do not compare these rich features with simple POS n-grams.

Features used for preposition error correction are usually less complex and base on lexical forms of surrounding words [11,20]. Linguistically more complex knowledge is encoded in features that make use of various aspects of preposition complements [23] or additional features derived from a constituency and a dependency parse trees [22].

The results of Rozovskaya et al. [18,19] are most similar to our work as all features are lexical, but the only type of n-grams tested in this work are pure word n-grams.

2.1 Confusion Sets

Typically, a confusion set for article and determiner error correction consists of three units: $\{$*a, the,* $\varnothing\}$[1]. This covers article insertion, deletion, and substitution

[1] $\varnothing$ stands for the English zero article.

errors. The distinction between *a* and *an* is usually made with heuristic rules during post processing. Since we made a point of not using any heuristic rules, our confusion set comprise both indefinite article variants, taking the final form: {*a, an, the,* ∅}.

In preposition error correction it is common to include in confusion set the top n most frequent English prepositions [2,8,20]. We restrict ourselves to the top twelve prepositions: {*in, at, on, for, since, with, to, by, about, from, of, as*} that cover 88.6% of all preposition errors in NUCLE (see Sect. 2.2). In contrast to previous studies that consider only incorrectly selected prepositions, we handle also extraneous and (for final models) missing ones.

2.2 Data Sets

For training and testing our models we use various versions of two learner data resources: the NUS Corpus of Learner English and the Lang-8 corpus. A brief summary of used corpora is presented in Table 1.

Table 1. Basic statistics of data sets used in experiments: size in sentences and error rates (in %) for article and preposition errors. For TS-2014 values are reported for two annotators separately.

Corpus	Size	ER_{art}	ER_{prep}
NUCLE	57,151	**6.68**	**2.34**
TS-2013	1,381	18.27	5.39
TS-2014 A0	1,312	10.23	5.08
TS-2014 A1	1,312	13.72	6.72
L8-NAIST	2,215,373	15.86	7.61
L8-WEB	3,386,887	18.55	9.22

The NUS Corpus of Learner English (NUCLE) [4] consists of 1,414 essays (57,151 sentences) which cover a wide range of topics, such as environmental pollution and health care. It was used as training data in two editions of the CoNLL shared task on Grammatical Error Correction [16,17].

We also make use of the official test sets from the shared tasks (TS-2013 and TS-2014). This data covers similar topics as NUCLE, but is smaller (1,381 and 1,312 sentences) and has higher frequencies of both error types.

By "Lang-8" we refer to a collection of posts scrapped from a language exchange social networking website named Lang-8[2]. We use the English part of the publicly available "Lang-8 Learner Corpora v1.0" [15] (L8-NAIST).

Furthermore, we have scrapped recent data from the Lang-8 website which resulted in a resource (L8-WEB) that is about one and a half times larger than L8-NAIST.

[2] http://lang-8.com/.

2.3 Classification Algorithm

Our largest models are trained on over 4 million training examples represented as binary feature vectors of a length that exceeds 1.5 million features. Therefore, we decided to use the L2-regularized logistic regression from LIBLINEAR [5] which supports large-scale multi-class classification. Logistic linear regression has been used before for correction of both, article and preposition errors [2,11,23].

2.4 Feature Sets

During experiments we use various combinations of the following features:

- *source*—a source confused word encountered in the input text, i.e. the original article or preposition.
- *tokens*—n-grams of lowercased tokens around the confused word. All n-grams have lengths between one and four, and include or are adjacent to the position of confused words.
- *POS*—n-grams of part-of-speech tags obtained by the Stanford Part-Of-Speech Tagger. The tagset consists of 43 tags.
- *AWC*—n-grams of automatic word classes created with the *word2vec* toolkit[3] [14]. The number of clusters and vector length were set to 200. Other than that, default options were used. We learnt word vectors from 75 millions of English sentences extracted from Common Crawl data[4].
- mix_{tags}—n-grams that consist of mixed tokens and tags, e.g. for tokens w_1, w_2, w_3, and corresponding tags t_1, t_2, t_3, the mixed n-grams are: t_1w_2, w_1t_2, $t_1w_2w_3$, $w_1t_2w_3$, $w_1w_2t_3$, $t_1t_2w_3$, $w_1t_2t_3$, etc.

For each word included in a confusion set encountered in the to-be-corrected text, we extract specific features which are later converted to binary feature vectors.

It should be emphasized that experimenting with various vector space representation models, size of space dimensions and number of clusters are not within the scope of this work.

2.5 Evaluation

We use the evaluation scheme and official test sets from the CoNLL-2014 shared task [16]. The system outputs submitted by participants are publicly available[5], so that we can easily compare our models with top systems from this competition. The participants were free too use all resources that were publicly available, in particular the NUCLE corpus and test set from 2013.

System performance is measured by the MaxMatch (M^2) metric [3] which computes an $F_{0.5}$ score for the proposed corrections against a gold standard that has been similarly annotated as NUCLE.

[3] https://code.google.com/p/word2vec/.

[4] https://commoncrawl.org/.

[5] http://www.comp.nus.edu.sg/~nlp/conll14st.html.

The original test sets contain annotations of errors from 28 error categories. Evaluation focused on specific errors only results in very low recall in a 28 error type context, which disturbs the tuning process as well as final results due to harmonic properties of F-score. Therefore, we have modified gold standards for each test set by preserving only annotations for which the erroneous or corrected texts concern words from our confusion set (keeping deletions and insertions as well).

This method works better for us than rely on the original error categories, since many annotations that involve articles or prepositions are categorized differently (e.g. many article deletions are categorized as "local redundancy").

3 Detection of Article Omissions

Article omissions represent a majority of article and determiner errors, for example, in NUCLE they constitute about 61.38% of all article errors. The most common solution for detecting positions where an article might have been incorrectly omitted is to use a shallow parser to identify spaces occurring before noun phrases [10,18]. All noun phrases headed by a personal or demonstrative pronoun are excluded. Some research extends this by taking into account additional spaces following a preposition or a verb even when these are not identified by the parser.

On the other hand, a naive method which includes every space as a potential position for article insertion is considered to produce a lot of noise.

3.1 Detection by Context Comparison

We tested a new method of detection of spaces for potential article insertions based on the comparison of surrounding context. The proposed method consists of a training and a detecting stage.

During training, we extract n-grams from a text corpus consisting of l tokens to the left and r tokens to the right of each occurrence of words from the confusion set. Next, in the to-be-corrected text we flag each space for which a matching n-gram from the set of n-grams extracted during the training stage is found. Changing the minimum count c required for n-grams to be used for detection allows for control of the number of detected spaces. This procedure can be used with token n-grams and POS or AWC tags.

We estimated experimentally the values $l = 1$ and $r = 3$ for article errors, and $l = 3$, $r = 1$ for preposition errors (final models only). The n-grams were trained on a part of English Common Crawl Corpus consisting of ca. 75 million sentences.

3.2 Comparison of Detection Methods

We compared several methods for finding spaces for potential article insertions in the task of zero article detection. We used the entire NUCLE corpus as test set. The only positive class during evaluation (true positive) was the proper detection of a space where, according to the annotation, an article is missing. A good method should achieve a high recall and a low false positive rate (FPR). Results on NUCLE are presented in Table 2.

Table 2. Comparison of various methods for detecting spaces for potential article insertions. Results for NUCLE corpus: number of true positives (TP) and false negatives (FP), false-positive rate (FPR) and recall (R).

Method	TP	FP	FPR	R
Naive	3,346	984,307	91.88	100.00
NP	2,871	**186,484**	68.20	**85.80**
$\mathrm{NP}_{verb,prep}$	3,041	324,531	78.87	90.88
tokens_5	1,059	35,543	29.02	31.56
AWC_5	2,797	348,746	80.04	**83.34**
AWC_{50}	2,157	**178,290**	67.22	64.27
AWC_{500}	1,159	60,402	40.99	34.54
POS_{50}	3,167	527,534	85.85	94.37
POS_{500}	2,901	315,537	78.40	**86.44**
POS_{5000}	2,520	**170,078**	66.17	75.09

A naive method (*naive*) detects all of 984,307 spaces between words excluding spaces before and after *a, an* or *the.* A method that uses a shallow parser (*NP*) results in recall of 85.80, similarly to methods based on AWC n-grams with $c = 5$ (AWC_5) and POS n-grams (POS_{500}). But the latter almost double the number of false positives. Enforcing similar FPR requires to set $c = 50$ for AWC and $c = 5000$ for POS tags. For the context coverage method, recall can be adjusted by setting c, which should be determined based on the size of the training data. In the experiment, all n-grams were extracted from a part of English Common Crawl Corpus [1] consisting of ca. 75 million sentences.

We further evaluate these methods in the article error correction task in Sect. 4.1

4 Experimental Results

We use 4-fold cross validation on NUCLE to adjust threshold values of the minimum classifier confidence required to accept its prediction. During each of the steps, additional data in the form of Lang-8 corpora is added as training data. Then, we train the classifier again on the entire data and for final evaluation we use an averaged confidence threshold.

To prevent the classifier from keeping the input text unchanged [2], the error rate of the training data was increased by randomly removing correct sentences.

We experimentally set the error rate to 30% for article errors and to 20% for preposition errors. This procedure keeps 873,917 and 1,660,896 sentences in L8-NAIST and L8-WEB respectively for training article models, and 1,219,127 sentences in L8-WEB for training preposition models.

The TS-2013 is used to determine an error rate for tuning data in cross validation as there is a significant disproportion in error rates. We report results on both test sets, for article and for preposition models. The higher results on the CoNLL-2014 test set are due to it has been annotated by two annotators.

4.1 Methods for Detecting Article Omissions

In order to compare the various methods of detecting spaces for possible article insertion (Sect. 3), we used the tuned L8-NAIST corpus as training data with feature set consisting of token *n*-grams. Table 3 presents the results.

Table 3. The comparison of different methods for article omission detection. All models are trained on L8-NAIST and use *source* and *tokens* features.

Method	TS-2013			TS-2014		
	P	R	$M^2_{0.5}$	P	R	$M^2_{0.5}$
Naive	57.58	3.66	14.59	68.97	12.20	35.71
NP	54.44	9.44	**27.87**	58.33	20.83	42.89
$\mathrm{NP}_{verb,prep}$	51.06	9.25	26.82	53.03	20.71	40.42
POS_{5000}	54.17	7.51	24.16	65.08	23.70	**48.24**
AWC_{50}	54.55	8.09	25.39	62.26	19.53	**43.31**

A naive method gives the lowest $F_{0.5}$ scores due to the high precision but low recall. Using a lower error rate in the training data shows a similar effect. Methods based on a shallow parser (NP) are more effective without augmenting them with spaces after each verb and preposition ($\mathrm{NP}_{verb,prep}$) on both test sets. The proposed methods that compare surrounding context are significantly better on TS-2014 and reach slightly lower results on TS-2013. It is unclear why AWC *n*-grams are more effective than POS *n*-grams for TS-2013 and vice versa for TS-2014.

We also experiment with applying the proposed methods to handle missed preposition errors in our final models (Table 5). This increases the recall, since it enables making corrections that can not be detected otherwise, but may reduce precision.

4.2 Different Feature Sets and Final Models

Next, we compare different feature sets in Table 4. For article models we chose a method of detecting omissions that uses AWC *n*-grams due to its speed and

Table 4. Final results on the CoNLL-2013 and CoNLL-2014 test sets for article (top) and preposition (bottom) error correction. Article models are trained on L8-NAIST and use AWC_{50} method for omission detection. Preposition models are trained on L8-WEB. All models use *source* features.

Feature set	TS-2013			TS-2014		
	P	R	$M^2_{0.5}$	P	R	$M^2_{0.5}$
tokens	53.01	8.48	25.85	62.30	21.84	45.45
tokens+POS	45.22	10.02	26.56	60.20	32.42	**51.39**
tokens+POS+mix$_{POS}$	40.22	13.87	**29.15**	54.78	33.69	48.69
tokens+AWC	44.92	10.21	26.74	63.64	28.16	**50.83**
tokens+AWC+mix$_{AWC}$	42.31	12.72	**28.87**	56.38	29.44	47.66
tokens+POS+AWC	30.10	17.34	26.24	41.74	49.51	43.09
tokens	42.42	7.37	21.74	70.00	17.36	43.57
tokens+POS	40.00	8.42	**22.86**	59.46	18.03	40.74
tokens+POS+mix$_{POS}$	34.09	7.89	20.49	48.72	14.96	33.57
tokens+AWC	36.36	8.42	21.86	67.65	19.66	**45.45**
tokens+AWC+mix$_{AWC}$	24.49	6.32	15.54	50.94	21.95	40.30
tokens+POS+AWC	37.21	8.42	22.10	68.75	19.13	45.27

simplicity. For preposition models we used L8-WEB corpus as training data to get a sufficient number of training examples.

Models trained only on lexical features result in $F_{0.5}$ values that are slightly lower than results achieved by models that use more complex features. Using POS or AWC n-grams shows improvement in performance for both, article and preposition models. Although adding mixed n-grams is shown to improve performance in contextual spell checking [7], in our experiments it has a positive effect only for articles on TS-2013.

The results for final models trained on L8-WEB corpus are presented in Table 5. Training article models on larger corpus shows further improvement since more training examples are used. It also shows that POS tags (51.56) are more effective in article error correction than AWC tags (49.10).

For preposition errors, the highest result on TS-2014 (52.63) is achieved by a model using tokens and AWC tags and handling preposition omissions. Further investigation of automatic word classes and various numbers of classes is required.

4.3 Top Systems from the CoNLL-2014 Shared Task

Finally, we compare our best models with top three systems participating in the CoNLL-2014 shared task.

The best system [6] (CAMB) participating in the task uses a hybrid approach, which includes both a rule-based and an SMT system augmented by a large

Table 5. Final results on the CoNLL-2013 and CoNLL-2014 test sets for article (top) and preposition (bottom) error correction. All models are trained on L8-WEB and use *source* features.

System	TS-2013			TS-2014		
	P	R	$M^2_{0.5}$	P	R	$M^2_{0.5}$
POS_{5000}; tokens+POS	47.26	13.29	31.28	57.89	35.87	**51.56**
AWC_{50}; tokens+AWC	42.13	14.45	30.46	55.56	33.52	49.10
POS_{5000}; tokens+POS	34.78	8.42	21.39	58.14	20.66	42.66
AWC_{50}; tokens+AWC	36.00	9.47	23.08	75.68	23.73	**52.63**

web-based language model. The system of Rozovskaya et al. [19] (CUUI) for article error correction makes use of the averaged perceptron algorithm and POS-tagger and chunker outputs to generate some of its features and correction candidates. For preposition errors a naive Bayes classifier is trained on n-grams counts from the Google n-gram corpus. AMU [9] is a phrase-based SMT system combining large training resources, task-specific parameter tuning and features.

The systems participating in the shared task were free to use test data from 2013. In addition to the NUCLE and test set from CoNLL-2013, all systems make use of other resources that are significant in size. The CAMB system uses Cambridge Learner Corpus, which is comparable in size with Lang-8 data. A module for preposition error correction in the CUUI system is trained on the Google 1 T 5-gram Corpus. The AMU system is trained on data scraped from Lang-8 of similar size to our L8-WEB corpus.

Table 6. Top systems from the CoNLL-2014 shared task.

System	ArtOrDet			Prep		
	P	R	$M^2_{0.5}$	P	R	$M^2_{0.5}$
CAMB	39.00	65.00	**42.39**	41.15	51.63	**42.89**
CUUI	28.41	72.06	32.32	32.04	26.61	30.78
AMU	40.54	25.28	36.17	46.05	28.00	40.79
this work	57.89	35.87	**51.56**	75.68	23.73	**52.63**

System outputs submitted by participants contain corrections of errors of various types. Thus, we removed corrections that do not concern words from confusion sets (i.e. from system outputs we extracted corrections that concern article or preposition errors only), similarly as reported for the official test sets. Results are presented in Table 6.

Our best model for article error correction trained on token and POS features significantly beats the CAMB system by nearly 10% F-score (42.39 vs.

51.56). The top preposition model that uses AWC features and handles preposition omissions outperforms the top system from CoNLL-2014 in similar amount (42.89 vs. 52.63). We generally achieve a higher precision and lower recall than other systems.

5 Conclusions and Future Work

In this paper we reinvestigated the classifier-based approach in grammatical error correction by reducing the linguistic knowledge hidden in many aspects of system development. We have shown that state-of-the-art results can be achieved without applying a multitude of heuristic rules, complex feature engineering, and advanced NLP tools.

Although, for article error correction the best performance is achieved by models trained on POS *n*-grams, AWC *n*-grams also outperform lexical features and top systems participating in the CoNLL-2014 shared task. For preposition error correction, models that use AWC features and allow preposition insertions outperform other systems. Our results have shown that the proposed simple contextual method for detecting omitted articles is competitive with methods relying on chunker outputs.

This work allows to believe that automatic word classes trained with unsupervised methods are promising substitution for part-of-speech tags at least in some applications.

In the future, we plan a deeper examination of the application of automatic word classes to GEC. Other models for unsupervised learning of word representations should be tested, as well as different numbers of word clusters.

Acknowledgements. This work has been funded by the National Science Centre, Poland (Grant No. 2014/15/N/ST6/02330).

References

1. Buck, C., Heafield, K., Van Ooyen, B.: N-gram counts and language models from the common crawl. In: LREC. vol. 2, p. 4 (2014)
2. Cahill, A., Madnani, N., Tetreault, J.R., Napolitano, D.: Robust systems for preposition error correction using Wikipedia revisions. In: NAACL-HLT, pp. 507–517 (2013)
3. Dahlmeier, D., Ng, H.T.: Better evaluation for grammatical error correction. In: NAACL-HLT, pp. 568–572 (2012)
4. Dahlmeier, D., Ng, H.T., Wu, S.M.: Building a large annotated corpus of learner English: the NUS corpus of learner English. In: BEA8 Workshop, pp. 22–31 (2013)
5. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR **9**, 1871–1874 (2008)
6. Felice, M., Yuan, Z., Andersen, Ø.E., Yannakoudakis, H., Kochmar, E.: Grammatical error correction using hybrid systems and type filtering. In: CoNLL, pp. 15–24 (2014)

7. Fossati, D., Di Eugenio, B.: A mixed trigrams approach for context sensitive spell checking. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 623–633. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-70939-8_55
8. Gamon, M., Gao, J., Brockett, C., Klementiev, A., Dolan, W.B., Belenko, D., Vanderwende, L.: Using contextual speller techniques and language modeling for ESL error correction. IJCNLP **8**, 449–456 (2008)
9. Grundkiewicz, R., Junczys-Dowmunt, M.: The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. CoNLL pp. 25–33 (2014)
10. Han, N.R., Chodorow, M., Leacock, C.: Detecting errors in english article usage by non-native speakers. JNLE **12**(02), 115–129 (2006)
11. Han, N.R., Tetreault, J.R., Lee, S.H., Ha, J.Y.: Using an error-annotated learner corpus to develop an ESL/EFL error correction system. In: LREC (2010)
12. Koehn, P., Hoang, H.: Factored translation models. In: EMNLP-CoNLL, pp. 868–876 (2007)
13. Leacock, C., Chodorow, M., Gamon, M., Tetreault, J.: Automated grammatical error detection for language learners. Synth. Lect. Hum. Lang. Technol. **3**(1), 1–134 (2010)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
15. Mizumoto, T., Hayashibe, Y., Komachi, M., Nagata, M., Matsumoto, Y.: The effect of learner corpus size in grammatical error correction of ESL writings. In: COLING, pp. 863–872 (2012)
16. Ng, H.T., Wu, S.M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., Bryant, C.: The CoNLL-2014 shared task on grammatical error correction. In: CoNLL, pp. 1–14 (2014)
17. Ng, H.T., Wu, S.M., Wu, Y., Hadiwinoto, C., Tetreault, J.: The CoNLL-2013 shared task on grammatical error correction. In: CoNLL (2013)
18. Rozovskaya, A., Chang, K.W., Sammons, M., Roth, D.: The University of Illinois system in the CoNLL-2013 shared task. In: CoNLL. pp. 13–19 (2013)
19. Rozovskaya, A., Chang, K.W., Sammons, M., Roth, D., Habash, N.: The Illinois-Columbia system in the CoNLL-2014 shared task, pp. 34–42 (2014)
20. Rozovskaya, A., Roth, D.: Generating confusion sets for context-sensitive error correction. In: EMNLP, pp. 961–970 (2010)
21. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. (CSUR) **34**(1), 1–47 (2002)
22. Tetreault, J., Foster, J., Chodorow, M.: Using parse features for preposition selection and error detection. In: ACL, pp. 353–358 (2010)
23. Tetreault, J.R., Chodorow, M.: The ups and downs of preposition error detection in ESL writing. In: COLING, pp. 865–872 (2008)
24. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: ACL, pp. 384–394 (2010)

Binary Classification Algorithms for the Detection of Sparse Word Forms in New Indo-Aryan Languages

Rafał Jaworski(✉), Krzysztof Jassem, and Krzysztof Stroński

Adam Mickiewicz University in Poznań, Poznań, Poland
rjawor@amu.edu.pl

Abstract. This paper describes experiments in applying statistical classification algorithms for the detection of converbs – rare word forms found in historical texts in New Indo-Aryan languages. The digitized texts were first manually tagged with the help of a custom made tool called IA Tagger enabling semi-automatic tagging of the texts. One of the features of the system is the generation of statistical data on occurrences of words and phrases in various contexts, which helps perform historical linguistic analysis at the levels of morphosyntax, semantics and pragmatics. The experiments carried out on data annotated with the use of IA Tagger involved the training of multi-class and binary POS-classifiers.

1 Introduction

The aim of the present paper[1] is twofold. Firstly, it attempts to give a brief overview of a tool designed for semi-automatic annotation of early New Indo-Aryan (NIA) texts. Secondly, it focuses on several aspects of automatic POS-tagging of early NIA.

There has been a considerable amount of corpus-based research into Old and New Indo-Aryan, which has already contributed much to our knowledge and understanding of the history of one of the main branches of Indo-European. However, there is still a need for research into early NIA. The present paper is a modest contribution to the corpora collation and preliminary analysis of early NIA morphosyntax from various perspectives. We decided to focus on selected early NIA tongues such as Rajasthani, Awadhi, Braj, Dakkhini and Pahari, and we present here the preliminary results of research into the Rajasthani language, based on short prose texts from the 15th to 18th centuries supplemented by early Awadhi poetry[2].

[1] This paper is a part of a research project funded by Polish National Centre for Science Grant 2013/10/M/HS2/00553.

[2] Optical recognition of Rajasthani texts was supported by a Hindi OCR program [11].

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 123–136, 2018.
https://doi.org/10.1007/978-3-319-93782-3_10

2 The IA Tagger Tool

2.1 Tool Overview

IA Tagger is a tool for text annotation specifically for Indo-Aryan languages. The key functionality of the tool is multi-level annotation of words and sentences of early NIA texts (see Sect. 2.2). IA Tagger provides several features that improve the efficiency of use. For most annotation levels the system displays a context-sensitive list of prompts of available annotation tags. For a word under annotation the system displays a "prompt cloud", which consists of a set of tag suggestions (see Sect. 2.3).

IA Tagger minimizes the cost of usage errors or system failure. Each annotation decision is saved automatically in a periodically backed-up database. There is no save button.

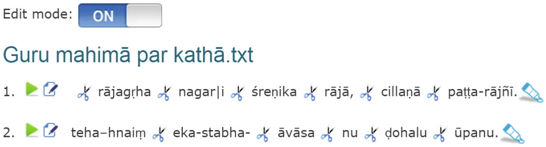
The wide variety of configuration settings ensures the flexibility of the tagger, allowing it to be used in various scenarios (see Sect. 2.4).

On request IA Tagger generates statistics concerning occurrences of specific classes of words and word collocations – in a specified document or collection of documents (see Sect. 2.5).

The system is intended for open access. It is accessible using any popular Internet browser at http://rjawor.vm.wmi.amu.edu.pl/tagging. Access credentials can be obtained on request from *rafal.jaworski@amu.edu.pl*.

2.2 Multi-level Tagging

Upon upload to the system, a document is automatically split into sentences (see Fig. 1).



Guru mahimā par kathā.txt

1. ✂ rājagṛha ✂ nagar|i ✂ śreṇika ✂ rājā, ✂ cillaṇā ✂ paṭṭa-rājñī.

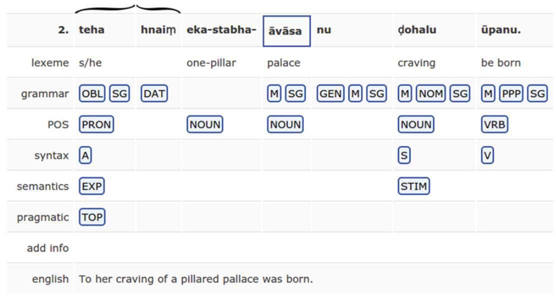
2. teha–hnaiṃ ✂ eka-stabha- ✂ āvāsa ✂ nu ✂ ḍohalu ✂ ūpanu.

Fig. 1. Sentence split.

The user can easily override the automatic sentence split (using "scissors" or "glue"; Fig. 1). The document is annotated in sentence-by-sentence mode.

Each sentence is automatically split into words. The user may override the word split, e.g. in order to divide a word into a stem and a suffix. Words are annotated at six levels: Lexeme (where the closest English lexical equivalent is given), Grammar (annotated by means of Leipzig Glossing Rules), POS (Parts of Speech), Syntax (exploring the basic Dixonian [7] scheme based on the three primitive terms: A, S and O, where A stands for the subject of a transitive sentence, S for the subject of an intransitive sentence and O for the object of a transitive sentence), Semantics (where we distinguish six basic thematic roles: Agent, Patient, Experiencer, Recipient, Stimulus and Theme, based on the RRG approach, e.g. [30]), and Pragmatics (distinguishing Topics).

Figure 2 represents an annotated sentence in Rajasthani.

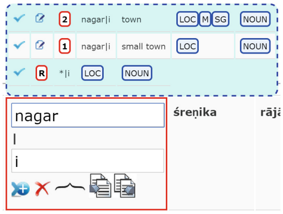


2.	teha	hnaiṃ	eka-stabha-	āvāsa	nu	ḍohalu	ūpanu.
lexeme	s/he		one-pillar	palace		craving	be born
grammar	OBL SG	DAT		M SG	GEN M SG	M NOM SG	M PPP SG
POS	PRON		NOUN	NOUN		NOUN	VRB
syntax	A					S	V
semantics	EXP					STIM	
pragmatic	TOP						
add info							
english	To her craving of a pillared pallace was born.						

Fig. 2. Annotated sentence.

2.3 Automatically Generated Suggestions

In order to improve tagging efficiency, the system suggests hints whenever possible, i.e. when a word has already been tagged or when the tagging could be deduced automatically. Tag suggestions appear in a "cloud" above the word (Fig. 3).



✓	✎	2	nagar|i	town	LOC M SG	NOUN
✓	✎	1	nagar|i	small town	LOC	NOUN
✓	R	*|i	LOC	NOUN		

nagar
|
i

śreṇika rājā

Fig. 3. Automatically generated suggestions.

Figure 3 shows tag suggestions for the word 'nagar—i' (the pipe indicates that the word 'nagari' has been split into a stem and a suffix). The first two lines come from previous annotations, whereas the third line is the set of suggestions deduced automatically. The user can accept the set of suggestions by clicking the 'check' symbol in the left-most column. The annotation shown in Fig. 2 was obtained by applying the third-line set of tags.

2.4 Configuration

IA Tagger may be configured to serve a variety of annotation tasks. The "configuration" option allows one to manage the languages of tagged documents as well as to configure annotation levels. Annotation levels may be freely ordered, added, deleted or edited. Editing of an annotation level consists in defining admissible values of respective tags.

2.5 Flexible Statistics Generator and Preliminary Results of Its Application

On request, IA Tagger generates statistics concerning occurrences of specific classes of words and word collocations – in a specified document or collection of documents. This facilitates a linguistic analysis at the levels of morphology, syntax, semantics and pragmatics. Initial analysis assumes a survey of alignment features, i.e. main argument marking (A and O). This kind of research has already been carried out by several authors for finite verbs (e.g. [13]), but IA Tagger makes it possible to generate statistics for texts belonging to various historical stages of NIA, and it has a much larger scope since it encompasses both finite and non-finite verbs such as converbs[3], infinitives and participles. Preliminary research on a Rajasthani annotated corpus shows a preponderance of unmarked O forms over marked ones. Out of 201 converbal chain constructions only 18 show marked O forms. Up to the 18th century there are only examples of animate and definite O's, or possibly human and indefinite, and from the 18th century onwards inanimate Os start appearing (see ex. (1)). Similarly, in a targeted search in an Early Awadhi text (Jāyasī's 'Padmāvat' from 1540) out of the 236 converbal chain constructions excerpted from the IA Tagger system we have found only 4 attestations of marked O's, 1 being animate and definite and 3 inanimate and definite.

(1) Old Rajasthani, 18th c. [2, 72]

tiṇa	sahanāṇa-ṇūṃ	dekha
that.OBL.SG	sign-ACC	see.CVB

'having seen that sign'

This conforms with Khokhlova's [13] findings for finites and with more general tendencies operating along definiteness and animacy hierarchies (cf. for example [1]). Morever, research on historical syntax of other early NIA tongues clearly demonstrates that O marking is rather a recent phenomenon (cf. for example [31]).

The next steps of the analysis will consist in a multilayered analysis of IA non-finites (focusing on converbs and infinitives) drawing from two frameworks: RRG [28–30] and Multivariate Analysis [3] where apart from morphological properties, syntactic semantic and pragmatic properties of converbs will be investigated.

[3] We accept here Haspelmath's [9, 3] definition of the converb: "a nonfinite verb form whose main function is to mark adverbial subordination".

Here we are going to focus briefly on control properties of converbs and scopal properties of selected operators in converbal chain constructions.

As it is the case of modern IA, converbs are predominantly controlled by the subject of the main clause (aee example (2)). However, there are single attestations of the violation of this rule. The violation of what is labelled 'the subject identity constraint' (hence SIC) has semantic or pragmatic motivation (for example, the subject of the main clause is in a possessor-like relation with the subject of the converb or it is discourse prominent - see the examples (3) and (4)). More recent, typologically oriented research by Subbarao [22] demonstrates that in IA SIC is permitted only when the subject of the converb is inanimate and the converbal clause denotes non-volitional act. As we can see from the examples (3) and (4), which has a volitional animate subject, it is not always the case of early NIA.

(2) Old Rajasthani, 15th c. [2, 13]

yakṣ-i	arjuna	ripu	bāṃdhī-karī	page	āṇi
Yaksha.INS.M.SG	Arjuna	enemy-M.NOM.SG	bind.CVB	foot	come.CVB
ghātiu					
throw.PPP.M.SG					

'Yaksha, having bound the enemy named Arjuna, threw him on his feet.'

(3) Old Rajasthani, 16th/17th c. [2, 44]

hemū [...]	pāṇīpaṃtha	āi	ḍerā	paṛiyā	chai
Hemu	Panipat	come.CVB	camp.NOM.M.PL	fall.PPP.M.PL	be.3SG.PRS

'And after that Hemu had come to Panipat, the camps were established.'

(4) Old Rajasthani, 15th c. [2, 11]

ti	puruṣa	raja-nai	vacani	karī	saṃgha-māhi
these	people	king-M.DAT.SG	speech	do.CVB	community-in
gayā					
go.PPP.M.PL					

'These men on hearing the king's speech (lit. of the king having spoken) went happy to their community.'

It seems that early NIA represents a more general South Asian type of a converb in terms of its syntactic lability - i.e. SIC violation can be permitted when we have so called *constructio ad sensum* (cf. [24]). This notion assumes a role oriented (e.g. posessor or experiencer like) relations between the converbal subject and the subject of the main clause. Actually, what we observe in (3) is a type of an absolute construction in which prototypically SIC is not observed[4]. Early NIA had absolute constructions in which the embedded clause was formed

[4] The following is a brief characteristics of an absolute constuction: "The head noun and its participle form a special type of a subordinate clause which could express an event contemporary with or anterior to that in main clause." [4].

by means of a subject modified by an inflected past participle but parallel the construction based on the converb was developed (for more detailed discussion see [25]).

Preliminary results of the analysis of scopal properties of selected operators in converbal chain constructions were presented in [21]. It has been observed that IF (Illocutionary Force) Operator can have conjunct or local scope, and this property is quite stable throughout the centuries (cf. ex. (5) and (6) from the 16 and the 17th century texts with the imperative scope local or conjunct).

(5) Old Rajasthani, 16th c. [2, 33]

paṇi	tumhẽ	mayā	karī	deśāntari	pahucaü
but	you	mercy.NOM.F.SG	do.CVB	abroad.OBL	reach.PPP.M.SG

'But you, having shown mercy, go abroad'.
'But you show mercy and go abroad'

(6) Old Rajasthani, 17th c. [2, 44]

upāṛi -ara	ghoṛā	māṃhe	ghālo
lift-CVB	horse.NOM.M.PL	in	throw.IMP.2PL

'After lifting the horses throw them into (the river).'
'Lift the horses and throw them into (the river).'

The T (Tense) Operator seems to have conjunct scope in those converbal chains which have the main verb in the past tense and almost exclusively local scope in those chains which have the main verb in the present tense (cf. ex. (7) and (8)). This finding has two important consequences. Firstly, it somehow implicitly presumes the perfectivity of the IA converb and secondly, it shows that converbal chain constructions are not characterised by the operator dependence. The operator dependence is a defining feature of the third type of linking, namely cosubordination which plays an important role in the theory of clause linkage in RRG [28]. Therefore, if the converbal chain constructions do not show the consistent operator dependence (and it is also the case of other operators such as IF-operator or NEG-operator) they are not instantiations of cosubordination. The support for such interpretations comes from other IA languages, both early and contemporary (see for a more detailed discussion see [23, 23-33]).

(7) Old Rajasthani, 15th c. [2, 15]

āmbā	leī	ḍohalu	pūriu
mango.NOM.M.PL	take.CVB	craving.M.SG	fill.PPP.M.SG

'Having taken mangos, (he) fulfilled the craving.'
'(He)took mangos and fulfilled the craving.'

(8) Old Rajasthani, 18th c. [2, 61]

phūladhārā	vica	uḍi	paṛaṃ
stream of flowers	middle.CVB	fly.CVB	fall.1PL.PRS.SBJ

'Having flown in the middle of the stream of flowers,
we shall fall'

The intermediate position of converbal chains between subordination and coordination can be investigated in terms of the operator scope which in IA may have a multiple motivation. Several authors have already tried to look at the problem from syntactic, semantic and pragmatic angles and the views are quite divergent (cf. for example [6,12]). It seems that only a more fine grained approach can bring interesting results. First attempts have already been made in synchronic and typological works by Peterson [18] and Bickel [3] and now we shall be able to extrapolate these methodologies to diachronic research.

3 Automatic POS-tagging

3.1 Similar Experiments

Experiments with automatic POS-tagging of less-resourced languages have already been conducted in recent years. This subsection briefly describes the techniques used and the outcome of two projects: an automatic tagger for Urdu, developed by [8], and Sanskrittagger by [10].

Urdu Tagger. The tagger for Urdu was developed by Andrew Hardie in 2005. The main difficulty in tagging Urdu texts identified by the author was word sense disambiguation. Two techniques were implemented in order to resolve this problem. One was based on hand-crafted rules prepared by a linguist, while the other relied on statistical analysis of manually annotated Urdu texts. The author reports the low effectiveness of the latter method, attributing it to the relatively small quantity of training data. Hence the author decided to use the tagger based on hand-crafted rules. It must be pointed out, however, that the statistical model used was HMM (Hidden Markov Models), which was considered state-of-the-art in the early 2000s, but was replaced in the following years by several other methods, such as Conditional Random Fields or Maximum Entropy.

The resulting rule-based tagger used a tagset of approximately 80 tags and achieved an accuracy of 88–90%. The author admitted that these results were lower than those of taggers for well-resourced languages, such as English. Such taggers score at least 95% accuracy. This, however, should not be considered the main flaw of this system. A more important drawback of the approach presented by Hardie is the heavy reliance on manually designed rules, which account for most of the positive results of the system. These rules were specially designed to work with Urdu, and even more specifically – with the Urdu texts that were at the author's disposal. In a different scenario the same rules may prove to be inapplicable, thus impairing the performance of the system significantly.

Sanskrittagger. Sanskrittagger, described in [10], is an automatic tokenizer and tagger for Sanskrit. Like Hardie's Urdu tagger, it uses HMM to perform the tagging. Interestingly, the same model is also applied to the task of tokenization, which is a non-standard solution.

The system uses a tagset of 136 tags. Unfortunately, accuracy figures are not known, as the evaluation of the system was performed on only five short passages of text. However, it is revealed that the system is purely statistical.

Among suggested methods of improvement, one seems particularly interesting – integrating tokenization and POS-tagging into one mechanism. The author argues that this might be a good approach for Sanskrit, even though it is not commonly used for other languages.

3.2 Training the Automated Tagger

The IA Tagger system has been used by a team of linguists for several months. The work has resulted in a manually annotated corpus of the early Rajasthani language supplemented by early Awadhi. The corpus so far contains 1284 sentences with 13 022 words. Even though the size of the corpus is too small for statistical data analysis, experiments were run to determine whether it is possible to create a usable POS tagger for early NIA.

Firstly, two separate POS tagging systems were developed. One of them uses a set of 22 tags to annotate the text. The tags are hierarchical, e.g. there is a NOUN tag and its child – NOUN-SINGULAR.

The other tagger is a detector of specific verb forms – converbs.

Multi-class POS Tagging. The task of annotation with 22 tags was seen as a multi-class classification problem. In order to implement such a tagger, a well-known Maximum Entropy tagging mechanism was used. This idea was first proposed by [19] and later used to implement the Stanford Part-Of-Speech Tagger (see [26,27]). The automatic tagger for early NIA is based on the Stanford software.

The main difficulty in training automatic taggers using the Maximum Entropy principle is the identification of the feature set. Possible features may include: suffix(n) of the word (i.e. last n letters), length of the word, whether the word starts with a capital letter (boolean feature) and many others. It is crucial, however, that all these features should be computable on unannotated text. Thus, features like "is located between a noun and a verb" are not acceptable.

The described automatic tagger for early NIA texts uses the following set of features: *Suffix(6)*, *Previous word (i.e. the literal text form of the previous word)*, *Next word* and *Distributional similarity class.*

Distributional similarity (often abbreviated *distsim*) is a method for categorizing words in a large corpus based on their contexts. Each word falls into a category with other words that appeared in similar contexts. The id of such a category can be used as a word feature.

In order to compute distributional similarity classes, an unannotated modern Rajasthani corpus of 81 843 words was used. It was processed with the help of word2vec software, described in [17]. The words were categorized into 209 classes, each containing between 1 and 66 words. For example, one of the classes contained the following words: *te* 'this', *teha* 's/he', *bi* 'two', *bewai* 'both', which are all pronouns.

Converb Detector. The second approach involved the training of a separate tagger, focused solely on identifying words of special interest – converbs. This is a case of binary classification. Two such binary converb detectors were implemented – one based on the Maximum Entropy algorithm and another one using the Vowpal Wabbit library [14].

The implementation of the converb detector using the Maximum Entropy (ME) algorithm is based on the Python NLTK library, described in [15], which is capable of using an optimization technique called MEGAM [5]. This makes it possible to create a robust binary classifier. This converb detector was trained on the same data as the multi-class tagger described in Sect. 3.2 The features used by this detector are presented in Table 1. Note that the features *cvbEnding* and $firstOrLast$ use linguistic knowledge about converbs. Firstly, Rajasthani converbs typically terminate in /i/ and /a/, although from the earliest texts onwards other suffixes are also attested. Secondly, converbs would never appear as the first or last word in the sentence. This approach recalls the hand-crafted rules as seen in [8]. However, the features are never strict. The decision on whether or not to use a specific feature is made by the statistical model.

Table 1. Features used by the ME converb detector

Feature name	Parameters	Description
word	none	literal text of the word
wordContext	n	n words to the left and n words to the right of the word
suffix	n	n last characters of the word
class	none	distributional similarity class
classContext	n	as in wordContext, only on *distsim* classes
cvbEnding	none	whether or not the word ends in a typical converb ending
firstOrLast	none	whether or not the word is first or last in the sentence

In a separate experiment, a second converb detector was build with the help of the Vowpal Wabbit (VW) software and was trained on a set of 5596 Awadhi words. All these words came from texts authored by the same person. Because of the homogeneity of the texts, we expected better evaluation results than in the previous experiments.

On the other hand, the classification algorithm used by the Vowpal Wabbit software allows is based on classic regression and features numerous improvements described thoroughly in [14]. Importantly, the software features a tool for assessing the importance of individual features in the process of prediction. The most informative features identified with the help of this tool were used in the process of classification and are presented in Table 2.

Table 2. Features used by the VW converb detector

Feature name	Description
Suffix(3)	Last three letters of the word
Context(1)	The literal forms of the previous and next word
Class-context(1)	The distributional similarity classes of the previous and next word

3.3 Experiment Results

This section presents the results of the experiment conducted using both of the automatic POS taggers. In both cases the tagged corpus (13 022 words) was used to perform 10-fold cross-validation. The magnitude of the test data complies with the standards for human evaluation experiments in the field of natural language processing (see for instance [20]).

Table 3 presents results for the multi-class tagger. It assigned tags to 10 730 out of 13 022 words (82.4%), leaving the remaining words untagged. Exact tag matching counts a tag as correct only if it matches exactly the tag in the golden standard. Partial tag matching allows, for example, the tagging of a NOUN-SINGULAR with the tag NOUN.

Table 3. Overall results of the multi-class tagger

Metric	Correct tags #	Accuracy
Exact	6210	57.9%
Partial	6874	64.1%

Some specific word forms were investigated more thoroughly. Table 4 presents precision, recall and F-measure scores (as proposed in [16]) for identifying these forms. All results assume the partial tag matching metric.

Table 4. Detailed performance of the multi-class tagger

Word form	Precision	Recall	F-measure
Verb	0.61	0.70	0.65
Noun	0.41	0.52	0.46
Past participle	0.70	0.60	0.64
Converb	0.33	0.07	0.11

The accuracy of the multi-class tagger, which was as low as 64%, was not a satisfactory result. However, the results in Table 4 reveal that even though the overall accuracy of the system is low, some word forms can be detected more accurately, such as verbs. However, converbs, the forms of our special interest, were detected poorly by the multi-class tagger. This inspired further study using the specialized converb detector.

The detector was expected to attain higher precision and recall scores in finding converbs than the multi-class tagger. The scores of the Maximum Entropy detector are presented in Table 5. These indeed show a considerable improvement over the multi-class tagger (see Table 4). This justifies the decision to implement a separate detector solely for word forms of particular interest.

Table 5. ME converb detector scores

Metric	Value
Precision	0.83
Recall	0.39
F-score	0.53

This success inspired further work on binary classification of converbs in other texts. Experiments were carried out on 5596 Awadhi words coming from a 16th century text “Padmāvat” by Malik Muhammad Jayasi. Out of all these words, 181 were annotated as converbs, which constitutes 3.2%.

First experiments with Awadhi were carried out in a 10-fold cross-validation scheme. Firstly, a baseline system was tested. The baseline built a dictionary of converbs from its training data and applied it on the test data to make the predictions. Such approach rendered the precision and recall scores of 46.7% and 57.0% respectively, which proved that the converb detection problem on the provided data is non-trivial. The best results in this scenario were achieved by the Vowpal Wabbit converb detector described above: precision of **80.2%** and recall of **64.4%**.

The VW detector trained on the whole corpus of 5596 words was then tested in another experiment. An excerpt of 11 501 words from a different part of the “Padmāvat” was tagged manually with only the converb tags and used as test data. The results of converb detection on this fragments were **74.8%** of precision and **66.4%** of recall.

Satisfactory results of the VW converb detector made it possible to use it in the following scenario. The detector was run on the whole “Padmāvat” and provided a list of words predicted as converbs. The sentences containing these words were given to human annotators for further analysis. Thus, the linguists received a set of sentences which exhibited a high probability of containing a converb. With regard to the value of the precision measure achieved in experiments, this probability can be assessed as approx. 75%. Furthermore, the value of the recall measure suggests that about two thirds of all converbs from the “Padmāvat” were successfully detected with this method. This way, the linguists obtained a large number of example sentences containing words of their interest without the necessity of manual annotation of the whole corpus, which was not feasible.

4 Conclusions and Future Work

The paper demonstrates how IA Tagger – a semi-automatic annotating tool – can help perform multi-level historical linguistic analyses pertaining to morphosyntax, semantics and pragmatics. A flexible statistics generator facilitates distributional analysis of various converbal forms and analysis of main argument marking with finite and non-finite verbs. At the semantic level it can also support analysis of the control properties of converbs, and at the pragmatic level it clearly helps establish the scope of main clause level operators.

In the future, the IA Tagger can further support research on other non-finite verb forms such as infinitives and participles, and what is more, it can easily identify the main grammaticalization paths with respect to light verbs.

The data acquired with the use of the IA Tagger enabled research on automatic POS-tagging for the early Rajasthani language. The research consisted in two experiments carried out on a small set (13 022 words) of annotated data. The first study had the aim of creating a multi-class POS classifier trained on the available data. The second investigated the accuracy of a binary classifier devoted to a class that was recognized poorly in the first experiment. The experiments applied standard machine learning techniques, including the recently investigated idea of distributional similarity classes. The evaluation results proved that applying purely statistical methods to a small corpus of annotated data does not yield practically applicable results for multi-class recognition. The binary classifiers, however, achieved satisfactory values for both the precision and recall measures.

We conclude that applying state-of-the-art machine learning techniques to languages that lack large annotated corpora may be useful for binary classification. Using such binary classifiers for searching of potentially interesting forms in a large text collection can greatly reduce the human effort in language analysis and help to obtain important linguistic findings.

References

1. Aissen, J.: Differential object marking: iconicity vs. economy. Nat. Lang. Linguist. Theory **21**, 435–483 (2003)
2. Bhanavat, N., Kamal, L.: Rajasthani gadya: vikas aur prakash. Shriram Mehra and Company, Agra (1997–1998)
3. Bickel, B.: Capturing particulars and universals in clause linkage: a multivariate analysis. In: Bril, I. (ed.) Clause Linking and Clause Hierarchy : Syntax and Pragmatics, No. 121 in Studies in Language Companion Series, pp. 51–102. John Benjamins, Amsterdam (2010). https://doi.org/10.5167/uzh-48989
4. Bubeník, V.: A historical syntax of late Middle Indo-Aryan (Apabhramśa). Amsterdam studies in the theory and history of linguistic science: Current issues in linguistic theory. John Benjamins, Amsterdam (1998). https://books.google.pl/books?id=abJjAAAAMAAJ

5. Daumé III, H.: Notes on CG and LM-BFGS optimization of logistic regression, August 2004
6. Davison, A.: Syntactic and semantic indeterminacy resolved: a mostly pragmatic analysis for the hindi conjunctive participle. In: Peter, C. (ed.) Radical pragmatics, pp. 101–128. Academic Press, New York (1981)
7. Dixon, R.M.: Ergativity. Cambridge Studies in Linguistics. Cambridge University Press, Cambridge (1994). https://books.google.pl/books?id=fKfSAu6v5LYC
8. Hardie, A.: Automated part-of-speech analysis of Urdu: conceptual and technical issues. Contemporary Issues in Nepalese Linguistics, pp. 48–72 (2005)
9. Haspelmath, M.: The converb as a cross-linguistically valid category. In: Haspelmath, M., König, E. (eds.) Converbs in cross-linguistic perspective: structure and meaning of adverbial verb forms - adverbial participles, gerunds, pp. 1–55. No. 13 in Empirical approaches to language typology, Mouton de Gruyter, Berlin (1995)
10. Hellwig, O.: A stochastic lexical and POS tagger for sanskrit. In: Huet, G., Kulkarni, A., Scharf, P. (eds.) ISCLS 2007-2008. LNCS (LNAI), vol. 5402, pp. 266–277. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00155-0_11
11. Hellwig, O.: ind.senz - OCR software for Hindi, Marathi, Tamil, and Sanskrit (2015). http://www.indsenz.com
12. Kachru, Y.: On the syntax, semantics and pragmatics of the conjunctive participle in Hindi-Urdu. Stud. Linguist. Sci. **11**(2), 35–49 (1981)
13. Khokhlova, L.: Ergativity attrition in the history of western new Indo-Aryan languages (Punjabi, Gujarati and Rajastahani). The Yearbook of South Asian Languages and Linguistics, pp. 159–184 (2001)
14. Langford, J., Li, L., Zhang, T.: Sparse online learning via truncated gradient. In: Advances in Neural Information Processing Systems, pp. 905–912 (2009)
15. Loper, E., Bird, S.: NLTK: The natural language toolkit. In: Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, ETMTNLP 2002, vol. 1. pp. 63–70. Association for Computational Linguistics, Stroudsburg, PA, USA (2002). https://doi.org/10.3115/1118108.1118117
16. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Proceedings of DARPA Broadcast News Workshop, pp. 249–252 (1999)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013). http://arxiv.org/abs/1301.3781
18. Peterson, J.: The Nepali converbs: a holistic approach. In: Singh, R., Dasgupta, P. (eds.) The Yearbook of South Asian Languages and Linguistics (2002), pp. 93–134. Walter de Gruyter, Berlin (2002)
19. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 16 April 1996
20. Tou, J.T.: Information systems. In: von Brauer, W. (ed.) GI 1973. LNCS, vol. 1, pp. 489–507. Springer, Heidelberg (1973). https://doi.org/10.1007/3-540-06473-7_52
21. Stroński, K., Tokaj, J.: The diachrony of cosubordination - lessons from Indo-Aryan. In: Proceedings of the 31st South Asian Languages Analysis Roundtable (SALA-31), pp. 59–62 (2015). Extended abstract. http://ucrel.lancs.ac.uk/sala-31/doc/ABSTRACTBOOK-maincontent.pdf
22. Subbārāo, K.: South Asian languages. A Syntactic Typology. Cambridge University Press, New York (2012). https://books.google.pl/books?id=ZCfiGYvpLOQC

23. Tikkanen, B.: The Sanskrit gerund: a synchronic, diachronic, and typological analysis. Studia Orientalia, Finnish Oriental Society (1987). https://books.google.pl/books?id=XTkqAQAAIAAJ
24. Tikkanen, B.: Burushaski converbs in their south and central Asian areal context. In: Haspelmath, M., König, E. (eds.) Converbs in cross-linguistic perspective: structure and meaning of adverbial verb forms - adverbial participles, gerunds. (Empirical approaches to language typology 13.), pp. 487–528. Mouton de Gruyter, Berlin (1981)
25. Tokaj, J.: A comparative study of participles, converbs and absolute constructions in Hindi and medieval Rajasthani. Lingua Posnaniensis, pp. 105–120 (2016)
26. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL, pp. 252–259 (2003)
27. Toutanova, K., Manning, C.D.: Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), Hong Kong, pp. 63–70, October 2000
28. Van Valin, R.D., LaPolla, R.J.: Syntax: Structure, Meaning, and Function. Cambridge University Press, Cambridge (1997)
29. Van Valin, R.J.: A synopsis of role and reference grammar. Advances in Role and Reference Grammar, pp. 1–164 (1993)
30. Van Valin, R.J.: Exploring the Syntax-Semantics Interface. Cambridge University Press, Cambridge (2005)
31. Wallace, W.D.: Object-marking in the history of Nepali: a case of syntactic diffusion. Stud. Linguist. Sci. **11**(2), 107–128 (1981)

Multilingual Tokenization and Part-of-speech Tagging. Lightweight Versus Heavyweight Algorithms

Tiberiu Boros(✉) and Stefan Daniel Dumitrescu(✉)

Research Institute for Artificial Intelligence, Romanian Academy,
Bucharest, Romania
{tibi,sdumitrescu}@racai.ro
http://www.racai.ro

Abstract. This work focuses on morphological analysis of raw text and provides a recipe for tokenization, sentence splitting and part-of-speech tagging for all languages included in the Universal Dependencies Corpus. Scalability is an important issue when dealing with large-sized multilingual corpora. The experiments include both lightweight classifiers (linear and decision trees) and heavyweight LSTM-based architectures which are able to attain state-of-the-art results. All the experiments are carried out using the provided data "as-is". We apply lightweight and heavyweight classifiers on 5 distinct tasks, on multiple languages; we present some lessons learned during the training process; we look at per-language results as well as task averages, we present model footprints, and finally draw a few conclusions regarding trade-offs between the classifiers' characteristics.

Keywords: Linear models · Neural networks
Long-Short-Term-Memory (LSTM) networks · Decision trees
Sequence labeling · Part-of-speech tagging · Morphological attributes
Tokenization · Sentence splitting

1 Introduction

Tasks such as sentiment analysis, machine translation, text-to-speech synthesis and information extraction require a certain level of text preprocessing aimed at segmenting the input into standard processing units (often into sentences and words but, depending on the application, also syllables, phonemes etc.) and at enriching these units with additional features designed to reduce the effect of data sparsity (lemmas, part-of-speech tags, morphological attributes etc.). Because this is a basic requirement, the literature is abundant with methods and techniques for low-level text processing, but **multilingual text-processing is still a challenging task**. This has been proven by the well-known shared task on Universal Dependencies (UD) parsing [12]. One very important conclusion is that while some algorithms have an overall better performance than others - and

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 137–151, 2018.
https://doi.org/10.1007/978-3-319-93782-3_11

we draw the attention to Stanford's [5] graph-based parser, there is **no "one size fits all" algorithm** that is language and corpora-size independent.

While accuracy carries a great weight in NLP applications, there are two other factors that impact the design of real-world systems: computational cost and memory footprint. This paper focuses on providing a comparison between two different types of algorithms applied on the same set of text processing tasks: Lightweight (decision trees and linear models) and Heavyweight (neural networks, more specifically bidirectional long-short-term-memory (LSTM) networks). We motivate our choice based on the computational/memory requirements of these algorithms:

Lightweight algorithms:

- **Decision trees** require virtually **no feature-engineering**, provide a **relatively small model footprint**, with a **logarithmic computational complexity** ($O(\log n)$), where n is the number of unique features and **low mathematical load**;
- **Linear models** require **feature-engineering**, yield models with **larger footprints**, with **linear computational complexity** ($O(n)$) and a **moderate mathematical load** (commonly multiplications and additions);

Heavyweight algorithms:

- **Neural networks** are able to learn patterns, yield **small footprint models** (even with compact feature embeddings), but generate a **high computational load**, mainly because of the large number of operations and the use of **complex mathematical functions** (multiplications, additions, tanh or σ activation functions).

In what follows, we use the Universal Dependencies (UD) data [7] to compute accuracies and model footprints obtained using the three machine-learning (ML) algorithms mentioned above. All our results are reported using the provided data "as is", without generating any alterations in the constituency of the training, development and test sets. Depending on the ML algorithm, we describe our strategy and feature engineering process and we try to keep the comparison as fair as possible, without introducing language-dependent tweaks that would provide one algorithm leverage over others. As expected, the LSTM-based models performed significantly better that their counterparts (in fact our architecture surpassed state-of-the-art results for some languages in the UD Shared Task), but their computational cost and storage footprint make them less appealing when one wants to host processing pipelines for many languages. Currently we host a web-service[1] that enables users to query our processing pipeline for all languages available in UD, but for the previously mentioned reason, we only use the lightweight version of the models (driven by linear and decision trees classifiers and Viterbi-decoding for POS tagging).

Before the actual description of the models and algorithms we wish to provide a context and to describe our configurable NLP processing system which was

[1] http://slp.racai.ro/index.php/mlpla-new/.

firstly introduced in our earlier work [11]. Our system is called Modular Language Processing for Lightweight Applications (MLPLA) and it provides a framework for building applications which are able to run on any type of device, from mobile phones to desktops and dedicated servers. This is achieved by letting the user control the trade-off between accuracy and computational/memory cost and providing him with a choice between various models and algorithms that perform the same task. Given that we are able to easily interchange between models/modules and classifiers, we focused our efforts into assessing what is the best trade-off between speed, accuracy and model size.

2 Corpora and Task Description

The training data used is part of the Universal Dependencies release 2.0 [7] and a comprehensive description and extensive comparison of our results can be easily done by investigating the CONLL Shared Task Proceedings [13]. Our experiments are performed on **44 languages**: Ancient Greek (up to year 1453), Arabic, Basque, Bulgarian, Catalan, Chinese, Church Slavic, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, Galician, German, Gothic, Hebrew, Hindi, Hungarian, Indonesian, Irish, Italian, Japanese, Kazakh, Korean, Latin, Latvian, Lithuanian, Modern Greek (1453–), Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Slovak, Slovenian, Spanish, Swedish, Tamil, Turkish, Uighur, Ukrainian, Urdu and Vietnamese.

Because we are dealing with a large number of languages (training sets) we consider that our evaluation has the ideal ecosystem for a fair comparison of the machine learning algorithms, which will be later-on described. That is: (a) there is a **high variation in the size of training datasets**; (b) we compare these algorithms using a **large number of text processing tasks** (tokenization, sentence splitting, POS tagging and morphological attribute resolution); (c) we are able to **asses the robustness of the algorithms** in detecting and extracting patterns in the data (morphological analysis is highly language-dependent: (i) we have large versus small tagsets, (ii) languages which only require suffix analysis for morphological attribute resolution and languages in which morphological attributes depend on inter-character patterns; (iii) various writing systems such as alphabetic, abjad and syllable-based logographic).

Our experiments include all basic text processing involved in typical NLP applications that use raw text as input:

- **Tokenization**: It involves splitting each sentence into the smallest composing units, later used in analysis. The UD data also introduces the idea of compound word expansion, which requires the system to automatically divide a word into smaller units which, in some cases, requires more than just determining a split point inside the original word (see Subsect. 4.1 for details);
- **Sentence splitting**: This represents the basic task of splitting a text into sentences. Though at first sight this should be a trivial matter, punctuation does not always clearly indicate the end of a sentence, since there are many

cases where it is used to mark acronyms, abbreviations, real-numbers, dates, etc. (see Subsect. 4.1 for details);

- **Part of speech tagging**: Is probably one of the best known NLP tasks, in which words/tokens inside an utterance are labeled with part of speech information using a predefined set of tags, referred to as a "tagset inventory". Here we cover both the UPOS task (Universal Part of Speech) and XPOS task (language-dependent part of speech) (please see Subsect. 4.2 for details);
- **Morphological attribute resolution**: Though morphological attributes are usually embedded inside the labels of the previously mentioned "tagset inventory", in the case of UD, information such as gender, case, number etc. (which is language dependent) is a stored in a different annotation layer and represents a distinct task;

Overall, we cover the following five tasks: **Tokenization**, **Sentence Splitting**, **UPOS**, **XPOS** and **Attribute** resolution.

3 Algorithms Description

Decision-tree based classifiers work by creating a tree in which nodes are "questions" regarding attribute values, arcs represent decisions taken based on the "answer" to the questions and leafs are the labels. More specifically, we are using the ID3-based decision trees [9], with o modified tree construction algorithm for computational efficiency [2].

During training, trees are build based on the maximum information gain (IG) criterion (Eq. 2): given a set of training data, the ID3 algorithm determines the feature that best splits the set into two smaller subsets (one in which the feature is present and one in which the feature is absent), in order to minimize the "chance" of a high entropy (Eq. 1).

$$H(S) = -\sum_{x=1}^{N} P_x \cdot \log_2 P_x \tag{1}$$

$$IG_i(S) = H(S) - \sum_{t \in T} (P(t) \cdot H(t)) \tag{2}$$

$P(t)$ is the probability to be in subset t and $H(t)$ is the entropy of that subset, whereas $IG_i(S)$ if the information gain of feature i.

Typically **linear models** work by introducing a linear transform between input features and output labels. The linear transform is a 2D matrix, where the number of columns is given by the number of unique input features (n) and the number of rows is the number of unique output labels (m). The output of the model $y_{\overline{1,m}}$ is computed as a weighted sum (let $w_{\overline{1,m},\overline{1,n}}$ be the weights) over all input features $x_{\overline{1,n}}$ (Eq. 3). The output label is computed by searching for the largest value of $y_{\overline{1,m}}$ and returning the corresponding tag.

$$y_k = \sum_{i=1}^{n} (w_{k,i} \cdot x_i) \tag{3}$$

Given the target output vector $t_{\overline{1,m}}$, during training, the model parameters (weights) are updated using the $\delta - rule$ and a predefined learning rate α[2] (Eq. 4). Most importantly, the linear model is able to produce good results if the input features are linearly independent and separable. As such, in practice it is required to enhance the feature-set by manually introducing feature combinations that satisfy these conditions. Also, whenever dealing with sequence labeling, it is a common practice to some form of dynamic programming (for instance Viterbi) in selecting the optimal label sequence.

$$w_{i,j} = w_{i,j} + \alpha \cdot (t_i - y_i) \cdot x_j \tag{4}$$

Neural networks are known as universal approximators, in the sense that, in theory, they are able to model any input function $x(n)$ into any desired output function $y(n)$. Long-Short-Term-Memory (LSTM) networks [6] are particularly interesting because they are able to capture long-range dependencies inside sequences of data. LSTMs work by letting the input vector x model an internal memory cell (update, input, forget) as well as the output of the LSTM itself. A single LSTM cell works by "seeing" the input sequence one frame at a time and outputting a value based on the previous cell-state and the current input. Given an input sequence $x_{1,n}$, at time step t the cell has already seen all input frames from 1 to t. However, in most cases the entire sequence $x_{1,n}$ is known a priori and it is possible to introduce another cell that "sees" the sequence in reverse from n to t. As such, this becomes a bidirectional LSTM (BLSTM) and in practice produces superior results to those of an unidirectional LSTM.

4 Experiments and Results

4.1 Tokenization and Sentence Splitting

The tokenization process decomposes a sentence into smaller units called tokens, which are later used in all the following processing steps. Tokens are either punctuation, words or sub-word units for some languages in UD. For this particular task, we train and compare a DT algorithm as well as a BLSTM. While for the BLSTM the input/output is quite simple (input: one character at a time, output: decision to split at this character or not), for DTs the task is a bit more involved. Formally, the blank character is always a tokenization indicator. However, for every character in the tokens themselves, the DT tokenization algorithm has to decide if it should perform an additional split, in order to cope with clitics (e.g. for Romanian, the token "a-mi" must be split into the tokens "a" which is a verb and token "-mi" which is short for the pronoun "îmi"), agglutinative words or other language or corpora-specific phenomena. Based on the input

[2] In most of our experiments we set $\alpha = 10^{-4}$.

features (detailed later), we trained the classifier to choose between 4 classes: SPLIT_LEFT, SPLIT_RIGHT, SPLIT_LEFTRIGHT and NONE, meaning it should split to the left of the current character, right, to put two spaces around the character or not make a split at all. For example, in Romanian, the word "într-o" should be split on the dash character with a SPLIT_RIGHT as "într" and "-o" while for "ducându-se" the dash should be assigned a SPLIT_LEFT to create tokens "ducându-" and "se" (Table 1).

Table 1. Tokenization and sentence splitting accuracies for all languages using Decision Trees (DT), Neural Network (NN) BLSTM classifiers

Language	Tokenization		Sentence Split		Language	Tokenization		Sentence Split	
	DT	NN	DT	NN		DT	NN	DT	NN
Anc. Greek	99.98	99.99	98.70	98.81	Italian	99.92	99.91	98.45	98.96
Arabic	99.98	99.97	60.50	56.52	Japanese	87.57	90.67	93.18	95.02
Basque	99.98	99.99	99.83	99.89	Kazakh	94.52	95.41	81.50	78.50
Catalan	99.96	99.97	98.95	99.32	Korean	99.73	99.80	93.05	93.15
Bulgarian	99.91	99.91	92.83	92.90	Latin	99.97	100.00	99.20	99.25
Chinese	88.91	89.75	98.19	97.50	Latvian	99.30	99.53	93.30	94.64
Church Slav.	99.96	100.00	36.05	38.24	Greek	99.83	99.86	87.56	90.10
Croatian	99.84	99.86	95.91	96.82	Norwegian	99.88	99.86	93.11	94.83
Czech	99.96	99.98	82.83	85.24	Polish	99.97	99.99	99.59	99.41
Danish	100.00	100.00	76.85	82.17	Portuguese	99.64	99.61	89.79	90.36
Dutch	99.85	99.81	74.52	75.00	Romanian	99.54	99.66	92.60	96.53
English	98.67	98.92	73.22	71.78	Russian	99.94	99.10	95.30	96.43
Finnish	99.46	99.63	86.05	89.37	Slovak	100.00	99.98	83.53	83.56
French	99.72	99.76	92.12	94.69	Slovenian	99.96	99.95	99.24	97.83
Galician	99.92	99.97	95.92	95.74	Spanish	99.94	99.92	94.17	94.41
German	99.44	99.70	76.80	78.47	Swedish	99.59	99.91	89.87	95.69
Gothic	100.00	99.99	27.85	26.69	Estonian	99.85	99.75	92.53	93.12
Hebrew	99.98	99.94	100.00	99.59	Turkish	99.76	99.79	95.77	96.72
Hindi	100.00	100.00	99.20	99.17	Uighur	98.57	97.96	68.17	70.16
Hungarian	99.70	99.80	89.75	95.87	Ukrainian	99.65	99.85	94.25	92.10
Indonesian	99.99	100.00	88.41	92.27	Urdu	100.00	100.00	96.83	97.94
Irish	99.73	99.67	96.69	97.79	Vietnamese	82.47	81.97	92.59	92.21

Normally, the system should decide if a split should occur before and/or after every character inside an utterance. However, this is a very computational expensive task and, in practice, not every character is a likely candidate. As such, we initially look for specific characters where we might have a word split, marked in the UD train file by "SpaceAfter=No"; we then normalize the frequency of these characters; iterating again on the training data, we choose only the character that is most probable to initiate a split, based on the normalized frequency; for sentence splitting we perform the same process, the only difference being we preseed the character list with a number of punctuation characters like: *.-!?* etc., because we had cases where the question mark character *?* was not frequent enough to remain in the split list, though it was a valid token splitter. Another small optimization worth mentioning is that we replaced all digits with zeroes to reduce some variability in the training data.

For the DT we extracted the following features for each split character: current letter, 3 characters before and after (4 for sentence splitting), and for the previous and next words a marker whether or not the word is punctuation only, if it ends with punctuation, if it contains punctuation, if it is uppercase, if it is capitalized and the number of periods in the word (we refer to these features as *"context features"*);

For the BLSTM we create an embedding randomly initialized for every character/symbol in the training data and then run a 2-layer stacked BLSTM followed directly by a Softmax output layer. Each BLSTM layer has 100 cells. We use a 0.1 standard-dev noise on the embeddings at training time. For both training and prediction we use sequences of approx. 1 K characters.

The sentence splitting models are build using features identical to tokenization, but the label-set is reduced to only two labels for DTs: SPLIT and NONE. For the BLSTM we perform joined tokenization and sentence splitting: we append an "X" output marker to each character where a sentence should end. This takes the total number of classes to be predicted by the BLSTM to 3: "O" means do nothing, "S" means split after this character, and "SX" means split and also mark a sentence end". As the "X" is appended, note that there cannot be an "OX" class as we cannot have a sentence end without a token end. To exemplify, the sentence "One." will be encoded at train time as "O" "O" "S" "SX", meaning that letter e is where a token should end ("S") and the . is itself a token and also a sentence end ("SX").

4.2 Part-of-speech Tagging

As opposed to classical POS tagging corpora, in the UD corpus each word is labeled with three different classes of labels:

- **UPOS**, referred to as a "universal part-of-speech" [8], is a set composed of 17 unique labels which correspond to coarse part-of-speech categories: adjective, adposition, adverb, auxiliary, coordinating conjunction, determiner, interjection, noun, numeral, particle, pronoun, proper noun, punctuation, subordinating conjunction, symbol, verb and other;
- **XPOS**: represents the set of "language dependent" parts-of-speech (meaning each language has its own set, sometimes significantly larger than the UPOS set, other times inexistent marked in the results table with a *n/a*)
- **Attributes**, a set of labels complementary to the UPOS labels, encode morphological attributes in the form of key-value pairs. Examples of morphological attributes: refer to: Pronoun Type, Gender, Verb Form, Numeral Type, Animacy, Mood, etc.

The three label classes (UPOS, XPOS and attributes) are highly correlated, in the sense that certain attribute classes are only valid for specific UPOS labels (i.e. gender is specific to adjectives, nouns and pronouns, while verb from, tense and voice are specific to verbs) and, as a rule-of-thumb, XPOS labels usually correspond to unique UPOS-attributes combinations. Also, part-of-speech tagging is a classical sequence labeling task and it is not well suited for DTs alone. In our experiments we found that, while it is better to treat each label-set independently when using BLSTMs, an alternative approach must be used for the lightweight classifiers. As such, we compare the same BLSTM architecture (see Fig. 1) trained on the three different label-sets with a combination of classifiers trained to act as a "tiered" classifier [10].

It is important to mention that morphological attributes are key-value pairs which could be learned in a multi-task fashion. In fact, in our previous experiments we attempted to predict each morphological attribute separately using multiple models and multi-task learning. However, all the results pointed that higher accuracies are obtained by predicting all attributes at once as if they were a monolithic label. A possible explanation for this is that treating each attribute type independently from the others masks the importance of inter-attribute-type dependencies.

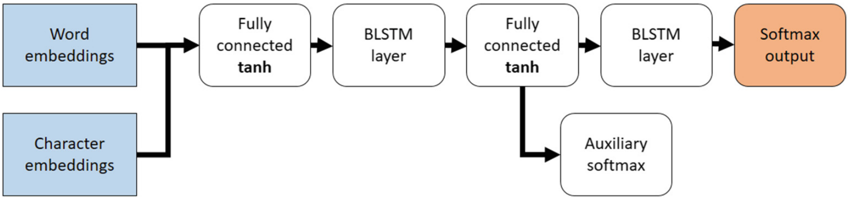


Fig. 1. Architecture of the tagging network

The BLSTM Classifier

Figure 1 shows the architecture of the tagging neural network. The input (gray boxes) is obtained by concatenating two types of embeddings: (a) word embeddings obtained using Facebook's fast-text [1] and (b) character level embeddings which are computed using a separate network (described later in this section). To prevent overfitting, we use the dropout mechanism described in [4], in which we do not mask individual neurons but instead mask the whole embeddings (word/character) with a probability of 33%, and we scale the remaining embeddings to compensate for the missing ones. In all our experiments this type of dropout has the highest impact on the generalization capacity of the network. Intuitively, the network learns robust features by trying to output the word's tag by using either type of input embeddings and, when both are dropped, it has to rely on contextual information (surrounding words) to guess the current word's tag.

The input is fully connected to a dense layer with a *tanh* activation on which we apply a standard dropout with 50% drop probability for each unit. Again, during training the activations of all non-masked neurons are scaled by 2 to cope for the missing activations. The dense layer serves as input for a BLSTM layer. The output of the BLSTM layer is then fed into another dense *tanh* layer with the same dropout probability of 50%. At this point we split the network and the activation of the second dense layer is fed into another BLSTM layer and an auxiliary softmax layer. Finally, the second BLSTM layer is connected to the final softmax layer which is used for classification at runtime.

In our experiments we used a layer size of 128 for the two dense layers (yellow boxes) and 64 LSTM units in both direction (a total of 128) for the two BLSTM layers (green boxes). No dropout is applied on the output of the BLSTM or on the internal cell states. During training both the main and the auxiliary softmaxes are trained to output the correct labels and we found best to use a weight of 0.2 for the auxiliary loss function.

The character embeddings are obtained by training a network with one LSTM layer and a softmax layer to output the correct tag for every word based on its composing characters. We pre-train the character embeddings network for 20 epochs by taking each word and feeding each character from the first to the last to the LSTM layer and then using the output of the final states of the LSTM layer as input to the softmax layer. The state-cells themselves are later used as character-level embeddings and we found that using 128 cells provides sufficient support for exceeding current state-of-the-art results for UD tagging. No dropout is applied for the character-level network and, it is worth mentioning that for some highly inflectional languages, this network alone achieves accuracies up to 93% for UPOS tagging on the devset and 90% for XPOS tagging and morphological attribute resolution.

4.3 Important Notes About Tagging

- **Using Adam or Adagard** optimizers systematically **provided sub-optimal results** as opposed to classical Stochastic Gradient Descent;
- **Removing the auxiliary loss function** slightly overfitted the training data and **decreased the accuracy** on the development set;
- In general **dropout inside the network was beneficial**, as opposed to no-dropout, but were unable to identify good insight on the hyper-parameters;
- **The input dropout** (full-masking of input vectors for different feature classes) proposed in both [5] **carried a great weight** over the robustness of the systems - in all our experiments no other hyper-parameter tweaks were able to cope with the absence of input dropout;
- For some corpora, smaller LSTM sizes slightly improve the generalization capacity of the tagger, but we found that **64 cells in both directions provide stable results over the entire corpora**;
- Though UPOS provides important information for XPOS and morphological attribute resolution, we found that simply **modeling each tagset independently yields higher accuracy values** when processing raw text with BDLSMs, mostly because we avoid propagating errors from UPOS tagging to XPOS and attribute tagging;

- Though intuitively a sound solution, **multi-task learning applied to morphological attribute resolution yields sub-optimal solutions** - thus we preferred to treat each attribute set as if it were a distinct tag in the tagset.

Tiered classifiers. Linear classifiers combined with Viterbi decoding for sequence labeling are a mainstream choice in the literature for part-of-speech tagging. However, a well-known issue is generated by data-sparsity which is abundant in the case of XPOSes and monolithic attribute labels. One way to reduce the effect of sparsity is to divide the tagging task into simpler labeling problems which are handled by layers of classifiers. Each layer works on different ambiguity sub-classes of the original label-set until the top-level layer is able to output the unique label. The labels assigned by lower-level classifiers are used as features by the higher layers.

Considering the dataset and labeling conditions, we can assume that UPOS tagging is a standalone process which can be performed apriori to the XPOS and morphological attribute resolution. As such, our architecture is composed of two layers: (a) the first layer solves the UPOS ambiguity class and (b) the second layer uses two classifiers that treat XPOS and monolithic morphological attributes independently from one another, but use the UPOS label as an input feature. While Viterbi decoding is crucial for UPOS labeling, we found that using DTs as top-layer classifiers provide good results while also keeping computational cost and model footprint to a minimum.

For each classifier we define a different, but "comparable" feature-set. Firstly, we must mention that all the classifiers share an identical feature called "writing style", which is defined at word-level and takes 4 values: (a) *ALL-CAPS* - the word is written in CAPS; (b) *ALL-LOWER* - the word contains only lower-cased symbols; *F-UPPER* - the word starts with a capital letter and all other symbols are lower-cased and (d) *F-UPPER-START* - similar with F-UPPER, only this time the word is also the first token of the sentence. Next, we define individual feature sets as:

- **Decision Trees:** For each training example (token) i the featured set is obtained by extracting the "writing style", the first four and last four characters and the lower-cased wordforms (see below for more details) for every token inside h-sized window[3] centered on the current token i;
- **Linear classifier:** Aside from the wordform and "writing style" of a token we include the localized feature set containing a large number of character combinations (90), including **cross-word letter n-grams**. The template for character combinations was manually obtained using a trial-and-error process and the character n-gram size n ranges from 2 to 5;
- **LSTMs:** Similarly to DTs, for every word inside the utterance we build a localized feature-set composed of the lower-cased wordform, the "writing style", the first four and the last four characters of that word. To avoid input sparsity, we project these features into a 64-dimensional space using a similar

[3] After a number of tests, we fixed $h = 5$ for all languages.

Table 2. Part-of-speech tagging accuracy for all languages using Linear (Lin) and BLSTM (NN) classifiers for UPOS tagging and Decision Trees (DT) and BLSTM (NN) for XPOS tagging

Language	UPOS		XPOS		Language	UPOS		XPOS	
	Lin	NN	DT	NN		Lin	NN	DT	NN
Anc. Greek	**86.48**	83.39	72.68	75.84	Italian	96.56	97.85	95.74	97.66
Arabic	90.01	94.54	81.96	89.58	Japanese	84.82	84.99	n/a	n/a
Basque	92.44	93.81	n/a	n/a	Kazakh	56.49	55.60	54.94	54.01
Catalan	97.60	98.67	97.60	98.63	Korean	91.21	94.97	81.02	91.29
Bulgarian	97.20	98.03	91.45	94.85	Latin	84.58	85.74	63.18	71.15
Chinese	82.81	90.18	82.50	89.96	Latvian	88.62	90.97	71.51	76.68
Church Slav.	94.06	94.16	94.28	94.02	Greek	95.78	95.76	95.78	96.73
Croatian	95.80	96.32	n/a	n/a	Norwegian	96.04	97.34	n/a	n/a
Czech	98.07	98.32	88.31	92.21	Polish	95.45	96.06	77.72	85.72
Danish	94.62	95.61	n/a	n/a	Portuguese	96.01	97.69	73.24	76.49
Dutch	91.19	92.99	85.67	89.32	Romanian	96.02	97.42	94.11	96.47
English	92.76	95.28	91.67	94.71	Russian	95.15	95.47	94.38	95.27
Finnish	92.56	95.14	93.77	96.03	Slovak	94.11	93.66	71.55	78.46
French	94.95	97.58	n/a	n/a	Slovenian	96.24	96.78	83.08	88.59
Galician	96.11	97.30	95.15	96.71	Spanish	95.21	96.88	n/a	n/a
German	91.02	93.69	91.27	95.58	Swedish	94.63	96.10	87.70	94.07
Gothic	93.39	94.37	93.13	95.12	Estonian	87.53	89.32	89.48	91.10
Hebrew	95.80	92.57	99.84	92.19	Turkish	89.30	93.93	87.31	93.52
Hindi	95.94	96.32	95.05	95.37	Uighur	72.79	79.63	74.73	81.78
Hungarian	90.52	92.10	n/a	n/a	Ukrainian	88.58	88.33	65.35	71.41
Indonesian	92.71	100.00	99.99	92.27	Urdu	92.12	93.69	89.64	92.15
Irish	88.95	99.67	83.25	97.79	Vietnamese	71.23	88.92	50.35	85.54

approach as [3]. For this set of experiments we used two chained layers of 128 bidirectional LSTMs, followed by the output Softmax layer (Table 2).

To prevent overfitting and obtain a robust model for out-of-vocabulary (OOV) words, we only include a wordform as a feature, if that word's occurrence frequency is higher than a threshold (k) in the training data[4] (Table 3).

5 Model Statistics

Table 4 shows the model sizes for each task for each different algorithm. The values were obtained by calculating, in turn, the average, standard deviation,

[4] In our experiments we observed that $k = 10$ is a good choice for many of the languages we used for tunning.

Table 3. Morphological attribute resolution accuracy for all languages using Decision Trees (DT) and BLSTM (NN) classifiers

Language	Attribute		Language	Attribute	
	DT	NN		DT	NN
Ancient Greek	82.01	90.67	Italian	96.07	97.04
Arabic	82.19	89.67	Japanese	87.55	92.56
Basque	83.06	88.23	Kazakh	40.50	46.23
Catalan	96.16	98.72	Korean	99.36	99.36
Bulgarian	93.22	95.23	Latin	66.91	73.21
Chinese	87.65	90.01	Latvian	79.36	86.91
Church Slavic	81.80	81.68	Greek	85.82	92.43
Croatian	79.94	87.59	Norwegian	90.91	97.43
Czech	87.10	92.38	Polish	78.02	87.11
Danish	91.92	95.12	Portuguese	93.05	95.36
Dutch	86.84	89.78	Romanian	94.26	97.11
English	93.26	95.09	Russian	79.42	87.23
Finnish	86.10	88.69	Slovak	72.23	81.41
French	93.56	96.81	Slovenian	83.48	90.03
Galician	99.78	99.78	Spanish	94.57	97.02
German	77.37	82.33	Swedish	91.05	95.17
Gothic	83.20	89.56	Estonian	79.04	86.33
Hebrew	78.96	85.67	Turkish	78.93	91.78
Hindi	87.61	93.46	Uighur	98.57	99.53
Hungarian	68.31	71.22	Ukrainian	65.41	75.12
Indonesian	99.49	99.55	Urdu	80.26	83.71
Irish	72.95	81.13	Vietnamese	82.24	88.76

minimum and maximum size for all languages. We observer very small sizes for the DT models, almost 100 times larger models for the neural networks, and very large models for the Linear models.

The large size of the Linear models is explained by the fact that this type of classifier requires carefully crafted feature combinations, which, in our case, take the form of letter n-grams. For computational reasons, each unique character n-gram is kept as a n-sized vector, where n is equal to the number of output labels. In our case, the number of labels is given by the size of the UPOS tagset and we perform pruning to prevent over-fitting of the model. Still, we use 88 templates for building our letter n-grams which span over the first four and last four characters of each word, that, for highly inflectional languages such as Czech yield a large number of frequent unique features, giving model sizes in the hundred MB range (e.g. the Czech model's size is 612 MB).

Table 4. Model size statistics on the 5 tasks, using all classifiers (DT = Decision Trees, Lin = Linear classifier, NN = Neural Networks with BLSTMs).

Task	Classifier	Avg	Stdev	Min	Max
Tokenization	DT	18 KB	71 KB	0.7 KB	394 KB
	NN	9.43 MB	35 MB	1.77 MB	236 MB
Sentence splitting	DT	21 KB	53 KB	1 KB	345 KB
	NN	9.43 MB	35 MB	1.77 MB	236 MB
UPOS	Linear	121 MB	100 MB	2.97 MB	612 MB
	NN	1.08 MB	3.54 KB	1.06 MB	1.08 MB
XPOS	DT	110 KB	200 KB	10.6 KB	1.09 MB
	NN	1.47 MB	0.82 MB	1.05 MB	5.93 MB
Attributes	DT	170 KB	219 KB	10.5 KB	1.34 MB
	NN	1.59 MB	0.85 MB	1.12 MB	6.44 MB

Table 5. Model accuracy statistics on the 5 tasks, using all classifiers (DT = Decision Trees, Lin = Linear classifier, NN = Neural Networks with BLSTMs).

Task	Classifier	Avg%	Stdev%	Min%	Max%
Tokenization	DT	98.73	3.53	82.47	100
	NN	**98.84**	3.32	81.97	100
Sentence splitting	DT	87.83	15.18	27.85	100
	NN	**88.74**	15.40	26.69	100
UPOS	Linear	90.98	7.73	56.49	98.07
	NN	**93.11**	7.17	55.6	100
XPOS	DT	84.39	12.28	50.35	99.99
	NN	**88.84**	9.63	54.01	98.63
Attributes	DT	84.30	10.96	40.5	99.78
	NN	**89.16**	9.52	46.23	99.78

For the neural networks the model footprints vary. The Tokenization and Sentence splitting tasks have the same values because we perform joint tokenization and sentence splitting, meaning we only have a single model for both tasks. The sizes vary quite a lot because we have a different character set for each language. For example, for Romanian we have 112 different characters while for Chinese we have over 3000. The need to store each character, encoded in UTF, to have an index-based vocabulary leads to large variations in the model sizes (35 MB standard deviation). For the UPOS task we observe near constant sizes because all languages use the same parts of speech, meaning the exact same number of parameters thus negligible variation. For the XPOSes and Attributes we observe a larger variation explained by the varying number of output labels leading to different neural network parameter counts. Please note that because for the UPOS, XPOS and Attributes we did not count the word embedding sizes, while for the Tokenization and Sentence splitting we included the vocabulary itself, their sizes are smaller. If we were to add the word embeddings (external resources) the model sizes for the neural networks would explode with models routinely over 1 GB.

Table 5 presents overall accuracies. For all tasks the neural network models obtain better accuracy figures than their lightweight counterparts. While the advantage of the neural models is minimal for the basic tokenization and sentence splitting tasks, the power of these models starts to show on the more complex tasks of part of speech tagging where the difference increases to over 2−3%.

6 Conclusions

We have successfully covered tokenization, sentence splitting and part-of-speech tagging on 44 languages from the Universal Dependencies Corpus. Because scalability is an important aspect for any application, we focused on providing an overview that covers both accuracy and model footprint.

As expected, the BLSTM classifiers outperformed the classical (lightweight) models in terms of accuracy and provided state-of-the art results for most languages, even surpassing the highest reported results in the UD Shared Task for some languages at the time of writing. However, training them proved to be a time-consuming task, not to mention the difficulties that arise in the hyper-parameter tuning process, mainly because of the high time-to-value. In fact, is is likely that the same network architecture we used would be able to produce better results if one would perform a grid-search hyper-parameter optimization. Other lessons learned about using the BLSTM classifier are outlined in 4.3.

An important conclusion is that there exists a sensible trade-off between accuracy and model footprint. For example, on average, the accuracies of decision trees (lightweight) and neural (heavyweight) models is minimal on the tokenization and sentence splitting task, while the footprint is $\tilde{2}$ orders of magnitude in favor of the lightweight algorithms. The average difference between linear and BLSTM classifiers on UPOS tagging is about 3% and lower than 4% for XPOS tagging. So, whenever a marginally lower accuracy is not an insurmountable disadvantage, one could resort to using a classical approach if memory and computational power are an issue.

We hope that the information provided in this paper will support researchers and engineers in selecting and designing an optimal system architecture depending on the available resources and computational requirements.

Our natural language processing framework is freely available for download and use and an on-line demo is hosted on our NLP Tools website[5].

[5] http://slp.racai.ro/index.php/mlpla-new/.

References

1. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information (2016). arXiv preprint arXiv:1607.04606
2. Boroş, T., Dumitrescu, S.D., Pipa, S.: Fast and accurate decision trees for natural language processing tasks. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, INCOMA Ltd., Varna, Bulgaria, pp. 103–110, September 2017. https://doi.org/10.26615/978-954-452-049-6_016
3. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: EMNLP, pp. 740–750 (2014)
4. Dozat, T., Manning, C.D.: Deep Biaffine attention for neural dependency parsing (2016). arXiv preprint arXiv:1611.01734
5. Dozat, T., Qi, P., Manning, C.D.: Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 20–30. Association for Computational Linguistics, Vancouver, Canada, August 2017. http://www.aclweb.org/anthology/K/K17/K17-3002.pdf
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Nivre, J., et al.: Universal Dependencies 2.0 (2017). http://hdl.handle.net/11234/1-1983, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague. http://hdl.handle.net/11234/1-1983
8. Petrov, S., Das, D., McDonald, R.: A universal part-of-speech tagset (2011). arXiv preprint arXiv:1104.2086
9. Quinlan, J.R.: Simplifying decision trees. Int. J. Man Mach. Stud. **27**(3), 221–234 (1987)
10. Tufiş, D., Dragomirescu, L.: Tiered tagging revisited. In: Proceedings of the 4th LREC Conference, pp. 39–42 (2004)
11. Zafiu, A., Dumitrescu, S.D., Boroş, T.: Modular language processing framework for lightweight applications (MLPLA). In: 7th Language & Technology Conference (2015)
12. Zeman, D., Ginter, F., Hajič, J., Nivre, J., Popel, M., Straka, M., et al.: CoNLL 2017 shared task: multilingual parsing from raw text to universal dependencies. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 1–20. Association for Computational Linguistics (2017)
13. Zeman, D., Popel, M., Nitisaroj, R., Li, J.: CoNLL 2017 shared task: multilingual parsing from raw text to universal dependencies. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 1–19. Association for Computational Linguistics, Vancouver, Canada, August 2017. http://www.aclweb.org/anthology/K/K17/K17-3001.pdf

Ontologies and Wordnets

A Semantic Similarity Measurement Tool for WordNet-Like Databases

Marek Kubis(✉)

Faculty of Mathematics and Computer Science,
Department of Computer Linguistics and Artificial Intelligence,
Adam Mickiewicz University, ul. Umultowska 87, 61-614 Poznań, Poland
mkubis@amu.edu.pl

Abstract. The paper describes a new framework for computing the semantic similarity of words and concepts using WordNet-like databases. The main advantage of the presented approach is the ability to implement similarity measures as concise expressions in the embedded query language. The preliminary results of the use of the framework to model the semantic similarity of Polish nouns are reported.

1 Introduction

Among various applications of WordNet [4], the task of modeling semantic similarity between words has attracted considerable attention over the last two decades. WordNet-based semantic similarity measures, ranging from simple path-length dependent functions [14,26] and measures that exploit the notion of the least common subsumer[1] [36] to those that utilize information content computed over corpora [10,16,27], have been proposed in the literature. These measures have been evaluated within the task of word sense disambiguation [21] and incorporated into natural language processing and information extraction systems [2,31]. Despite a wide range of applications, the issue of using other wordnets in place of Princeton WordNet as resources for modeling similarity among words appears not to have gained the same level of attention. Our aim is to use PolNet [35] and PlWordNet [17] to model the semantic similarity of Polish nouns. Since we have not found a software package for measuring semantic similarity that could be easily adapted to make use of both Polish wordnets (cf. Sect. 2), we decided to implement our own. Therefore, the goal of this paper is twofold. First, we present WSim: a new tool for determining degrees of semantic similarity using measures computed over WordNet-like databases.[2] Second, we report the preliminary results of the use of WordNet-based similarity measures to model the similarity of Polish nouns. This is, to the best of our knowledge,

[1] A joint transitive hypernym of two synsets such that no other joint transitive hypernym of these synsets is placed below it within the hypernymy hierarchy.

[2] Databases that are organized similarly to WordNet [4], called wordnets in the rest of the paper.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 155–168, 2018.
https://doi.org/10.1007/978-3-319-93782-3_12

the first attempt to apply two wordnets developed for the same language in a shared application-oriented task.

The paper is a revised version of [13]. It presents new, unpublished results of supervised models of semantic similarity built on the values of wordnet-based measures (cf. Sect. 6). Furthermore, it reproduces the experiments from the original paper using a Polish Wikipedia dump from November 20, 2017 instead of the February 6, 2014 dump used previously. Lastly, more detailed results on measuring semantic similarity between English counterparts of Polish nouns are provided (cf. Tables 5 and 9).

2 Related Work

WordNet::Similarity [23] is a widely-cited software package that implements a range of WordNet-based semantic similarity measures. This package has become a de facto standard tool for computing similarity scores using WordNet and serves as a reference point for other implementations (e.g., [24]). Unfortunately, WordNet::Similarity operates only on Princeton WordNet and is not able to load wordnets that do not conform to the internal storage format of the wn program distributed with Princeton WordNet [32]. The same restriction applies to the Python interface to WordNet provided by the NLTK toolkit [1]. In addition to Princeton WordNet, the Java reimplementation of WordNet::Similarity by Shima [29], called WS4J, can load the Japanese WordNet [9]. PolNet is not distributed in the Princeton WordNet conformant form and we have not found any tool that could be used to convert it to this format without a vast amount of preprocessing.

A major advance in terms of interoperability is the WordnetTools library [24], which can load any wordnet that is stored in a file conforming to the Wordnet-LMF format [30]. However, at the time of writing, neither PolNet nor PlWordNet had been released in this format. WordnetTools also accepts files in the Global WordNet Grid format [6], but we were unable to load into it the DEBVisDic [8] conformant XML file, which is part of the PlWordNet distribution.

Since exact replication of results using different software packages is not easy to achieve (see [24, sect. 5.3]), we did not want to use separate tools for computing values of similarity measures for the two wordnets (e.g. NLTK for PlWordNet and WordnetTools for PolNet). Therefore, we decided to reimplement WordNet-based semantic similarity measures on top of the WQuery suite [11,12], which is able to load both PlWordNet and PolNet. An additional advantage of this approach is the ability to modify the similarity measures by revising the concise expressions of the WQuery language (cf. Sect. 4) instead of the Java code of WordnetTools, which, in the case of any changes, would require recompilation. Furthermore, since WQuery (version 0.10) can load wordnets stored in Wordnet-LMF, DEBVisDic [8], and the Princeton WordNet internal format,[3] we acquired the ability to make direct comparisons between the values of similarity measures computed for the lexical databases stored in all of the aforementioned formats.

[3] Through the JWI library [5].

3 WSim

As mentioned in the previous section, WSim is built on the WQuery suite. Therefore, before computing the values of similarity measures, the wordnet must be converted into the WQuery database format using the `wcompile` command[4] from the WQuery toolkit. Since both PlWordNet and PolNet are available in the XML files compatible with the DEBVisDic editor [8], the `-t deb` option must be passed to the command

```
wcompile -t deb polnet.xml > polnet.wq
```

With the wordnet in the WQuery format, the similarity of pairs of words (or word senses) can be computed by passing them to the standard input of the `wsim` command, separated by tab characters.

```
wsim polnet.wq < pairs
```

By default, `wsim` determines the similarity of a pair of words by inverting the value of the shortest path length in the hypernymy hierarchy linking the synsets containing the given words; thus for the pair *samochód* (Eng. car) and *rower* (Eng. bicycle) the similarity determined with PolNet is

```
0.25
```

WSim implements six semantic similarity measures:

1. inverted length of the shortest path,
2. Wu-Palmer [36],
3. Resnik [27],
4. Jiang-Conrath [10],
5. Leacock-Chodorow [14],
6. Lin [16].

Following [23], we denote these measures by *path*, *wup*, *res*, *jcn*, *lch*, and *lin*, respectively. They can be selected by passing the `-m` option to the `wsim` command. For instance, to compute the Wu-Palmer measure, the command

```
wsim polnet.wq -m wup < pairs
```

must be executed. In the case of information content dependent measures [10, 16, 27] word (or sense) counts can be submitted in a file passed as an argument of the `-c` option, e.g.

```
wsim polnet.wq -m res -c counts < pairs
```

If the counts are distributed along with a wordnet (as is true in the case of Princeton WordNet) the `-c` option can be skipped.

```
wsim wordnet.wq -m res < pairs
```

[4] We assume in the following examples that all commands are invoked in the Linux shell environment.

4 Implementation of Measures

The similarity measures are implemented in WSim as functions formulated in the WQuery language [11]. Every function that ends with the _measure suffix is interpreted as a similarity measure and is available through the -m option of the wsim command. For every pair of senses read from the input, the wsim command determines their corresponding synsets and passes them to the function indicated by the argument of the -m option. In the case of pairs of words, wsim returns the maximum of the similarity values computed for every pair of senses of the submitted words.

Let us consider the Wu-Palmer measure as an example. The measure is given by the following formula (cf. [2,36]):

$$\frac{2 * dep(lcs(l,r))}{dist(l, lcs(l,r)) + dist(r, lcs(l,r)) + 2 * dep(lcs(l,r))}$$

where l and r are synsets, $lcs(l,r)$ denotes the least common subsumer of l and r, *dist* denotes the distance between two synsets in the hypernymy hierarchy, and *dep* returns the distance of a synset from the hypernymy root. The Wu-Palmer measure has the following implementation in WQuery:

```
function wup_measure do
  %l, %r := %A
  %lcs := lcs_by_depth(%l, %r)
  %dl := lcs_dist(%l, %lcs)
  %dr := lcs_dist(%r, %lcs)
  %dlcs := root_dist(%lcs)
  emit 2*%dlcs/(%dl + %dr + 2*%dlcs)
end
```

We will not be discussing WQuery in detail.[5] In order to follow the examples it is enough to understand that arithmetic expressions, variable assignments (:=), and function calls (f(...)) are interpreted in a manner similar to that of scripting languages such as Python. The arguments are passed to a function in the %A variable and return values are passed using the emit statement. The main advantage of using WQuery in place of a generic scripting language to implement similarity measures is the ability to use regular expressions over the semantic relation names to denote paths in the wordnet graph. In the case of wup_measure the sub-function lcs_dist that computes the distance from a synset to its least common subsumer determines the paths from a synset %s to its subsumer %lcs via the regular expression

```
%s.hypernym*.%lcs
```

that traverses zero or more times through the hypernym relation from the synset %s to its subsumer %lcs. The root_dist function that computes the distance

[5] Interested readers can consult [11].

from a synset to the hypernymy root uses the expression

```
%A.hypernym*[empty(hypernym)]
```

to denote the paths from a synset `%A` through zero or more hypernymy links to the synsets that do not have hypernyms.[6] We present the complete code implementing these functions below.

```
function lcs_dist do
  %s, %lcs := %A
  emit min_size(%s.hypernym*.%lcs) - 1
end

function root_dist do
  emit min_size(
    %A.hypernym*[empty(hypernym)]) + 1
end

function min_size do
  emit distinct(min(size(%A)))
end
```

The `lcs_by_depth` function, which is also called by `wup_measure`, is a built-in function of WQuery that determines the least common subsumers of synsets.

The similarity functions are loaded into WSim at the beginning of execution from a designated directory. Thus, given a correspondence between arguments of `wsim` and function names and the ability to address arbitrary paths in the wordnet graph using the WQuery language, the user can easily experiment with definitions of new measures. For instance, the user can consider a meronymy-based variant of the *path* measure by providing the following function to `wsim`:

```
function mpath_measure do
  %l, %r := %A
  %mpaths := %l.meronym*.^meronym*.%r
  emit 1/min_size(%mpaths)
end
```

5 Semantic Similarity Computation Using Polish Wordnets

Given a tool that accepts lexical databases stored in the DEBVisDic editor compatible format, we can compute the values of similarity measures for both Polish wordnets and compare them to the human similarity ratings. In the case of English, the Rubenstein and Goodenough dataset of 65 human-rated noun pairs [28] and its 30-pair subset from Miller and Charles [19] are often used

[6] The synsets satisfying the condition `empty(hypernym)`.

Table 1. Correlation coefficients between the human ratings and similarity measure scores determined for PL39 word pairs that occur in both Polish wordnets.

Measure	Pearson's		Spearman's	
	PlWN	PolNet	PlWN	PolNet
path	0.6051	0.6421	0.4658	0.6530
wup	0.6322	0.6835	0.6079	0.6902
lch	0.5981	0.6865	0.4658	0.6530
res	0.6026	0.6369	0.6389	0.6539
jcn	0.5358	0.4938	0.6148	0.6700
lin	**0.6584**	**0.7081**	**0.6520**	**0.7029**

for the purpose of evaluating similarity measures (e.g., [2,22,27]). Paliwoda-Pękosz and Lula [20], who translated this dataset into Polish and had it rated, also report the performance of several similarity measures on 39 pairs of the translated nouns covered by version 0.95 of PlWordNet. We refer hereafter to this dataset as PL39 and to the Rubenstein and Goodenough dataset as RG65. For the purpose of our analysis we use version 2.2 of PlWordNet [17] and version 3.0 of PolNet [35]. Furthermore, in order to determine the values of measures that utilize information content (i.e. Resnik, Jiang-Conrath, and Lin), we use word frequencies derived from Polish Wikipedia.

PlWordNet 2.2 and PolNet 3.0 cover 38 and 26 pairs of nouns from the PL39 dataset, respectively. The correlation coefficients between the values of the similarity measures and the human rating of 26 noun pairs common to both wordnets are given in Table 1. It can be seen that, regardless of the correlation type, the Lin measure performs best. The same measure achieves the best results in the case of all 38 word pairs covered by PlWordNet (cf. Table 2). We report the pairs of words from PL39 and the corresponding values of the Lin measure in Table 3.[7] For the purpose of comparison we also computed the correlation coefficients between the human ratings of the RG65 word pairs and similarity measure scores determined using version 3.0 of WordNet. The results obtained for 26 word pairs from RG65, which are English counterparts of PL39 word pairs common to both Polish wordnets, are given in columns 2 and 3 of Table 5. Columns 4 and 5 present the results for 38 pairs of words from RG65, which are counterparts of all PL39 word pairs that occur in PlWordNet. In the case of WordNet, the Leacock-Chodorow measure results in the highest Pearson's correlation and the Jiang-Conrath and path measures achieve the highest value of Spearman's correlation coefficient among the analyzed similarity functions.

Given the correlation coefficients for a fixed measure and the same corpus[8] it is tempting to compare the differences between the two wordnets with respect

[7] The pair *środek dnia/południe* is omitted in Table 3, since *środek dnia* occurs in neither PlWordNet 2.2 nor in PolNet 3.0.

[8] In the case of information content-based measures.

Table 2. Correlation coefficients between the human ratings and similarity measure scores determined for all PL39 word pairs that occur in PlWordNet.

Measure	Pearson's	Spearman's
path	0.5915	0.5537
wup	0.6896	0.6738
lch	0.6423	0.5537
res	0.6780	0.6866
jcn	0.4419	0.6544
lin	**0.7069**	**0.6941**

to the results on the same dataset. However, it must be noted that although the correlation coefficients between human ratings for the 26 nouns from PL39 and measure values induced from PolNet are generally higher[9] than the corresponding coefficients derived for PlWordNet, the results are difficult to interpret due to size of the dataset size and are not significant at the $\alpha = 0.05$ level according to the Meng, Rosenthal, and Rubin's z-test as implemented by Diedenhofen [3].

6 Supervised Similarity Models

Given the values of semantic similarity measures computed for PL39 word pairs, we decided to determine whether supervised models of similarity can be built on the measured values. We developed a range of regression models using the wordnet-based similarity measures as explanatory variables and the similarity score from PL39 as the response variable. The methods of regression we considered are: linear regression (lr), neural networks (nn), regression tress (rt), random forests (rf), and ϵ-support vector regression (svr). We used R environment [25] with `stats`, `nnet` [34], `rpart` [33], `randomForest` [15] and `e1071` [18] packages to develop and evaluate the regression models. For the neural network architecture, we chose a multilayer perceptron with one hidden layer and performed a grid search with 5-fold cross-validation on the training set to determine the number of neurons in the hidden layer. In the case of random forests, we performed a grid search with 5-fold cross-validation to determine the number of trees and the minimum size of the terminal nodes. For the support vector regression we examined linear, polynomial,[10] radial basis, and sigmoid kernels and performed a grid search for values of the C, γ, and ϵ parameters (Table 4).

The models were evaluated using the leave-one-out cross-validation technique (i.e. the similarity of a given pair of words is predicted using the model trained on the similarity scores measured for the other pairs). Table 6 presents the correlation coefficients between the human ratings of the PL39 dataset word pairs and

[9] With the exception of Pearson's correlation coefficient for the Jiang-Conrath measure.

[10] Polynomial kernels of degrees 2 and 3 were considered.

Table 3. The values of similarity measures determined for the PL39 word pairs.

Words		lin		rf	
		PlWN	PolNet	PlWN	PolNet
południe	sznurek	0.0000	0.0000	1.0624	1.1758
owoc	piec	0.2268	0.3524	1.4508	1.2489
autograf	wybrzeże	0.0000		0.7924	
auto	czarodziej	0.0000		0.6829	
kopiec	kuchenka	0.1763	0.3671	0.3007	1.4083
azyl	owoc	0.0000	0.0000	1.2264	1.1191
azyl	zakonnik	0.0000	0.0000	0.7361	0.6371
chłopiec	kogut	0.4860	0.6430	1.5495	1.9392
poduszka	klejnot	0.2825		1.8769	
zakonnik	niewolnik	0.7039	0.3311	1.9370	1.0559
azyl	cmentarz	0.4954	0.2848	1.3131	1.5667
wybrzeże	las	0.6917	0.7449	2.1272	2.3904
chłopiec	mędrzec	0.5167		1.0175	
auto	poduszka	0.3290		1.6236	
kopiec	wybrzeże	0.0000	0.0000	0.9607	0.6992
chłopak	czarodziej	0.2250		1.4009	
las	cmentarz	0.0000	0.2892	0.8224	1.1178
jedzenie	kogut	0.7364	0.3480	2.1898	0.5905
wybrzeże	pagórek	0.6107	0.6395	1.6843	1.7047
piec	narzędzie	0.4342	0.5911	1.1178	1.7462
żuraw	kogut	0.6008		2.2085	
cmentarz	kopiec	0.0000	0.0000	0.7922	0.3618
szkło	klejnot	0.3005	0.6592	0.5466	1.1403
żuraw	przyrząd	0.3890		1.5338	
brat	chłopak	0.8419	0.6843	2.9803	1.7356
mędrzec	czarodziej	0.5862		2.2495	
ptak	żuraw	0.7493		2.9402	
ptak	kogut	0.7520	0.7521	3.1408	1.4753
jedzenie	owoc	0.2761	0.8863	0.8367	3.2757
brat	zakonnik	1.0000	1.0000	3.4973	3.2666
piec	kuchenka	0.5393	0.3850	1.8863	0.5378
pagórek	kopiec	1.0000	1.0000	3.5480	3.4066
przewód	sznurek	0.0000	0.5194	0.9194	1.3328
szkło	szklaneczka	0.6077		2.1330	
autograf	podpis	0.9018		3.1604	
narzędzie	przyrząd	0.9794	1.0000	2.6707	2.7127
chłopiec	chłopak	1.0000	1.0000	3.5633	3.1999
auto	samochód	1.0000	0.8629	3.0764	2.9260

Table 4. Correlation coefficients between PlWordNet- and PolNet-based measures.

Measure	Pearson's	Spearman's
path	**0.8503**	0.7344
wup	0.8450	**0.8544**
lch	0.8369	0.7344
res	0.7682	0.7258
jcn	0.7045	0.7895
lin	0.8015	0.7902

Table 5. Correlation coefficients between the human ratings and similarity measure scores determined for the RG65 word pairs which are counterparts of the PL39 pairs.

Measure	26 pairs		38 pairs	
	Pearson's	Spearman's	Pearson's	Spearman's
path	0.7274	0.6351	0.7300	**0.6911**
wup	0.6795	0.5785	0.7260	0.6749
lch	**0.7373**	0.6243	**0.7678**	0.6826
res	0.6598	0.5903	0.7033	0.6521
jcn	0.4310	**0.6610**	0.3642	0.4315
lin	0.6773	0.5837	0.5652	0.4041

Table 6. Correlation coefficients between the human ratings and supervised model scores determined for PL39 word pairs that occur in both Polish wordnets.

Measure	Pearson's		Spearman's	
	PlWN	PolNet	PlWN	PolNet
lin	0.6584	**0.7081**	**0.6520**	**0.7029**
lr	0.5863	0.5779	0.5348	0.4835
nn	0.4833	0.5719	0.3484	0.5232
rt	0.5839	0.6643	0.3546	0.1228
rf	**0.6634**	0.5535	0.5413	0.3919
svr	0.6411	0.6112	0.5423	0.5574

the similarity scores determined by the supervised models built on the values of wordnet-based similarity measures. For this experiment, we restricted the dataset to 26 word pairs from PL39 that occur in both Polish wordnets. Table 7 reports the results obtained for models built on 38 noun pairs from PL39 that occur in PlWordNet. It can be seen that in both settings the Lin measure outperforms the supervised models, with the sole exception of the random forest model built from the similarity measures determined using PlWordNet for the dataset restricted to 26 common word pairs. Furthermore, even in this exceptional case, the difference between the correlation coefficients determined for the Lin measure and random forest model is not significant at the $\alpha = 0.05$ level according to the Meng, Rosenthal, and Rubin's z-test. Similar results can be observed for the word pairs from RG65 which are English counterparts of the pairs of nouns from PL39 (Table 8). The Leacock-Chodorow measure outperforms the supervised models with respect to Pearson's correlation, whereas the Jiang-Conrath and path measures outperform the supervised models with respect to Spearman's rank correlation. This suggests that, in the case of a small dataset, it is worth choosing one of the wordnet-based similarity measures instead of trying to build a supervised regression model on top of them.

Table 7. Correlation coefficients between the human ratings and supervised model scores determined for all PL39 word pairs that occur in PlWordNet.

Measure	Pearson's	Spearman's
lin	**0.7069**	**0.6941**
lr	0.6497	0.5909
nn	0.5373	0.4269
rt	0.5889	0.4398
rf	0.7012	0.6444
svr	0.6826	0.6337

Table 8. Correlation coefficients between the human ratings and supervised model scores determined for the RG65 word pairs which are counterparts of the PL39 pairs.

Measure	26 pairs		38 pairs	
	Pearson's	Spearman's	Pearson's	Spearman's
path	0.7274	0.6351	0.7300	**0.6911**
lch	**0.7373**	0.6243	**0.7678**	0.6826
jcn	0.4310	**0.6610**	0.3642	0.4315
lr	0.7208	0.5036	0.7367	0.5762
nn	0.4746	0.4044	0.3843	0.3374
rt	0.6971	0.2144	0.6734	0.2831
rf	0.7013	0.4366	0.6993	0.4884
svr	0.7199	0.5781	0.7431	0.6509

Table 9. The values of similarity measures computed for the RG65 word pairs which are counterparts of the PL39 pairs.

Words		path	lch	jcn
Noon	String	0.08333	1.2040	6.527e–02
Fruit	Furnace	0.11111	1.4917	6.094e–02
Autograph	Shore	0.10000	1.3863	0.000e+00
Automobile	Wizard	0.07692	1.1239	7.383e–02
Mound	Stove	0.14286	1.7430	6.815e–02
Asylum	Fruit	0.14286	1.7430	6.531e–02
Asylum	Monk	0.09091	1.2910	5.530e–02
Boy	Rooster	0.08333	1.2040	7.266e–02
Cushion	Jewel	0.14286	1.7430	6.944e–02
Monk	Slave	0.20000	2.0794	6.614e–02
Asylum	Cemetery	0.08333	1.2040	5.510e–02
Coast	Forest	0.16667	1.8971	6.276e–02
Boy	Sage	0.16667	1.8971	6.802e–02
Automobile	Cushion	0.16667	1.5404	8.940e–02
Mound	Shore	0.20000	2.0794	1.672e–01
Lad	Wizard	0.20000	2.0794	7.588e–02
Forest	Graveyard	0.11111	1.4917	5.871e–02
Food	Rooster	0.06250	0.9163	6.711e–02
Coast	Hill	0.20000	2.0794	2.187e–01
Furnace	Implement	0.12500	1.6094	7.640e–02
Crane	Rooster	0.12500	1.6094	0.000e+00
Cemetery	Mound	0.09091	1.2910	5.825e–02
Glass	Jewel	0.16667	1.7430	7.163e–02
Crane	Implement	0.20000	2.0794	7.840e–02
Brother	Lad	0.20000	2.0794	8.296e–02
Sage	Wizard	0.16667	1.8971	5.800e–02
Bird	Crane	0.25000	2.3026	0.000e+00
Bird	Cock	0.50000	2.9957	2.681e–01
Food	Fruit	0.10000	1.3863	8.607e–02
Brother	Monk	0.50000	2.9957	6.894e–02
Furnace	Stove	0.10000	1.3863	5.969e–02
Hill	Mound	1.00000	3.6889	1.288e+07
Cord	String	0.50000	2.9957	6.553e–01
Glass	Tumbler	0.50000	2.9957	3.789e–01
Autograph	Signature	0.50000	2.9957	0.000e+00
Implement	Tool	0.50000	2.9957	8.484e–01
Boy	Lad	0.50000	2.9957	2.929e–01
Automobile	Car	1.00000	3.6889	1.288e+07
Midday	Noon	1.00000	3.6889	1.288e+07

7 Conclusion

We presented a new framework for semantic similarity computation, using wordnet-based measures. The main advantages of our tool are compatibility with various wordnet database formats and the ability to implement new measures using embedded query language. The framework was employed to model the semantic similarity of nouns using measures derived from two Polish wordnets, PlWordNet and PolNet. The results must be considered preliminary due to the small size of the dataset used for the purpose of evaluation. Nevertheless, this is the first attempt to use both Polish wordnets within the context of a shared task.

In the future, we plan to extend the framework with additional measures (e.g., [7]). We also intend to create a larger evaluation set that will cover the content of PolNet more extensively.

References

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly Media, Sebastopol (2009)
2. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Comput. Linguist. **32**(1), 13–47 (2006)
3. Diedenhofen, B.: cocor: Comparing correlations, (Version 1.0-0) (2013). http://r.birkdiedenhofen.de/pckg/cocor/
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)
5. Finlayson, M.A.: Java libraries for accessing the princeton wordnet: comparison and evaluation. In: Proceedings of the 7th Global Wordnet Conference, Tartu, Estonia, pp. 78–85 (2014)
6. Global WordNet Association: Global WordNet Grid (2012). http://globalwordnet.org/global-wordnet-grid/. Accessed 20 Sept 2015
7. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms, chap. 13, pp. 305–332. In: Fellbaum [4] (1998)
8. Horak, A., Pala, K., Rambousek, A., Povolny, M.: DEBVisDic - first version of new client-server wordnet browsing and editing tool. In: Sojka, P., et al. (eds.) Proceedings of the Third International WordNet Conference - GWC 2006. Masaryk University, Brno, Czech Republic (2005)
9. Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., Kanzaki, K.: Development of the Japanese WordNet. In: Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco, 26 May–1 June 2008, European Language Resources Association (2008)
10. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of 10th International Conference on Research in Computational Linguistics, ROCLING 1997 (1997)
11. Kubis, M.: A query language for wordnet-like lexical databases. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ACIIDS 2012. LNCS (LNAI), vol. 7198, pp. 436–445. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28493-9_46

12. Kubis, M.: A tool for transforming wordnet-like databases. In: Vetulani, Z., Mariani, J. (eds.) LTC 2011. LNCS (LNAI), vol. 8387, pp. 343–355. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08958-4_28
13. Kubis, M.: A semantic similarity measurement tool for WordNet-like databases. In: Vetulani, Z., Mariani, J. (eds.) Proceedings of the 7th Language and Technology Conference, pp. 150–154. Fundacja Uniwersytetu im. Adama Mickiewicza, Poznań, Poland, November 2015
14. Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification, chap. 11, pp. 265–283. In: Fellbaum [4] (1998)
15. Liaw, A., Wiener, M.: Classification and Regression by randomForest. R News **2**(3), 18–22 (2002). http://CRAN.R-project.org/doc/Rnews/
16. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning, ICML 1998, pp. 296–304. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)
17. Maziarz, M., Piasecki, M., Szpakowicz, S.: Approaching plWordNet 2.0. In: Proceedings of the 6th Global Wordnet Conference. Matsue, Japan, January 2012
18. Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F.: e1071: Misc functions of the department of statistics (e1071), TU Wien, R package version 1.6-3 (2014). http://CRAN.R-project.org/package=e1071
19. Miller, G.A., Charles, W.G.: Contextual correlates of semantic similarity. Lang. Cognit. Process. **6**(1), 1–28 (1991)
20. Paliwoda-Pękosz, G., Lula, P.: Measures of semantic relatedness based on wordnet. In: International Workshop For Ph.D. Students. Brno, Czech Republic (2009). ISBN: 978-80-214-3980-1
21. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 241–257. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36456-0_24
22. Pedersen, T.: Information content measures of semantic similarity perform better without sense-tagged text. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT 2010, pp. 329–332. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
23. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity: Measuring the Relatedness of Concepts. In: Demonstration Papers at HLT-NAACL 2004, pp. 38–41, HLT-NAACL-Demonstrations 2004, Association for Computational Linguistics, Stroudsburg, PA, USA (2004). http://dl.acm.org/citation.cfm?id=1614025.1614037
24. Postma, M., Vossen, P.: What implementation and translation teach us: the case of semantic similarity measures in wordnets. In: Orav, H., Fellbaum, C., Vossen, P. (eds.) Proceedings of the Seventh Global Wordnet Conference, Tartu, Estonia, pp. 133–141 (2014)
25. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2014). http://www.R-project.org/
26. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Trans. Syst. Man Cybern. **19**(1), 17–30 (1989)
27. Resnik, P.: using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI 1995, pp. 448–453. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)

28. Rubenstein, H., Goodenough, J.B.: Contextual correlates of synonymy. Commun. ACM **8**(10), 627–633 (1965)
29. Shima, H.: ws4j - WordNet Similarity for Java (2015). https://code.google.com/p/ws4j/. Accessed 28 Aug 2015
30. Soria, C., Monachini, M., Vossen, P.: Wordnet-LMF: Fleshing out a standardized format for wordnet interoperability. In: Proceeding of the 2009 international workshop on Intercultural collaboration, pp. 139–146. ACM, New York, USA (2009)
31. Stevenson, M., Greenwood, M.A.: A semantic approach to IE pattern induction. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL 2005, pp. 379–386. Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
32. Tengi, R.I.: Design and Implementation of the WordNet Lexical Database and Searching Software, chap. 4, pp. 105–127. In: Fellbaum [4] (1998)
33. Therneau, T., Atkinson, B., Ripley, B.: rpart: recursive partitioning and regression trees, R package version 4.1-8 (2014). http://CRAN.R-project.org/package=rpart
34. Venables, W.N., Ripley, B.D.: Modern Applied Statistics with S. Springer, New York (2002). https://doi.org/10.1007/978-0-387-21706-2. http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0
35. Vetulani, Z., Kubis, M., Obrębski, T.: PolNet - Polish WordNet: Data and Tools. In: Calzolari, N., et al. (eds.) Proceedings of the Seventh International Conference on Language Resources and Evaluation, ELRA, Valletta, Malta, May 2010
36. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL 1994, pp. 133–138. Association for Computational Linguistics, Stroudsburg, PA, USA (1994). https://doi.org/10.3115/981732.981751

Similarity Measure for Polish Short Texts Based on Wordnet-Enhanced Bag-of-words Representation

Maciej Piasecki(✉) and Anna Gut

Faculty of Computer Science and Management,
Wrocław University of Technology, Wrocław, Poland
maciej.piasecki@pwr.edu.pl
http://nlp.pwr.edu.pl

Abstract. We present a method for computing semantic similarity of Polish texts with main focus given to short texts. We have taken into account the limited set of language tools for Polish, and especially that syntactic and semantic parsers do not express accuracy and robustness high enough and to become a stable basis for similarity computation. A very large wordnet of Polish, namely plWordNet is used to construct meaning representations for words in such a way that different words of the similar meaning receive similar representations. The use of a Word Sense Disambiguation (WSD) tool for Polish brought positive results in one of the method variants, regardless of the limited accuracy of the WSD tool. The proposed measures have been compared with the manual evaluation of sentence pairs. The measures were also applied as a part of the Question Answering system. Improved performance of answer finding was achieved in several types of tests.

1 Introduction

Computing text-to-text similarity is a key issue for many applications of the Natural Language Engineering. It is getting more difficult, if the compared texts are short, or at least one of them, e.g. in Question Answering, comparison of sentences in automated Summarisation or building rankings of text snippets in Information Retrieval.

A good text similarity measure should go beyond string comparison, and should be based on semantic content of texts. The problem is naturally separated into two similarity levels of *words* and linguistic *text structures*. The depth of the analysis of text structures (syntactic, semantic or even pragmatic) is determined by the available language tools.

Methods for calculating semantic similarity of short texts can be divided [1] into three main types:

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 169–182, 2018.
https://doi.org/10.1007/978-3-319-93782-3_13

1. methods based on *overlaping of words*,
2. *TF.IDF-based* methods, i.e. based Term Frequency – Inverse Document Frequency [24],
3. and *linguistic measures.*

Methods of the first two groups represent a document as a bag of words (collection) and neglect linguistic structures. Typically for the first group, the similarity measure is calculated on the basis of the extent to which words of the two compared texts overlap, e.g. with the help of Jacquard or Dice measure. The comparison is done mostly on the level of text words but filtered by a stop word list or limited to selected grammatical classes. The Polish language, which we focus on, expresses rich inflection and weakly constrained word order, thus similar information can be conveyed in much larger diversified of ways than in, e.g., English. The variety of word forms can be reduced by mapping them onto their *lemmas*[1]. This requires preprocessing on the morpho-syntactic level by a morpho-syntactic tagger.

TF.IDF family of measures refer to the well known and mostly effective *vector model* for Information Retrieval that achieves surprisingly good results in many applications.

Linguistic measures are a heterogeneous group and explore information provided by the available language resources and tools. Several approaches in this group are based only on word similarity, e.g. [4], [8], [3], taking text words as an input. Corley and Mihalcea [4] proposed a mixed measure based on calculating similarity of words between two texts with more weight given to more specific words. Any word-to-word similarity can be used. This approach was tested on word similarity measures based on Princeton WordNet [5]. IDF was used as a factor representing word specificity. This method has been further extended in [16] with a larger number of word similarity measures analysed, including measures based on text corpora, not only on WordNet.

Li et al. [8] considered not only the similarity of words but also the order of their occurrence. Liu and Wang [9] expanded wordnet-based word similarity to computing the similarity of sentences. Word-to-word similarity measure proposed in [19] was used. The semantic similarity of sentences was computed in four steps. First text words are mapped onto concepts in ontology. The exact way of solving the ambiguous cases was not explained. The identified nodes are expanded, i.e. direct descending nodes are added and all ancestors along one street path. The nodes are weighted according to the path distance from the directly mapped node. Finally the constructed vectors are compared with the cosine measure.

Corley and Mihalcea [4] proposed merging together several measures. Bär et al. [3] tested a large number of measures and selected those producing the best results. In addition to the text similarity measures, they calculated also the longest common substring and the number of n-grams based on characters and also words.

[1] A lemma is a basic morphological form (or entry form) which represents a set of word forms that differ only in the values of their grammatical categories, like case, gender, person.

Tree Edit Distance has been also applied in measuring semantic similarity of texts, e.g. [7]. Punyakanok et al. [21] achieved higher accuracy using graphs produced by the dependency parser.

Methods based on the comparison of syntactic structures can be applied to Polish to a limited extent. Dependency parsers for Polish express quite substantial error rate that can result in a noise. However, their quality is continuously improving, e.g. [20]. Parsers of other types do not provide disambiguation of structures or have limited coverage. WordNet is associated with a corpus including manually disambiguated word senses that allows for collecting the word sense frequencies. Such data that are a basis for many word similarity measures are not accessible for Polish. Thus, we concentrated on the optimal use of the available language resources, e.g. plWordNet, – a very large wordnet of Polish, and selected language tools, especially robust morpho-syntactic taggers.

Our goal was to develop a method for computing similarity of short texts in Polish aimed at capturing the similarity of the information conveyed by texts regardless of particular words used. We wanted to base the description of the lexical meanings on plWordNet and to apply a set of possibly simple language tools without the need of referring to some form of parsing.

2 Wordnet-Based Text Similarity Measure

The same message can be often expressed using different words. In longer texts different synonyms occur interchangeably across a single document, while in short texts one synonym of all possible is used only. When two bag-of-word representations of synonymous short texts are compared, the mismatch is very likely, if the comparison is done on the level of words or lemmas. In order to check if two short texts are about the same topic, we need to abstract from the exact words used in them.

Facing the lack of robust parsers for Polish, we selected bag of words representation instead of a structure-based one. In UKP system [3] a thesaurus built by the means of crowdsourcing was utilised. In the case of Polish, plWordNet – a very large Polish wordnet [12] can be used instead. Texts to be compared are morpho-syntactically tagged and lemmatised. A lemma can correspond to several lexical meanings represented by plWordNet synsets (sets of near synonyms) and described by lexico-semantic relations linking synsets into a complex network[2].

In order to reduce the variety of ways for expressing identical lexical meanings text words can be mapped onto the appropriate synsets – synonymous word uses are mapped to the same synset. However, two problems appear. Such mapping

[2] plWordNet includes also relations linking *lexical units* (i.e. triples: lemma, Part of Speech and sense identifier), and selected of them were used also in the proposed methods. plWordNet 3.0 emo is the largest wordnet in the world including 197,721 synsets, 179,125 lemmas and 260,214 lexical units described by more 40 types of lexico-semantic relations (more than 90 together with subtypes) and more than 600,000 relation links, cf. [13]. The vast majority of plWordNet has been manually mapped onto Princeton WordNet.

requires the use of a Word Sense Disambiguation (henceforth WSD) tool that still expresses a significant error (around 30%). Moreover, not all words in text are used in their literal meanings and the same words can be used in utterances describing or referring to different subtopics. The same subtopic can be discussed in different texts by using slightly different words. However, we assume that all words used for discussing the same subtopic are closely semantically related. Thus synsets corresponding to them are located in the same semantic field. As a wordnet is a large lexical semantic network, semantic fields can be represented by subgraphs consisting of links of lexico-semantic relations. In order to cope with both problems, we decided to represent the meaning of every word in text by a set of synsets consisting of:

- the corresponding synset,
- and synsets linked to it by paths in the wordnet graph of the limited length.

In practice, we limited the paths to single links in order to avoid introduction of too much semantic noise to the representation.

Due to the different character of semantic relations, the synsets they link express varied information about each other. The paths used to identify the set leads to synsets that are related to a different extent to the central synset. This is modelled by weights assigned to synsets: 1 for the corresponding synset and < 1 for synsets linked to it by relations. So, finally, a text word occurrence is mapped on a collection of synsets assigned weights from $(0, 1]$.

In addition to relations linking synsets, selected relations directly linking lexical units, e.g. antonymy (linking to a lexical unit of the opposite meaning), have been also utilised in constructing collections, i.e. for a synset s corresponding to a text word w and including k lexical units $l_1, \ldots, l_k$, all synsets including lexical units that are linked by lexical relations to one of $l_1, \ldots, l_k$ are also added to the collection of w, i.e. such a collection includes all synsets that are targets for relations outgoing from the lexical units of the central synset.

We used the WSD tool called *WoSeDon*[3] [6] to assign plWordNet synsets to words in text. WoSeDon is based on the spreading activation model. Text words are first mapped onto plWordNet relation graph expanded with additional lexical and knowledge sources. Next, a variant of a spreading activation algorithm is used to identify synsets that are most related to the meaning of the text or a text fragment. The accuracy of WoSeDon is around 52% in tests on a balanced set of word senses[4] and around 68% for an average text sample, i.e. there is a big chance that the synset selected for a given word is wrong. However, the coverage of WoSeDon is very extensive as it can process all words described by plWordNet. That is why we have considered and evaluated also models in which the $p = 30\%$ or $p = 100\%$ top scored synsets (i.e. without WSD) were used to build a collection for a word. Scores produced by WoSeDon, that depend on

[3] Web application and Web Service: http://ws.clarin-pl.eu/wsd.shtml.

[4] The balanced test set is a more difficult test case as even very unfrequent senses are represented in it, while the average text sample has a character of running text and do not include many rare senses.

the applied exact version of the spreading activation algorithm, are in the range [0, 1], but do not sum up to 1 for synsets of a given word[5]. If the p percent of top scored synsets are selected, each gets a weight equal to the normalised scores produced by the WSD tool, i.e. the scores are normalised by the total sum of scores for the collection for the given word. In this model, a text word is mapped on the sum of collections built for the top scored synsets according to the WSD tool. If there is no synset for a text word in plWordNet, e.g. in the case of Proper Names, then such a word is mapped onto a singleton collection including its lemma[6] with the weight 1.

The final weight for a member of the collection C is calculated as following:

$$w_c = w_s * w_r * (1/|r \in C|) \tag{1}$$

where

- w_s, a synset weight, which is calculated on the basis of the frequency of s in the given collection or the scores from WSD, depending on the type of the collection, see below,
- w_r depends on the relation due to which s was added to C,
- the last factor reduces weights for more frequent relations.

Due to the limited accuracy of the WSD tool, we decided also to test a simpler model in which a word w – precisely its lemma – is mapped to a collection built from lemmas directly linked to w by lexico-semantic relations in plWordNet regardless their senses. Synonymy, as it is expressed by the membership in synsets, was treated in this model as just one more lexico-semantic relation. In this model no WSD is applied. All synsets that w belongs to, marked as $S(w)$, are used to build one merged collection and the collection consists of lemmas, not synsets. The collection for w is built from all synset members of $s \in S(w)$, as well as synset members linked to any $s \in S(w)$ by one of the selected lexico-semantic relations in the same way, as discussed above for the synset-based collections. The way of calculating weights is identical to the one applied to synset collections.

According to relation types used, we defined three collection types:

CollHHM – only *synonymy*[7], *hypernymy*, *hyponymy*, *meronymy* and *holonymy* are used for building collections,

[5] The total score in the whole graph is equal to 1.

[6] For Proper Names, the morpho-syntactic tagger used returns often the word form as a lemma.

[7] *Synonymy* is expressed by synsets. *Hypernymy* links more general synsets with more specific ones. However, in fact, in plWordNet all relations are defined for lexical units, cf. [14]. Synset relations are notational abbreviations and a link between two synsets means that all pairs of their members are linked by the given relation. *Hyponymy* (a reverse relation to the hypernymy). *Meronymy* links a part/element/portion etc. to the whole, e.g. [18], and *holonymy* links a whole to its part/element/portion, but it is not necessarily reverse to meronymy.

CollV1 – all relations from **CollHHM** plus additional relations used for the automated wordnet expansion [17], because lexical units linked by them express similar distributions in texts:

- synset relations: *type*/*instance* and *inter-register synonymy*[8],
- lexical unit relations: *femininity*, *markedness* (with subtypes: diminutive, augmentative, young being), *antonymy* and *converse*[9];

CollV2 is like **CollV1** but with antonymy and converse excluded.

In the wordnet expansion algorithm, lemmas associated by antonymy and converse to a new lemma to be added seemed to provide information that allowed for better identification of the appropriate locations in the wordnet graph for this new lemma and its possible senses. As 'description by antonymy' may be a little vague or non-direct, in the model **CollV2** we wanted to check whether removing both opposition relations can bring improvement in similarity calculation, e.g. the presence of antonyms in the collection could result in accidental associations between texts.

Concerning the values for relation weights, we also followed the solution developed for automated wordnet expansion in which weights for different relation types described how much semantic information can be "transferred" across the given relation link, i.e. how much information from one lexical unit can be ascribed to the other one at the link end, see details in [17]. The exact weight values for the wordnet expansion task were defined heuristically and verified experimentally as follows: synonymy 1.0, hypernymy 0.49, hyponymy 0.7, meronymy/holonymy 0.42, type 0.49, instance 0.7, femininity 0.7, inter-register synonymy 0.7, markedness (diminutive, augmentative, young being) 0.7, antonymy 0.28, converse 0.28.

Weights of the collection members discussed so far are aimed at expressing semantic information concerning the word represented by the given collection. However, such weights tell a little about how good is the given element in discriminating different texts. Discriminability should be calculated on the basis of a representative collection of documents. In the case of synset collection this is not possible, as there are no WSD corpora for Polish. Thus, in the case of both types of collections: the element specificity is estimated by the IDF factor [24]

[8] *Instance* links Proper Names with common nouns that represent their most specific characterisation and *type* is the reverse relation. *Inter-register synonymy* links synsets whose members are very close in meaning but differ in use in the language practice, i.e. they differ by their stylistic register, e.g. one synset belongs to the general language and the other one to vulgar.

[9] These relations are not shared among lexical units, and that is why they link only selected lexical units, not synsets. *Femininity* links lexical units of feminine forms with their masculine counterparts. *Markedness* is a general relation for several semantic associations signalled by the derivational relations, i.e. diminutive, augmentative, young being. *Antonymy* expresses binary opposition in meaning, and *converse* is also binary semantic opposition, but such that the two lexical units have (verbs) argument structures with arguments of opposite roles or filling the opposite roles in some predicates (nouns).

calculated for the lemma of this element on a basis of a text corpus (corpora used in experiments are described in the next section). The weights are multiplied by IDF before calculating similarity of vectors. For computing similarity of vectors we analysed several measures. The best results were achieved with: cosine, Jaccard and Dice measures.

3 Evaluation

There is no golden standard for similarity of Polish short texts. Instead, we applied two kinds of evaluation:

1. *comparison with the human judgements* about the similarity of sentences,
2. and *evaluation by application* in a QA system.

In all tests, texts were preprocessed by segmentation into sentences and tokens by MACA [23] and next analysed morpho-syntactically with the help of WCRFT tagger (performing lemmatisation, disambiguation) [22].

3.1 Semantic Similarity of Sentences

First, we wanted to compare the similarity measure values generated for sentence pairs with human similarity judgements. We followed the method proposed in [2] for gathering test data and the way in which their experiments were performed. In our case, first we selected 50 sentence pairs (i.e. 100 in total) from articles published in the Polish edition of Wikinews[10]. The sentence pairs have been chosen randomly in such way that we tried to find sentences coming from different articles and expressing some semantic similarity. Next, we asked volunteer annotators to assess sentences pairs that were generated randomly on the basis of the selected ones. The evaluators were asked to judge their potential semantic similarity. The random generation process was controlled in order to achieve comparable number of human judgements per a sentence pair. In a similar way to [2], we asked the to annotators to assign a presented sentence pair to one of the six categories:

0 – the sentences are about different topics,
1 – are not equivalent but on the same topic,
2 – not equivalent, but share some details,
3 – approximately equivalent but some important informations are different,
4 – equivalent, but differ in some minor details,
5 – equivalent, have the same meaning.

As a result, we have obtained 609 answers, i.e. individual assessments of sentence pairs, in total, and 12 scores per a single sentence pair on average. The range of scores per sentence was quite large in many cases and the inter-annotator agreement had to be low. However, we could not measure it, as the

[10] https://www.wikinews.org .

evaluation results were collected anonymously. Thus, these data provide only some insights and we have cleaned them by calculating average, median and removing cases that were identified as statistical anomalies.

In order to calculate IDFs we used a corpus of: 91,446 documents, where about 70% of them come from Wikipedia, the rest was selected from Wikinews.

Comparison of the proposed similarity measures with human evaluation was performed with two evaluation metrics: *Mean Square Error* and the *Significant Error Rate* – defined below.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{X}_i)^2 \tag{2}$$

where X_i is the value of the semantic similarity produced by the tested measure, and $\hat{X}_i$ is the average or median from the scores assigned by evaluators.

Significant Error Rate is calculated as the number of cases in which the tested measure expressed the error value beyond the lack of agreement between human evaluators, i.e. in our case the difference is greater then the standard deviation:

$$SER = size(\{X_i : i \in [1, n] \& |X_i - \hat{X}_i| < \sigma_i\}) \tag{3}$$

As a baseline for the similarity we used cosine measure for lemma-based TF.IDF representation of test sentences. The MSE of the baseline was 0.0926 in relation to the average and 0.0994 for median. SER was 25 and 24, respectively. So, while MSE is relatively small, the baseline was beyond the evaluator disagreement in half of the cases.

Table 1. Comparison of the similarity measures with the average of manual evaluation.

Key	Coll.	Weight	WSD	Mean Square Error			Significant Error Rate		
				Cos	Dice	Jac	Cos	Dice	Jac
Lemmas	CollV2	n	–	0.0916	0.0987	0.1464	27	25	21
Lemmas	CollHHM	y	–	0.0813	0.0840	0.1446	29	29	15
Synsets	CollV1	y	Best	0.0790	0.0814	0.1400	30	28	16
Synsets	CollV2	y	All	0.0868	0.0902	0.1406	27	27	16
Synsets	CollHHM	y	Best	0.0886	0.0918	0.1534	26	26	17
Synsets	CollHHM	y	30%	0.0893	0.0926	0.1511	25	25	17

Selected best results from the evaluation are presented in Table 1. They were compared with the average of the human scores. Similar results were obtained in comparison with median, but MSE was slightly higher and the number of test cases overcoming the baseline with respect to SER was lower. In Table 1, we can notice that all types of collections achieved results better than the baseline. In all cases weighting of relations improved the performance. Collections based on larger numbers of relations, namely CollV1 and CollV2 are better than CollHHM producing shorter vectors. Collections based on WSD express lower error in the case of taking only top synset, but collections based on lemmas mostly performed better than those based on synsets.

However, as the number of test sentence pairs was limited and the agreement between evaluators low, we should not go too far with conclusions. Generally, the proposed measures showed their potential beyond a typical TF.IDF-based cosine measure, that is often used. In order to check the influence of the proposed measures in a more reliable way, we will analyse its influence on the large scale Question Answering system for Polish in the next section.

3.2 Selection for Question Answering

In Question Answering (QA) systems the answer to the user's question expressed in a natural language is found by comparing it with documents and next text snippets from documents. The goal is to find such a sentence or snippet that is most likely to include the answer. Next the answer is extracted. However, in this work we are focusing only on the first step: using the text similarity measures in comparing questions with documents and sentences. We assumed that a good similarity measure would improve the selection process, i.e. we applied the evaluation by application scheme. As a basis for the evaluation a QA system for Polish called *Borsuk* [10,11] was utilised. Borsuk is available on an open licence.

To asses the results we applied *Mean Reciprocal Rank* (MRR) which is typically used for the evaluation of QA systems:

$$MRR = \frac{1}{n} \sum_{i=1}^{N} \frac{1}{rank_i} \tag{4}$$

where n is the number of questions, N is the number of documents returned for a question (in our case it is constant for all questions) and $rank_i$ is the rank of a document including the correct answer for the question i.

The closer is the right answer to the top of the ranking the higher MRR is. The maximum value 1 is achieved, when for all questions the answers are returned on the top ranking positions. Position changes in the top part of the ranking have significant influence on MRR value, while rank changes on further positions have very limited effect on MRR.

In all cases questions were compared with individual sentences with the help of the proposed measures. The whole QA process was evaluated in three *accuracy modes*:

1. *document*,
2. *overlapping snippet*
3. and *exact match* of a snippet and the expected answer.

In the *document mode*, score for a document is calculated as the maximum over the scores of its sentences. However, in the case of documents including the answer, it is not checked if the selected maximum score sentence really includes the answer.

In both snippet modes, i.e. the last two, the ranking is based on the scores of single sentences. Sentences are expanded to snippets of ±m sentences around the analysed sentence. The score of the snippet equals to the score of the central sentence. However, in the *overlapping snippet* mode, it is enough for the analysed snippet to overlap with the snippet annotated as including the answer in the test dataset to be considered as a positive choice. Contrary, in the *exact mode*, the analysed snippet must match exactly the snippet marked as the answer.

200 questions have been randomly selected from *Czy wiesz* dataset [11] for the needs of the evaluation. In all tests, questions were first processed by *Borsuk* and next for each question 50 top-ranked documents found in the searching step were returned.

The proposed measures were used to re-ranked documents and text snippets. Questions and sentences (from documents and snippets) were represented as weighted collections of the plWordNet synsets, see Sect. 2 As there is no large corpus mapped to synsets, IDF weights were calculated locally on the basis of the 50 returned documents only. So, the IDF values describe the local specificity of synsets that could introduce some accidental bias. The results are presented in Table 2.

Table 2. Evaluation of similarity measures for short texts in the application to Question Answering.

Key	Coll.	Weight	WSD	Documents	Overlapping			Exact		
					m=0	m=1	m=2	m=0	m=1	m=2
Cosine										
Lemmas	CollV2	y	–	0.6092	0.2287	0.4490	0.4798	0.0698	0.3436	0.4611
Lemmas	CollV1	n	–	0.5747	0.2173	0.4171	0.4512	0.0734	0.3244	0.4315
Synsets	Coll-HHM	y	30%	0.6122	0.2217	0.4550	0.4831	0.0705	0.3536	0.4590
Synsets	Coll-HHM	y	All	0.5883	0.2082	0.4410	0.4695	0.0637	0.3464	0.4495
Synsets	CollV2	y	30%	0.6011	0.2195	0.4496	0.4784	0.0750	0.3513	0.4592
Jaccard										
Lemmas	CollV2	y	–	0.5984	0.2118	0.4324	0.4632	0.0737	0.3297	0.4427
Lemmas	CollV1	y	–	0.5971	0.2239	0.4375	0.4637	0.0698	0.3421	0.4512
Synsets	CollV2	y	Best	0.5594	0.2287	0.4311	0.4557	0.0744	0.3289	0.4367
Unweighted correction factor × Cosine										
Synsets	Coll-HHM	y	30%	0.6231	0.4191	0.5059	0.5334	0.1889	0.3914	0.5308
Synsets	CollV2	y	All	0.6676	0.4068	0,5253	0.5533	0.1821	0.4108	0.5457
Weighted correction factor × Cosine										
Lemmas	CollV2	y	–	0.6345	0.3514	0.5177	0.5439	0.1305	0.4006	0.5353
Synsets	CollV2	y	All	0.6583	0.3441	0.5471	0.5735	0.1246	0.4307	0.5635
Basic configuration of QA system *Borsuk*										
–	–	–	–	0.8380	0.5369	0.7376	0.7697	0,2334	0.5861	0.7647
Borsuk enhanced with the selected similarity measures										
Synsets	CollV2	y(w=0.10)	30%	0,8471	0,5578	0,7688	0,7996	0,2407	0,6054	0,7946
Synsets	CollV2	y(w=0.08)	30%	0.8473	0.5553	0.7688	0.7996	0.2407	0.6020	0.7946
Synsets	CollV2	y(w=0.07)	All	0.8439	0.5521	0.7633	0.7958	0.2389	0.5988	0.7908
Synsets	Coll-HHM	y(w=0.08)	30%	0.8427	0.5513	0.7642	0.7951	0.2406	0.5989	0.7901

As *Borsuk* is based on the *Lucene* [15], we tested also how the proposed measure can fit into the scheme of the *Lucene Practical Scoring Function*, that is a complex equation with several constituents. Following the unweighted correction factor:

$$coord = |q \cap s|/|q| \tag{5}$$

we proposed proportion correction factor in order to decrease accidental similarity of questions to short sentences:

$$coord_w = \frac{2\sum_{i=1}^{n} q_i s_i}{\sum_{i=1}^{n} q_i^2} \tag{6}$$

where q_i and s_i are weights in the vector representations of, respectively, a question and sentence (from a document).

Similarity measure was multiplied by the correction factors in the second group of tests. Selected best results for different measure variants are presented in Table 2. The best MRR scores for whole documents without correction factor were above 0.6, i.e. the proper answer was mostly on the first or second position. Concerning the results for snippets, we can notice that the accuracy of selecting one sentence as the answer is low, but it is also the case of the whole QA system *Borsuk*. One of the reasons for this is the fact that answers are often scattered across whole paragraphs or even larger blocks of text in documents. Cosine measure expressed better results than Jacquard and Dice (not shown in Table 2). However, the advantage of the cosine measure was mostly due to better treatment of short sentences from the test documents.

Multiplication by the correction factor improved the overall results in most cases. The increases were especially significant for the comparison of questions with sentences and text snippets. The best results were achieved for the broader versions of the similarity measures, i.e. expanding lemmas or synsets with larger number of relations. The similarity measure represents the lexical component in the complex comparison while the correction factor expresses the search heuristic. It is worth to notice, that in all tests the collection CollV2 with only ‘positive’ relations, i.e. without antonymy and converse (a specific type of antonymy) produced better results than CollV1 including both potentially ‘noisy’ relations[11]. The variants based on the exact choice of WSD were worse, but in the case of 30% and ‘all’ synsets used for building collections, the results of WSD are visible in the weights assigned to the collection elements.

On the basis of the tests, three measure configurations: ⟨synsets, Coll-HHM, 30%⟩, ⟨synsets, CollV2, 30%⟩, ⟨synsets, CollV2, all⟩ – were selected for the tests inside the full *Borsuk* system. In all cases $m = 2$ was set for extracting snippets and the weighted correction factor.

[11] The opposition relations resulted in improvement in the application of the word expansion algorithm, as lemmas linked by the opposition relations are often very similar according to the Distributional Semantic methods, e.g. word embeddings.

3.3 Inside QA System

The selected measures have been added to the QA system *Borsuk* as an additional knowledge source for ranking the potential answers. The goal was to check if the use of a similarity measure can improve the overall performance of *Borsuk*. The optimised values for *Borsuk* parameters [11] were applied.

The same set of 200 questions was used. For each question only the 50 top scored documents were analysed.

Borsuk ranks documents and text snippets according to a complex measure defined as a linear combination of several individual measures. In order to included the measure proposed here as a component in the complex one several weight values were tested. The final values are provided together with results expressed by the enhanced *Borsuk* in Table 2.

The introduction of the proposed measure into *Borsuk* ranking improved MRR by 0.009 for documents and by 0.01-0.03 for text snippets. These differences, as well as differences for text snippets are statistically significant according to Wilcoxon test [25]. The differences may seem small, but the baseline of the optimised *Borsuk* was high and the observed increase of MRR was caused by improved positions of documents and snippets in the top part of the ranking and minor drops in the further part of the ranking. Manual inspection of the results showed that in many cases the shift was from the more remote ranking positions to the top three positions.

4 Conclusions

Semantic similarity measures for short texts were proposed. They are based on the description of lexical meanings in terms of the lexico-semantic relations provided by plWordNet. Text words are mapped onto semantic representation that is similar for words of the similar meanings. Some of the proposed measures showed improvement in the Question Answering system that can be attributed to better performance in selecting documents and text snippets. Comparison of the produced similarity values for sentence pairs showed better correlation than a baseline solution based on a commonly used vector model. Half of the proposed methods utilise results of the WSD tool and produced good results that were better than we could expect taking into account the limited accuracy of the WSD tool. With the use of a better WSD tool the performance of the similarity methods can be improved. A wide set of wordnet relations was applied, but still selection of the final set and optimal assignment of weights to relation links is an open issue for further research.

Acknowledgments. This work was co-financed as a part of the investment in the CLARIN-PL research infrastructure (www.clarin-pl.eu) funded by the Polish Ministry of Science and Higher Education.

References

1. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 305–316. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85836-2_29
2. Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W.: *SEM 2013 shared task: semantic textual similarity, including a pilot on typed-similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM) - Proceedings of the Main Conference and the Shared Task, vol. 1, pp. 32–43. Association for Computational Linguistics, Atlanta (2013)
3. Bär, D., Biemann, C., Gurevych, I., Zesch, T.: UKP: computing semantic textual similarity by combining multiple content similarity measures. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics, pp. 435–440. ACL (2012)
4. Corley, C., Mihalcea, R.: Measuring the semantic similarity of texts. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pp. 13–18. ACL (2005)
5. Fellbaum, C. (ed.): WordNet - An Electronic Lexical Database. The MIT Press, Cambridge (1998)
6. Kędzia, P., Piasecki, M., Orlińska, M.J.: Word sense disambiguation based on large scale polish clarin heterogeneous lexical resources. Cognitive Studies 14 (2015). To appear
7. Kouylekov, M., Magnini, B.: Recognizing textual entailment with tree edit distance. In: Proceedings of the PASCAL RTE Challenge, pp. 17–20 (2005)
8. Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. **18**(8), 1138–1150 (2006)
9. Liu, H., Wang, P.: Assessing text semantic similarity using ontology. J. Softw. **9**(2), 490–497 (2014)
10. Marcinczuk, M., Radziszewski, A., Piasecki, M., Piasecki, D., Ptak, M.: Evaluation of baseline information retrieval for polish open-domain question answering system. In: Angelova, G., Bontcheva, K., Mitkov, R. (eds.) Recent Advances in Natural Language Processing, RANLP 2013, 9–11 September 2013, Hissar, Bulgaria, pp. 428–435. RANLP 2011 Organising Committee/ACL (2013). http://aclweb.org/anthology/R/R13/R13-1056.pdf. ACL Anthology
11. Marcińczuk, M., Radziszewski, A., Piasecki, M., Piasecki, D., Ptak, M.: Open dataset for development of Polish question answering systems. In: Proceedings of 6th Language & Technology Conference LTC 2013, Poznań (2013)
12. Maziarz, M., Piasecki, M., Rudnicka, E., Szpakowicz, S.: plWordNet as the cornerstone of a toolkit of lexico-semantic resources. In: Proceedings of the Seventh Global Wordnet Conference, pp. 304–312 (2014). http://aclweb.org/anthology/W14-0142
13. Maziarz, M., Piasecki, M., Rudnicka, E., Szpakowicz, S., Kędzia, P.: plwordnet 3.0 - a comprehensive lexical-semantic resource. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 11–16 December 2016, Osaka, Japan, pp. 2259–2268. ACL (2016). http://aclweb.org/anthology/C/C16/

14. Maziarz, M., Piasecki, M., Szpakowicz, S.: The chicken-and-egg problem in wordnet design: synonymy, synsets and constitutive relations. Lang. Resour. Eval. **47**(3), 769–796 (2013). http://link.springer.com/article/10.1007 15pkt
15. McCandless, M., Hatcher, E., Gospodnetic, O.: Lucene in Action. Manning Publications, Greenwich (2010)
16. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI 2006, vol. 1. pp. 775–780. AAAI (2006). http://dl.acm.org/citation.cfm?id=1597538.1597662
17. Piasecki, M., Ramocki, R., Kaliński, M.: Information spreading in expanding wordnet hypernymy structure. In: Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013, pp. 553–561. INCOMA Ltd., Hissar, September 2013. http://www.aclweb.org/anthology/R13-1073
18. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Wrocław University of Technology Press, Wrocław (2009)
19. Pourgholamali, F., Kahani, M.: Semantic role based sentence compression. In: 2nd International eConference on Computer and Knowledge Engineering (ICCKE), pp. 210–214. IEEE Computer Society (2012)
20. Przepiórkowski, A., Wróblewska, A.: Supporting LFG parsing with dependency parsing. In: Dickinson, M., Hinrichs, E., Patejuk, A., Przepiórkowski, A. (eds.) Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT 14), pp. 168–178. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2015). http://nlp.ipipan.waw.pl/Bib/prz:wro:15.pdf
21. Punyakanok, V., Roth, D., Yih, W.: Mapping dependencies trees: an application to question answering. In: Proceedings of AI&Math, pp. 1–10, January 2004. http://cogcomp.cs.illinois.edu/papers/PunyakanokRoYi04a.pdf
22. Radziszewski, A.: A tiered CRF tagger for Polish. In: Bembenik, R., Skonieczny, Ł., Rybiński, H., Kryszkiewicz, M., Niezgódka, M. (eds.) Intelligent Tools for Building a Scientific Information Platform: Advanced Architectures and Solutions. SCI, vol. 467, pp. 215–230. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35647-6_16
23. Radziszewski, A., Śniatowski, T.: Maca – a configurable tool to integrate Polish morphological data. In: Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation (2011)
24. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1986)
25. Wilcoxon, F.: Individual comparisons by ranking methods. Biom. Bull. **1**(6), 80–83 (1945)

Methods of Linking Linguistic Resources for Semantic Role Labeling

Balázs Indig[1,2(✉)], Márton Miháltz[2], and András Simonyi[2]

[1] Pázmány Péter Catholic University, Faculty of Information Technology and Bionics, Práter u. 50/a, Budapest 1083, Hungary
[2] MTA-PPKE Hungarian Language Technology Research Group, Práter u. 50/a, Budapest 1083, Hungary
{indig.balazs,mihaltz.marton,simonyi.andras}@itk.ppke.hu

Abstract. This paper presents the process of enriching the verb frame database of a Hungarian natural language parser to enable the assignment of semantic roles. We accomplished this by linking the parser's verb frame database to existing linguistic resources such as VerbNet and WordNet, and automatically transferring back semantic knowledge. We developed OWL ontologies that map the various constraint description formalisms of the linked resources and employed a logical reasoning device to facilitate the linking procedure. We present results and discuss the challenges and pitfalls that arose from this undertaking. We also compare our rule-based approach with that of using a state-of-the-art English semantic role labeler pipeline for the thematic role transferring task.

Keywords: Linked resources · Ontology · Verb argument frames

1 Introduction

Semantic role labeling (SRL) is a significant step in enabling syntactically analyzed sentences to have basic semantic information in order to make sense of the meaning, making possible further applications such as semantic search, question answering, knowledge base development etc. [1]. The goal of SRL is to classify verb arguments into specific semantic roles to allow further processing at the predicate level. This paper details the process of enriching the verb frame database of a novel, psycholinguistically motivated Hungarian natural language parsing model [2,3] to enable the assignment of thematic roles (detailed in Sect. 1.1).

We describe a rule-based, ontology-driven approach to transferring semantic information by linking two verb frame databases which are different with respect to surface language forms but have a lot in common at the semantic level. We also present the results of using a state-of-the art statistical SRL system to assign thematic roles to numerous examples which are from parallel corpora. By analyzing the results achieved by the ontology-driven approach and comparing

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 183–197, 2018.
https://doi.org/10.1007/978-3-319-93782-3_14

it with that of the statistical semantic role labeler we were able to test the robustness of our method and assess its performance in real-life circumstances.

1.1 Our Parsing Model

ANAGRAMMA is a computational text understanding approach which does not follow the traditional parsing algorithms originated from information theory that are well-established in language technology, but uses some of the principles of human sentence processing set forth in [4]. The goal of this performance-based algorithm processes linguistic input no matter how ill-formed as long as the human parser could parse it.

We employ novel ideas, such as strict left-to-right operation. Each parsing step uses a trigram window where the first of the three tokens is processed by parallel threads, sometimes with the help of the two following tokens. The basic unit of processing is a (written) word. We may also think of the series of input words as a clock signal coordinating the work of the processing threads. The aforementioned parallel threads are overriding and correcting each other, which implement the matching of "offers" and "demands" representing different levels of linguistic knowledge, in a fashion similar to Categorial Grammars [5].

The first step is the morphological analysis, but because our focus is on syntactic processing we conceptualize this process as a black box, which provides the lemma of the input and those linguistic and non-linguistic features which serve as the basis for further processing. Some of these features called "demands", they create threads that look for suitable features that may satisfy them, while others called "supplies", which may satisfy the demand created by already existing or future threads [3].

For example the relationships between verbs and their arguments are detected by connecting the "offers" (lexical, morphological, and semantic properties) of potential arguments such as noun, adjective and adverbial phrases to the "demands" of verb argument positions [3]. The latter are introduced by looking up the sentence's finite verbs in a verb argument database consisting of more than 30,000 entries, developed for a machine translation project [6].

The process of 'caching substructures' is also well known in psycholinguistics: in human language comprehension we call it holistic processing. Further details about the parser can be found in [7]. In our model we mimic this property of human parsing: frequently occurring structures may enter the analysis with their full internal structure already in place. Multi-word expressions (proper nouns, conversation formulae, idioms, etc.) are processed in a similar way, but they do not have internal structures but behave as if they were written in a single word.

1.2 Extending Verb Frame Resources in Our Parsing Model by Linking

Our goal was to extend the aforementioned existing verb frame database with thematic role information to enable the assignment of semantic roles in the parser to allow further semantic processing. We accomplished this by linking the verb frame database to available external linguistic resources such as VerbNet [8] and WordNet [9], and by transferring as much semantic role information as possible. The linking was achieved by mapping the different constraint description formalisms of the source and target resources using two OWL ontologies and by employing the Racer OWL reasoner [10].

2 Related Work

Semantic role labeling was pioneered by [11]. CoNLL-2005 introduced a shared task to evaluate Semantic Role Labeling approaches [12]. [1] gives an in-depth overview. A recent work [13] boosts SRL with grammar and semantic type related features extracted with the help of a Chinese Treebank and Propbank.

There are several resources that link together structured linguistic databases for NLP applications. VerbNet, which we refer to in this paper is linked to PropBank, WordNet, FrameNet and OntoNotes Sense Groupings in the Unified Verb Index [14]. UBY is a large-scale lexical-semantic resource based on the Lexical Markup Framework (LMF) and combines various resources for English and German (WordNet, FrameNet, VerbNet, Wiktionary, OntoWiktionary) [15]. BabelNet is a multilingual encyclopedic dictionary and a semantic network which connects concepts and named entities in a very large network of semantic relations by integrating resources such as WordNet, Wikipedia, OmegaWiki, Wiktionary and Wikidata [16]. The Linked Open Data concept brings together many other different semantic and linguistic ontologies via semantic web technologies such as RDF links (e.g. [17]).

3 Resources

The verb frame database originates from the MetaMorpho Hungarian-to-English rule-based machine translation system [6], which uses deep syntactic analysis for the source language. It contains more than 30,000 verb frame patterns that represent the various possible argument configurations of over 17,000 Hungarian verbs. Each frame pattern contains a verb with lexical and morphological restrictions on it, and part-of-speech, semantic, morphological and (optionally) lexical restrictions that describe the verb's argument slots. Some argument positions are optional (are not required to be present in the sentence for the verb frame matching to hold).

For example, the following verb frame entry for "ábrándozik" (*to dream*) describes the equivalent of the English verb frame "somebody dreams about something": `HU.VP = SUBJ(human=YES) + TV(lex="ábrándozik") + COMPL#1(pos=N, case=DEL)`. Here, the first argument position (`SUBJ`, for subject) is restricted to phrases that have the `human` semantic property, while the second argument position (`COMPL#1`, for complement) is required to be a noun phrase in the *delative* case.

There are 27 binary semantic properties, representing semantic classes, and 54 further morphological and other grammatical features describing restrictions on the argument positions in the whole database. The verb elements of each verb frame entry are described by 6 grammatical features.

Since the verb frame database originates from a MT system, each entry describing a Hungarian verb frame also has an English translation equivalent. This English verb frame contains the English equivalent verb and argument positions equivalent to the Hungarian argument positions (and optionally more slots that introduce new tokens that constitute the semantically equivalent VP in English). The English equivalent of the verb frame shown above for “ábrándozik” is `EN.VP = SUBJ + TV(lex="dream") + COMPL#1(prep="about")`. This shows, for instance, that the argument slot (`COMPL#1`), which is expressed by a delative case marker in Hungarian, is expressed by a prepositional phrase headed by “about” in English.

Our central idea was to use the English verb frame equivalents to link the MetaMorpho (MMO) Hungarian verb frame database to an English verb semantic resource *at the argument level* in order to transfer thematic role information. We focused on VerbNet (VN), a high-quality and broad-coverage online verb lexicon for English [8,14]. It is organized into hierarchical verb classes extending Levin's classes [18]. Each verb class in VN contains syntactic descriptions (syntactic frames), and selectional restrictions (such as semantic types and syntactic properties) on the arguments, whose thematic roles are also described. Continuing our example, the Hungarian verb frame entry for “ábrándozik” can be mapped to the following VN frame entry for its English translation, “dream” (which belongs to the `wish-62` VN verb class):

```
NP V NP
Experiencer V Theme<-sentential>
```

By using the mapping between Hungarian MMO, English MMO and English VN arguments in the linked entries, we can infer that the thematic role of the `SUBJ` argument of the Hungarian verb “ábrándozik” in the above verb frame is *Experiencer*, while the other argument (`COMPL#1`) is a *Theme*.

In VN, in contrast to the flat list structure of MMO, verbs are grouped into classes according to the similarity of their frames, and each class may contain multiple frames that are valid for all verbs in the class. There is a class hierarchy, which means that classes may have subclasses and subclasses inherit properties from the higher classes and may specify them further. See detailed figures in Table 1.

Table 1. Verbs in VerbNet

Description	Number of verbs
Verbs in VerbNet	6343
Has no frame, only mentioned in other resources	2057
Has frames, possible to link	4286
Verbs occurring in only one class	2957

There is a ratio of about 1 to 10 between the number of verb frames and unique verbs in MMO, as seen in Table 2. This is due to various idiomatic and other intricacies, which produce several different frames for the majority of verbs. This phenomena affects little more than the third of the rules. On the other hand, during the development of MMO it was not a goal to achieve good recall on the English side of the verbs. It was enough to keep the lexical coverage high on the Hungarian side and optimize the translation equivalents for the target language for precision, which presents a problem for linking.

Table 2. Verbs in MetaMorpho

Description	No
Number of verb frames	30292
Number of unique English verb stems	3505
Number of verb stems that are not in VerbNet	920
Verbs treated as misspelled or unknown by the spell checker	143
Idiomatic or otherwise restricted English verb frames	10694
Idiomatic or otherwise restricted Hungarian verb frames	8347

According to our measurements, 42% of the verbs in MMO are listed in multiple classes of VN. Consequently, in addition to the VN frames, the VN classes corresponding to MMO frames also had to be disambiguated. For a brief overview of MMO verbs see Table 2.

4 Linking the Resources

We used multiple knowledge sources such as WordNet and our ontologies (see Sect. 4.2 for details) to ensure that Hungarian verb frame entries in the MMO database are linked precisely to those entries in VN that correspond to them both syntactically and semantically, and incorrect links are eliminated.

The employed procedure was the following. First, we took English verbs contained by the resources and filtered out those that do not appear in both of them. Using this filtered verb set we created all possible connections between frames with identical English verbs, and used this maximal mapping as our baseline. In the subsequent steps we tried to reduce the number of incorrect links by applying different constraints on the mapping in an iterative development style.

In a given MMO–VN mapping the links between specific MMO and VN entries can be categorized into 5 different types:

(i) There might not be any linked VN entry.
(ii) Unambiguous (one-to-one) mapping: there is only one link, which can be either

(iia) correct or
(iib) incorrect.
(iii) Ambiguous (one-to-many) mapping: there are more than one links, and they either
(iia) include the correct mapping (if it exists) or
(iib) not (possibly because it does not exist).

Because of the different granularity and level of completeness of the two resources the baseline contained a large number of entirely unsatisfactory mappings of the types (iib) and (iiib). In particular, there were many verb frames that could be found only in one of the resources, in spite of the fact that the verb itself was present in both of them. It was part of our goal to identify these entries to ease later processing.

Before applying our constraints on the baseline mapping we further reduced the number of entries by selecting only those frames from MMO that do not have optional arguments and do not require reordering of the arguments either. These mono- and ditransitive verbs had a good coverage in the original baseline set.

To determine the real-life occurrence frequencies of various MMO verb frame types, we used the Verb Argument Browser (VAB) [19,20], a resource derived from the 180-million word Hungarian National Corpus [21]. The VAB contains analysis of 18.3 million finite verb clauses in which the finite verb and the heads of the nominal phrases that are either arguments of modifiers of the verb are annotated. We mapped the case markings of the VAB argument nominals to MMO verb frame terminology: nominative case=SUBJ, accusative case=OBJ, other case markings or postpositons=COMPL. Using these labels we counted the occurrences of each different verb frame type in the corpus. As you can see in Table 3, the top 4 types account for 88% of all verb occurrences in the corpus. Based on this, we only considered the intransitive, mono-transitive (object or complement with non-accusative case marking) and ditransitive (object and complement) frames in the further stages.

Table 3. Verb frame type occurrences in the Hungarian National Corpus

Type	Occurrences	%
`SUBJ TV OBJ`	5,535,334	30.22%
`SUBJ TV COMPL#1`	4,501,736	24.57%
`SUBJ TV OBJ COMPL#1`	3,859,952	21.07%
`SUBJ TV`	2,465,005	13.46%
(13 more types)	1,957,700	10.68%
Total:	18,319,727	

On this reduced set we successively applied our different constraints and checked the differences between the mappings before and after each application. In applying and fine-tuning each constraint our goal was to filter out ambiguous and incorrect links keeping as many good connections as possible.

4.1 Filters

The first constraint that was used to filter the links in the baseline mapping required the number of arguments of the linked MMO and VN frames to be equal. This step required some conversion, because in VN prepositions are treated as separate elements of the verb frames whereas in MMO prepositions are properties of the argument slots.

As a further constraint we checked whether the verb on the Hungarian side of the MMO entry had a similar meaning to that of the English verb on the VN side. The satisfaction of this constraint could be checked only for a small fraction of the links since the available mappings between MMO and the Hungarian WordNet, on the one hand, and the Hungarian WordNet and Princeton WordNet, on the other, are incomplete. It was also checked whether the two sides of the MMO entry correspond to the same synset in WordNet.

Restrictions on argument slots of prepositional verb phrases provided an additional constraint for filtering: the prepositional restrictions had to be identical, or at least compatible for each argument position of the linked verb frames. In contrast to MMO, which specifies concrete prepositions in its descriptions of English prepositional verb frames, VN organizes prepositions into a class hierarchy and its restrictions frequently indicate only a preposition class. In these cases only the compatibility of the two prepositional restrictions could be checked by testing whether the preposition required by the MMO entry is a member of the preposition class in the VN entry.

The last two constraints that were used for filtering the links required that the syntactic and semantic restrictions in the linked MMO and VN entries had to be compatible for all argument positions. In contrast to the constraints used for the previous filters, the formalisms in which the two resources describe these restrictions were so different and, especially in the case of semantic selectional restrictions, so complex that it became necessary to introduce explicit formal representations of their logical relations in the form of two manually created OWL ontologies, and to use an OWL reasoner to check the compatibility of the restrictions. For a brief overview of the number of verbs linked by the application of the aforementioned filters see Table 3.

4.2 The Ontologies and the Reasoner

The Syntactic Restriction Ontology. While VN relies on a rich repertoire of more than 40 features to describe syntactic restrictions, MMO's descriptions of English frames make use only of the attributes *clausetype* (6 possible values), *poss*(essive), *num*(ber) and *tense* (3 possible values). The syntactic restriction ontology we have created represents all syntactic VN features and all possible syntactic MMO attribute/value combinations by OWL classes, and encodes their logical relationships by equivalence axioms of varying complexity (e.g., MMO's *poss* and VN's *genitive* features were simply stated to be equivalent, but VN's *sentential* feature was expressed as a boolean combination of 7 different MMO attribute/value pairs).

The Semantic Restriction Ontology. Both VN and MMO describe selectional restrictions on verbal argument positions in terms of boolean combinations of a small number of semantic categories that are organised into ontologies. However, the two ontologies are very different: both of them contain categories that are difficult to relate to those of the the other ontology (e.g., MMO's *punct* (punctuation) or VN's *communication*), and they interpret seemingly identical categories strikingly differently (e.g., in MMO's categorisation events can be *abstract*, while VN considers *event* and *abstract* to be disjoint categories).

In view of these differences, we decided to represent the logical relationships between the selectional categories of the two systems in a single, manually created semantic restriction ontology that contains both original ontologies, together with a number of bridging concepts and axioms. The bridging concepts are high-level concepts taken from the EuroWordNet top ontology [22], which served as a starting point for the development of the VN selectional ontology [8]. They are organizational devices that help expressing logical relations between MMO and VN categories in a succinct and conceptually clear form. For instance, although both ontologies contain several functional categories such as *drink* (MMO) or *instrument* (VN), neither of them had EuroWordNet's general *function* category. Adding this concept to the OWL ontology enabled expressing generalisations about functional categories (e.g., that they are all subcategories of VN's *concrete* category). Since neither MMO's nor VN's selectional restriction ontology has a detailed documentation clarifying the intended interpretation of all categories they use, in the case of many categories bridging axioms were added on the basis of a careful analysis of their actual usage in the resources.

The ontology represents bridging concepts and selectional categories by OWL classes whose names follow a uniform naming scheme that encodes their source (VN, MMO or EuroWordNet) by suffixes. There are no named individuals or properties, and axioms are limited to stating that one of the `subClassOf`, `equivalentClass` or `disjointWith` relations holds between certain boolean combinations of classes.

The Reasoner. The two restriction ontologies described so far reduced the problem of determining the compatibility of MMO and VN selectional restrictions to a reasoning problem: a pair of restrictions is compatible if and only if the restriction ontology does not imply that the corresponding (typically complex) ontology classes are disjoint. The general solution to this problem required the introduction of a reasoner software component into our system. Since the two ontologies consist only of boolean axioms, a simple propositional reasoner would have been sufficient, but because of its maturity and excellent support of the OWL format we used the open source version of the Racer OWL reasoner [10], which the system accessed via the OWLlink client-server protocol [23].

5 A Parser-Driven Approach

MMO as a rule-based translation system includes simple example sentences for every verb frame translation rule, which are supposed to match exactly the rule they belong to. These sentences were used as regression tests since each sentence had to trigger only the rule it belonged to. We used these example sentences to obtain corresponding VN frames and thematic roles for the MMO verb frames in our gold standard data set and compared the results with our annotations.

Naturally, we had to add the actual sequence of thematic roles for the manually found MMO–VN links in the gold standard as previously it contained only VN classes and frames without that information. Those MMO frames in the gold standard that had no corresponding VN class and frame pair were manually annotated with thematic roles. Using this new gold standard data set of 400 MMO verb frames and the corresponding thematic roles we were ready to measure the results obtained by using a state-of-the-art English semantic role labeler.

First, we translated the Hungarian example sentences with MMO to English. This was an important step since other translation systems would most probably have produced English predicates different from the desired ones, which were exactly the corresponding verbs in the MMO frame database. Having obtained the English versions of the Hungarian example sentences, we ran an SRL system on them that was capable of identifying predicates and labeling their arguments with semantic roles. Based on its performance and availability we chose the state of the art PathLSTM semantic role labeler [24], which utilizes lexicalized dependency path embeddings and certain binary features to identify and label semantic arguments. For tokenization, dependency parsing, and semantic predicate identification and disambiguation we used the pipeline described in the documentation of the PathLSTM source code [25], which consists of the Stanford CoreNLP WSJ tokenizer [26], the Bohnet dependency parser [27], and the mate-tools semantic role labeler [28]. PathLSTM was run with a model supporting PropBank role labeling and the resulting labels were transformed into VN thematic roles via the SemLink project's PropBank–VN mapping. [14][1]

We took only the main predicates into account that matched the verb on the English side of the corresponding MMO rule. The other identified predicates were excluded. As the used PropBank–VN SemLink mapping did not always produce unique and fully matching VN frames for the identified PropBank predicates and arguments we introduced the following rules for dealing with frame ambiguity and partial matches: For each VN frame corresponding in SemLink to a parsed PropBank predicate, if the frame had an element that did not occur in the parse then it was considered a partial match, else a full match. If there were full matches for a predicate then we dropped the partial matches and selected the element with the broadest coverage. We did the same when there were only partial matches available. We preferred those partial matches where the VN frame had fewer arguments than in the parse and the other cases were considered only after them. Relying on these rules we could assign the best matching VN frame and thematic roles to each sentence.

[1] The whole system with pretrained models can be downloaded at https://github.com/microth/PathLSTM.

6 Results

6.1 Filtering

To measure the performance of our system we created a random sample of 400 MMO entries from the output of the last filter. Ambiguous entries (with a one-to-many mapping in the output) and unambiguous ones (with a one-to-one mapping) were treated equally. The sample was processed by two independent annotators and unified by a third one. The sample contained 90 MMO entries that had no corresponding entry in VN. These entries were removed and the remaining entries together with their manually determined VN links constituted our gold standard.

Table 4. The number of links after subsequent filters

Description	No. of linked entries (unambiguous/ ambiguous)
Baseline set	431/26,560
Possible reordering needed	291/12,664
The lengths of MMO Hungarian and English sides are not equal	285/12,347
Mono- and ditransitive constructions	267/10,146
Equal no. of arguments both in MMO and VN	2301/7,745
WordNet mapping	2181/6,858
Prepositional restrictions	2929/4,610
Ontology (semantic restrs)	2967/4,455
Ontology (both)	2733/3,286

Table 5. Precision and number of links after subsequent filters with regard to the gold standard

Description	No. of linked entries (unambiguous/ ambiguous)
Baseline set	100% (9)/98.38% (183)
Possible reordering needed	100% (9)/98.38% (183)
The lengths of MMO Hungarian and English sides are not equal	100% (9)/98.38% (183)
Mono- and ditransitive constructions	100% (9)/98.38% (183)
Equal no. of arguments both in MMO and VN	100% (114)/96.29% (78)
WordNet mapping	100% (101)/97.14% (68)
Prepositional restrictions	90.43% (104)/79.62% (43)
Ontology (semantic restrs)	90.98% (111)/76.59% (36)
Ontology (both)	92.59% (100)/70.83% (17)

Since the gold standard was not representative of the whole MMO database and we considered only those entries from each test set that were in the gold standard, only the precision of the results could be assessed reliably. We checked each filter's output in the following way: if an MMO entry was unambiguously mapped and the mapped VN entry was identical to the one specified by the gold standard then it was considered correct, otherwise it was incorrect. In the ambiguous case set containment was used instead of equality: if the correct VN entry was in the set of linked entries then the mapping was considered correct, otherwise it was incorrect.

As can be seen in Table 4, the final mapping that was produced by our procedure contained four times more unambiguous links than the baseline, while the number of ambiguous links was radically reduced. The figures in Table 5 show that the precision of the filters described in Sect. 4.1 was nearly perfect in the case of those unambiguously mapped MMO entries for which the gold standard specified a valid corresponding VN entry. As for ambiguous mappings, they were regarded correct if the right entry was among the linked entries, but these numbers could be weighted by the number of links, which would lead to lower values.

6.2 Parser-Driven Approach

We used label-based and sentence-based evaluation (see Table 6), and only the precision of the parses was considered. In total 429 sentences were parsed but only 327 sentences had at least one argument with a thematic role left after checking the frame consistency checking phase.

Table 6. Result of the parser based thematic role labeling task

	Good	All	Precision (%)
No. of Labels	428	602	71.096
No. of Frames	193	327	59.021

The gold standard data set contained mainly simple verb frames where one can easily translate arguments from English to Hungarian as no argument reordering is needed. In the case of the few examples where the arguments were reordered during translation we compared the automatic result to the thematic roles of the English language sentences, as it is a trivial task to reorder the arguments for specific rules in the translation system ensuring that the identified thematic roles match the correct Hungarian arguments.

7 Discussion

A number of issues made the linking of MMO and VN entries more than a trivial exercise. Some of these obstacles arose from inherent problems in the used resources.

On the one hand, the MMO verb frame database was not conceived as a general-purpose resource for NLP applications, but rather to support a specific MT system. As a consequence, the lexical coverage of verbs in the English side is low, compensated by paraphrase-like translations which are hard to look up in a lexical resource such as VerbNet. The English MMO verb frames also include a large number of idioms or semi-compositonal structures (one or more of the arguments are bound lexically, eg. *take part in sg., make room for sg.* etc.), which are totally absent from VerbNet. Furthermore, while the features used for specifying selectional restrictions in the Hungarian verb frames fare well within the original MT system, the lack of a strict and formal system presents challenges when mapping to another feature system.

On the other hand, VerbNet has recursive, complex selectional restriction feature expressions, which are hard to process (4.2). Even though VN is an elaborate resource, the semantic features and categories used in the syntactic frames are not well documented, or come from vaguely documented resources, which sometimes makes their interpretation difficult or a work of guessing. We found VN to be sometimes incomplete, for example, the only intransitive frame for "knock" (class `sound_emission-43.2`) marks the subject *Theme*, while we believe a frame with an *Agent* subject exists in English ("Somebody knocked.").

Finally, WordNet presents some problems of its own. Its noun hypernym hierarchy, which is very useful as a taxonomic network, represents a level of granularity which does not reflect general (domain-independent) language use (e.g., the immediate superclasses of "dog" cover its biological taxonomy), making graph distance-based inferences difficult. The differences between the data formats of various WordNet resources (Hungarian WordNet and different Princeton WordNet versions) also presented difficulties.

From the parser-driven approach we expected better results, but it turned out that the highly advanced statistical generalizations on which the semantic role labeler relies do not play well with the hand-crafted, linguistically motivated MMO resource we were experimenting with. The parsing results were highly inconsistent and many of the problems could have been fixed inside the parser. For example some inflected verbs resulted in non-existent PropBank classes, due to bad lemmatization. There were many cases in which the resulting predicates had nothing in common with the expected classes as some arguments were missing or some extra arguments were mistakenly detected. If a known verb is found then it would probably be better to choose from the existing frame patterns instead of trying to generalize them, as further processing usually relies on the completeness of the underlying resource. Due to this erroneous behavior the results obtained using the parser fell short of what could be expected from a highly advanced statistical parsing method. Consequently, we can draw the conclusion that currently our proposed rule-based method for the cross-language transfer of thematic roles yields better results than the parser-based alternative we described, although we expect a slight deterioration in the results if a larger number of possibly more complex examples is compared to an extended gold standard.

8 Conclusion

In this paper, we presented the verb frame database that is used in our Hungarian natural language parsing model, and our initiative to link it to the VerbNet English verb lexicon, by exploiting the available English verb frame translations. The goal was to transfer the thematic role information available in VerbNet to Hungarian verb frames. We created two ontologies to harmonize the different descriptive formalisms of the two resources, and applied a logic reasoner to disambiguate candidate links based on translations. While this methodology presents some issues and does not present a full-fledged solution, it enabled us to enrich our verb database with thematic role information in a way that did not require the costly manual processing of all resources.

We also experimented with a parser-driven approach that acquires the thematic roles from translated sentences, but this method utterly failed compared to the rule-based approach on a moderate sized gold standard data set because of the inconsistencies between the parser and the lexical resources. As more and more components come into play, the issue of inconsistency between the components assumes a major role that cancels the positive effects and yields worse results than fewer but consistent components and a more rigid rule-based approach.

Acknowledgements. An earlier, shorter version of this paper, in which the evaluation was based on a substantially smaller gold standard and a smaller set of frames (excluding complements), was presented at the 7th Language & Technology Conference in Poznań in 2015 [29]. Another paper detailing the publicly available underlying ontology was presented at LREC 2016 in Protorož [30].

References

1. Palmer, M., Gildea, D., Xue, N.: Semantic role labeling. Synth. Lect. Hum. Lang. Technol. **3**(1), 1–103 (2010)
2. Prószéky, G., Indig, B., Miháltz, M., Sass, B.: Egy pszicholingvisztikai indíttatású számítógépes nyelvfeldolgozási modell felé. In: XI. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, 2014, pp. 79–87 (2014)
3. Sass, B.: Egy kereslet-kínálat elvű elemző működése és a koordináció kezelésének módszere. In: XI. Magyar Számítógépes Nyelvészeti Konferencia, Szeged, 2015, pp. 309–311 (2015)
4. Pléh, C.: Formal connexity and pragmatic cohesion in anaphora interpretation. In: Text and Discourse Connectedness. In: Proceedings of the Conference on Connexity and Coherance, Urbino, pp. 137–52 (1989)
5. Morrill, G.: Categorial Grammar: Logical Syntax, Semantics, and Processing. Oxford University Press, Oxford (2010)
6. Prószéky, G., Tihanyi, L.: MetaMorpho: a pattern-based machine translation system. In: Proceedings of the 24th Translating and the Computer Conference, pp. 19–24 (2002)
7. Prószéky, G., Indig, B.: Magyar szövegek pszicholingvisztikai indíttatású elemzése számítógéppel. Alkalmazott Nyelvtudomány **XV**(1–2), 29–44 (2015)

8. Schuler, K.K.: VerbNet: a broad-coverage, comprehensive verb lexicon. Ph.D. thesis, University of Pennsylvania (2005)
9. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
10. Haarslev, V., Hidde, K., Möller, R., Wessel, M.: The RacerPro knowledge representation and reasoning system. Semant. Web J. **3**(3), 267–277 (2012)
11. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Comput. Linguist. **28**(3), 245–288 (2002)
12. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 shared task: semantic role labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL 2005, pp. 152–164. ACL, Stroudsburg (2005)
13. Ku, L.W., Virk, S.M., Lee, Y.H.: A dual-layer semantic role labeling system. In: ACL-IJCNLP 2015, p. 49 (2015)
14. Loper, E., Yi, S.T., Palmer, M.: Combining lexical resources: mapping between PropBank and VerbNet. In: Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg (2007)
15. Gurevych, I., Eckle-Kohler, J., Hartmann, S., Matuschek, M., Meyer, C.M., Wirth, C.: UBY - a large-scale unified lexical-semantic resource based on LMF. In: Proceedings of the 13th Conference of the European Chapter of the ACL (EACL 2012), pp. 580–590, April 2012
16. Navigli, R., Ponzetto, S.P.: BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012)
17. Schmachtenberg, M., Bizer, C., Jentzsch, A., Cyganiak, R.: Linking open data cloud diagram 2014 (2014). http://lod-cloud.net/
18. Levin, B.: English Verb Classes and Alternations: A Preliminary Investigation. University of Chicago Press, Chicago (1993)
19. Sass, B.: A unified method for extracting simple and multiword verbs with valence information. In: Proceedings of RANLP 2009, Borovec, Bulgaria, pp. 399–403 (2009)
20. Sass, B.: The verb argument browser. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2008. LNCS (LNAI), vol. 5246, pp. 187–192. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87391-4_25
21. Váradi, T.: The Hungarian National Corpus. In: Proceedings of the Second International Conference on Language Resources and Evaluation, Las Palmas, pp. 385–389 (2002)
22. Vossen, P., Bloksma, L., Rodriguez, H., Climent, S., Calzolari, N., Roventini, A., Bertagna, F., Alonge, A., Peters, W.: The EuroWordNet base concepts and top ontology. Technical report, EuroWordNet project (1998)
23. Liebig, T., Luther, M., Noppens, O., Wessel, M.: OWLlink. Semant. Web Interoper. Usability Appl. **2**(1), 23–32 (2011)
24. Roth, M., Lapata, M.: Neural semantic role labeling with dependency path embeddings. In: Proceedings of ACL 2016, Berlin, pp. 1192–1202 (2016)
25. Roth, M., Lapata, M.: PathLSTM, GitHub repository (2017). https://github.com/microth/PathLSTM
26. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Proceedings of ACL 2014: System Demonstrations, pp. 55–60 (2014)
27. Bohnet, B.: Very high accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 89–97. Association for Computational Linguistics (2010)

28. Björkelund, A., Hafdell, L., Nugues, P.: Multilingual semantic role labeling. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task, pp. 43–48. Association for Computational Linguistics (2009)
29. Indig, B., Miháltz, M., Simonyi, A.: Exploiting linked linguistic resources for semantic role labeling. In: 7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, pp. 140–144. Uniwersytet im. Adama Mickiewicza w Poznaniu, Poznań (2015)
30. Indig, B., Miháltz, M., Simonyi, A.: Mapping ontologies using ontologies: Cross-lingual semantic role information transfer. In: Chair, N.C.C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA), Paris, pp. 2425–2430, May 2016

Machine Translation

A Quality Estimation System for Hungarian

Zijian Győző Yang[1,2](✉), Andrea Dömötör[1], and László János Laki[1]

[1] MTA-PPKE Hungarian Language Technology Research Group,
Práter Street 50/A, Budapest 1083, Hungary

[2] Faculty of Information Technology and Bionics, Pázmány Péter Catholic
University, Práter Street 50/A, Budapest 1083, Hungary

{yang.zijian.gyozo,domotor.andrea,laki.laszlo}@itk.ppke.hu

Abstract. Quality estimation is an important field of machine translation evaluation. There are automatic evaluation methods for machine translation that use reference translations created by human translators. The creation of these reference translations is very expensive and time-consuming. Furthermore, these automatic evaluation methods are not real-time and the correlation between the results of these methods and that of human evaluation is very low in the case of translations from English to Hungarian. The other kind of evaluation approach is quality estimation. These methods address the task by estimating the quality of translations as a prediction task for which features are extracted from only the source and translated sentences. In this study, we describe an English-Hungarian quality estimation system that can predict quality of translated sentences. Furthermore, using the predicted the quality scores, we combined different kinds of machine translated outputs to improve the translation accuracy. For this task, we created a training corpus. Last, but not least, using the quality estimation method we created a monolingual quality estimation system for a psycholinguistically motivated parser. In this paper we summarize our results and show some partial results of ongoing projects.

Keywords: Quality estimation · Machine translation

1 Introduction

As machine translation (MT) has become popular among people and companies, the measurement of the translation output has become necessary. A quality indicator for MT could save a lot of time and money for users. Knowing the quality scores of machine translated segments can help human annotators in their post-edit tasks, or help MT systems to find and combine the best translations. Last but not least, quality indicators can filter out and inform about unreliable translations. There are two kinds of evaluation methods for MT. The first type uses reference translations, i.e. it compares machine translated sentences to human translated reference sentences, and measures the similarities or

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 201–213, 2018.
https://doi.org/10.1007/978-3-319-93782-3_15

differences between them. These methods are automatic evaluation approaches such as BLEU, and other methods based on BLEU, TER, HTER etc. The problem is that automatic evaluation methods cannot perform well enough in this task, because these need reference translations. It means that along with producing the automatic translation, we also have to create a human translation for the given sentence (for the sentences of the test set) to compare it to the machine translated output. Creating human translations is very expensive and slow. Thus, another approach is needed to solve these problems, i.e. a method which can predict translation quality in real-time and does not need reference translations.

Other types of evaluation methods do not use reference translations. The family of the supervised approaches is called Quality Estimation (QE) of MT. This method addresses the problem by evaluating the quality of machine translated segments as a prediction task. Using QE we can save considerable time and money for human annotators, researchers and companies.

In this study, we use the QE method to build a quality estimation system for English-Hungarian machine translation and for predicting the quality of Hungarian monolingual texts. We built the eπque (π Quality Estimation) system, which has three parts: English-Hungarian QE system (Hun-QuEst), MaTros composite MT system and πRate monolingual QE system.

The structure of this paper is as follows: First, we will shortly introduce our previous experiments on English-Hungarian corpora. Then, we will present experiments, methods and results of our recently developed methods that use our QE algorithm on English-Spanish machine translated texts and on Hungarian monolingual data.

2 Quality Estimation in Machine Translation

2.1 Quality Estimation

QE is a prediction task, where different quality indicators are extracted from the source and the machine translated segments. The QE model is built with machine learning algorithms based on these indicators. Then the QE model is used to predict the quality of unseen translations. The aim is that the QE model correlates with human judgments, thus the QE model is trained on human evaluations.

In the last couple of years, there have been many WMT workshops with quality estimation shared tasks[1]. The datasets are evaluated with HTER, METEOR, ranking or post-editing effort, etc. scores. Unfortunately, there hasn't been a Hungarian training set in this field. Thus, in the recent years, we built a quality estimation system for Hungarian. To implement this system, we created training corpora to train the QE models. We also did experiments in the field of feature selection and feature set optimization. Using the QE models, we combined MT

[1] http://www.statmt.org/wmt17/quality-estimation-task.html.

outputs to create a composite MT system. In addition, using the QE method, we built a monolingual QE system for a psycholinguistically motivated parser.

In the quality estimation task (see Fig. 1), using various features, we extract different kinds of quality indicators from the source and translated sentences without using reference translations. From the source sentences, complexity features can be extracted (e.g. number of tokens in the source segment). From the translated sentences, QuEst extracts fluency features (e.g. percentage of verbs in the target sentences). From the comparison between the source and the translated sentences, adequacy features are extracted (e.g. ratio of percentage of nouns in the source and target). We can also extract features from the decoder of the MT system. These are the confidence features (e.g. features and global score of the SMT system). We can divide the features into two more main categories: "black-box" features (independent from the MT system) and "glass-box" features (MT system-dependent). Since in our experiments we have translations from different MT systems, we did use only the "black-box" features. After the feature extraction, using these quality indicators, we can build the QE model with machine learning methods. The aim is that the predictions of the QE model are highly correlated with human evaluations. Thus, the extracted quality indicators need to be trained on human judgments.

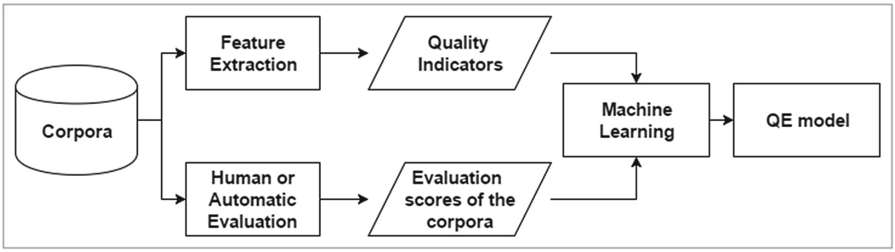


Fig. 1. The QE algorithm

2.2 The Hun-QuEst

The Hun-Quest [16,19] is a QE system for English-Hungarian. We used the QuEst framework [15], developed by Specia et al., to train and apply QE models for Hungarian.

Hungarian is an agglutinating and compounding language. There are significant differences between English and Hungarian, regarding their morphology, syntax and word order or number. Furthermore, the free order of grammatical constituents, and different word orders in noun phrases (NPs) and prepositional phrases (PPs) are also characteristics of Hungarian. Thus, features used in a QE task for English-Spanish or English-German, which produced good results, perform much worse for English-Hungarian. Thus, if we would like to use linguistic features in QuEst, we need to integrate the available Hungarian linguistic tools into it.

First, we had to integrate the available Hungarian linguistic tools into the QuEst framework. For Part-of-Speech (POS) tagging and lemmatization, we used PurePos 2.0 [11], which is an open source, HMM-based morphological disambiguation tool. PurePos 2.0 has the state-of-the-art performance for Hungarian. It has the possibility to integrate a morphological analyzer. Thus, to get the best performance, we used Humor [12], a Hungarian morphological analyzer. For NP-chunking, we used HunTag [13] that was trained on the Szeged Treebank [2]. HunTag is a maximum entropy Markov-model based sequential tagger.

After the integration of Hungarian tools, we performed experiments to find new semantic features. We used dictionaries, WordNet, LSI and embedding models to create new features. Using these features we could gain higher evaluation results than the baseline feature set.

To train the QE models, we created a training corpus, which is called HuQ corpus [17]. The corpus contains 1500 English-Hungarian segment pairs. To create this corpus, we translated 300 English sentences to Hungarian with four different machine translation systems. We used the Google Translate, the Bing Translate, the MetaMorpho [9] rule based MT system and the MOSES statistical MT toolkit [7]. Beside the MT systems we also used human-translated segments. The source was English, the target was Hungarian. All the 1500 segment pairs were evaluated by 3 human annotators. They used the Likert scale (1–5 scores) for evaluation. There are many cases, when we do not need 5 grades. For instance, the companies and translators need only 2 or 3 classes: needs post-edit – does not need post edit; correct – needs correction, etc. Thus, from the human evaluated scores, we also created 3 classification classes: BAD, MEDIUM, GOOD.

Using the HuQ corpus, we trained different QE models for English-Hungarian that can predict the quality of machine translated sentences in real-time.

2.3 The MaTros System

The MaTros system [8] is a composite MT system. We used QE models and methods to combine different MT systems to achieve higher translation accuracy. The advantage of our system is that we used sentence level QE calculation for the phrase-based (PB) and hierarchical-based (HB) MT system outputs to choose the best translation. The QE estimates the quality of the MT translated segment without reference translation. It is based on a statistical model trained by regression analysis. Quality indicators are extracted based on the source segment, the machine translated segment and inner parameters of the MT system. The QE model is trained based on these indicators and on human or automatic evaluation scores. After all, using the trained statistical model, we can predict the quality of the new unseen segments. In our research we translated the source segments with two different MT systems, then using a QE model, we chose the better quality translation as the final MT output.

3 Quality Estimation of Monolingual Texts

Quality estimation is not only relevant for machine translation but also for monolingual texts. Although these are created by native speakers, quality issues can still come up due to the non-standard use of language. Quality information about the input can be useful for any tool that works with monolingual texts, such as POS-taggers or parsers. Our motivation to use quality estimation for monolingual texts is to measure the quality of input for a performance-based, psycholinguistically motivated parser developed by the MTA-PPKE Hungarian Language Technology Research Group [6].

3.1 Monolingual Corpus

The error types of monolingual human texts are different from the ones that occured with machine translation outputs, therefore the monolingual quality estimation task needs different training data and features. The training and test corpora were extracted from the Hungarian Gigaword Corpus [10]. We queried random data from the spoken language and personal subcorpora which contain transcriptions of radio programmes and texts from internet forums and comments. We chose these kinds of texts because these presumably show significant deviance from the standard. The training data was annotated manually and based on linguistic aspects. We used two kinds of annotations: a Likert-score (1–5) and a classification model.

The Likert-score values represent the difficulty level of automatic parsing of the sentence. The scores were counted from the ratio of correctly parseable argument chunks. This measurement can inform the parser about the reliability of the input. However, the type of deviance from standard can be even more informative to an automatic tool: if it receives information about the nature of the problem, it can apply the appropriate correction module (spell checker, accent restorer etc.).

Our classification model uses five error types which are typical in spoken language and in informal texts written on internet. These are:

1. Errors (lack) of punctuation or omission of capital letters
2. Typos, orthographic or grammar mistakes
3. Data is not from the target language
4. Lack of accents
5. Hardly parseable spoken or informal texts (repetitions, slang expressions, abbreviations, emoticons etc.)

If a sentence contains more than one error types the we chose the most characteristic one. In addition, we also defined a class for segmentation fails in the corpus, in which cases the data gathered by the crawler was erroneously identified as a sentence.

The relevant linguistic features were also specified based on the analysis of corpus data.

3.2 The πRate System

The πRate system [18] is a task-oriented quality estimation system for monolingual natural parsing. Psycholinguistically motivated natural parsing is a new, human-oriented computational language processing approach. This complex real-time model has several parallel threads to analyze the input words, phrases or sentences. One of the main threads can be the quality estimation module, which informs, controls and filters the noisy or erroneous input. To build this quality controller module we implemented the quality estimation method that is traditionally used in the field of machine translation evaluation. Standard QE models do not perform well enough for this task, therefore we modified and optimized the architecture, that is inspired by the task-oriented architectures.

We have built the πRate system for the AnaGramma project [6].

4 Methods and Experiments

4.1 Quality Estimation for English-Spanish MT

In our previous research, we have built an English-Hungarian (En-Hu) QE system. We performed experiments with different kinds of feature sets and training sets to gain the highest performance of QE [16,19]. For training QE models, we used corpora that were evaluated by automatic metrics and human annotators. The corpora we used in our experiments for English-Hungarian are the following:

- C1 corpus: contains 1950 English sentences of mixed topics (literature, law, subtitles) from the Hunglish corpus [4].
- C2: subset of the C1 corpus containing 1500 English-Hungarian sentence pairs with human evaluated Likert scores.
- C3: subset of the C2 corpus containing 550 English-Hungarian sentence pairs.

We also created new semantic features for English-Hungarian. The most significant features are the WordNet features [16]. We would have liked to examine the utility of these features in other languages, thus in this research, we performed experiments to use our WordNet features for English-Spanish (En-Es) QE. Our recent research uses the system and WordNet features developed for Hun-Quest to predict the quality of English-Spanish machine translations.

To extract the WordNet features, we integrated a Spanish WordNet to our Hun-QuEst system. For this task, we used the MCR 3.0 (Multilingual Central Repository 3.0) [3].

Inspired by the research of quality estimation for Hungarian [16], to train the QE models we used the SVR (support vector regression) and the SVM (support vector machine) algorithms.

For training and testing, we used the training and test data provided by the quality estimation shared task of ACL 2014 Ninth Workshop on Statistical Machine Translation[2]. Our research focused on the sentence-level QE. We did 3 experiments from the sentence-level QE shared task:

[2] http://www.statmt.org/wmt14/quality-estimation-task.html.

- 1. Predict perceived post-editing effort (PPEE): The training set (C4) contains 3816 English-Spanish translation sentence pairs and the test set contains 600 sentence pairs.
- 2. Predict percentage of edits needed (HTER [14]): The training set (C5) contains 896 English-Spanish translation sentence pairs and the test set contains 208 sentence pairs.
- 3. Predict post-editing time (PET): The training set (C6) contains 650 English-Spanish translation sentence pairs and the test set contains 208 sentence pairs.

4.2 Quality Estimation of Monolingual Texts

In our recent research, we performed experiments to estimate reliable quality scores or classes for monolingual texts. For this task, we used the πRate monolingual QE system [18]. To train the πRate system, we used the corpus described in 3.1. For now, we have 1000 human evaluated sentences in the corpus.

For the machine learning task, we used the Weka system [5] to create classifiers with 10 fold cross-validation. We tried more machine learning methods: support vector machine (SVM), Support vector machine for regression (SVR), Linear regression, M5P Tree and J48. The support vector regression (SVR) and support vector machine (SVM) produced the highest results, thus further on, we show only the results of these two classifiers.

For building the QE model, monolingual features as quality indicators are needed, which are extracted from the monolingual corpus. In our experiment we had 33 different kinds of monolingual features. According to the functionality, we can separate the features into the following categories:

- linguistic features:
  - percentage of nouns, verbs, pronouns, adverbs, adjectives, conjunctions, determiners, preverbs, interjections in the sentence;
  - ratio of number of nouns and verbs in the sentence;
  - ratio of number of nouns and adjectives in the sentence;
  - ratio of number of verbs and preverbs in the sentence;
  - ratio of number of nouns and determiners in the sentence;
  - number of tokens;
  - average word length in the sentence;
- n-gram features:
  - sentence LM probability;
  - sentence LM perplexity;
  - LM probability of lemmas and POS tags of the sentence;
  - LM perplexity of lemmas and POS tags of the sentence;
- error features:
  - percentage of non-Hungarian words in the sentence;
  - percentage of accented words in the sentence;
  - percentage of unknown words in the sentence;
  - percentage of punctuation marks in the sentence.

To train the n-gram models (for the n-gram features), we used a subcorpus of the HGC [10] that contains 98500 lemmatized and POS tagged sentences.

Using the human evaluated Likert golden standard scores and the Classification tags, we built 2 QE models:

- LS model: QE model using the Likert scores.
- CS model: QE model using the Classification scores.

We also did experiments in the optimization task. According to machine translation evaluation [1], not all the features are relevant to the QE model. We used the forward selection method [19] to find the optimized feature sets:

- OptLS set: Optimized feature set for LS model.
- OptCS set: Optimized feature set for CS model.

5 Results and Evaluation

For evaluating the performance of our methods, we used the statistical correlation (COR), the MAE (Mean absolute error), the RMSE (Root mean-squared error) and the correctly classified instances (CCI) evaluation metrics. The correlation ranges from −1 to +1, and the closer the correlation to −1 or +1 is, the better it is. In the case of MAE and RMSE the closer the value to 0, the better.

5.1 Quality Estimation for English-Spanish MT

First, In Table 12 we can see the previous results of English-Hungarian QE experiments. We can compare the results of using different corpora were evaluated by different kinds of metrics (EvalMC - Evaluation using different Metrics and Corpora). Our optimized feature sets produced higher correlation results than the baseline set in all the cases. As we can see in Table 1 and in Table 2, increasing the corpora (size of C2 > size of C3), we could gain ∼1.5% higher correlation and ∼3.5% more CCI results.

Secondly, we implemented our WordNet features for English-Spanish. Using these features and PPEE, HTER and PET scores, we built the English-Spanish QE model. In Tables 1 and 2, we can see that adding our WordNet features to the baseline set, we could gain ∼1% higher correlation and correctly classified instances in all the cases.

The Spanish WordNet did not contain adverbs and did contain only 37 verb synsets, thus the WordNet features that extract adverb and verb synsets were unusable. According to our opinion, if there were more verbs in the WordNet, the result could be better.

5.2 Quality Estimation of Monolingual Texts

In Tables 3 and 4, we can see that the 33 feature set could gain ∼75,5% correlation, and ∼59% correctly classified instance results.

We did optimization with forward selection method. After optimization, as we can see in Tables 3 and 4:

Table 1. Evaluation of QE models using numeric evaluation metrics

Language	EvalMC	QE model	COR	MAE	RMSE
En-Hu	TER (C1)	103 features	0.3550	0.3275	0.4357
	BLEU (C1)	103 features	0.4404	0.2201	0.3474
	NIST (C1)	103 features	0.3669	2.6695	0.4777
	Human (C2)	Baseline model	0.5101	0.9333	1.1217
	Human (C2)	103 features	0.5851	0.8621	1.0739
	Human (C2)	Optimized 23 features	**0.6275**	**0.795**	**1.0292**
	Human (C3)	Baseline model	0.4931	0.8345	1.0848
	Human (C3)	103 features	0.5618	0.7962	1.0252
	Human (C3)	Optimized 26 features	**0.6100**	**0.7459**	**1.9775**
En-Es	HTER (C5)	Baseline model	0.4078	0.1444	0.2117
	HTER (C5)	Baseline + WordNet features	**0.4149**	**0.1438**	**0.2106**
	PET (C6)	Baseline model	0.6677	15170	22462
	PET (C6)	Baseline + WordNet features	**0.6715**	15228	**22354**

Table 2. Evaluation of QE models using nominal evaluation metrics

Language	EvalMC	QE model	CCI	MAE	RMSE
En-Hu	Human (C2)	Baseline model	57.80%	0.3433	0.4417
	Human (C2)	103 features	60.33%	0.3347	0.4318
	Human (C2)	Optimized 12 features	**61.80%**	**0.3299**	**0.4263**
	Human (C3)	Baseline model	53.55%	0.3546	0.4544
	Human (C3)	103 features	55.19%	0.3526	0.4518
	Human (C2)	Optimized 8 features	**58.47%**	**0.3404**	**0.4385**
En-Es	PPEE (C4)	Baseline model	58.67%	0.3450	0.4437
	PPEE (C4)	Baseline + WordNet features	**59.46%**	**0.3420**	**0.4403**

- The OptLS set, using only 9 features, could gain almost the correlation as the full feature set but with significantly less effort.
- The OptCS set, using only 12 features, could gain ~6% more correctly classified instances.

As we can see, the Likert-score model works with sufficiently high correlation but the classificational results show less efficiency. The lower results of the classificational model can be explained with the annotation methodology. As we already mentioned in 3.1, a sentence could be annotated with only one error class, however in reality one sentence may contain errors of several classes. That means, the classifier's task is to find the principal error class of the sentence which can be hard if it has other, less relevant errors of other classes.

Table 3. Evaluation of LS model and OptLS set

	Correlation	MAE	RMSE
LS model - 33 features	0.755	0.7675	1.0036
OptLS set - 9 features	0.7614	0.7206	1.0278

Table 4. Evaluation of CS model and OptCS set

	CCI	MAE	RMSE
Cs model - 33 features	59.1%	0.2153	0.3192
OptCS set - 12 features	65.2%	0.2138	0.3169

For this reason, we made an other measurement with less classes. The initial 6 classes were reduced to 3:

1. Punctuation and segmentation faults
2. Foreign language texts or texts without accents
3. Grammatical and orthographical errors

In Table 5 we can see the classificational results with the reduced number of classes.

Table 5. Evaluation of CS model and OptCS set with 3 classes

	CCI	MAE	RMSE
Cs model - 33 features	61.3%	0.2959	0.3781
OptCS set - 12 features	69.7%	0.288	0.3675

The results show that the percentage of correctly classified instances sightly improved (4,5%) with the reduction of classes.

In Tables 6 and 7 we can see the optimized feature sets (sorted by the degree of the improvement of the models).

For the prediction of quality, the most relevant linguistic features are in connection with accented characters, number of tokens and punctuation marks. This fact is comprehensible from a linguistic aspect. The most typical deviance from standard in informal written communication is the omission of accents and punctuation marks. The latter causes segmentation problems, so if a sentence is very long, the quality estimation system can suspect that it is an unedited text which is normally related to bad quality.

In the optimized feature set of the classification model appear linguistic features that are informative about grammar weaknesses, such as number of verbs/number of preverbs or number of pronouns. These are necessary to detect

Table 6. Optimized 9 features for the Likert QE model

Feature
Number of accented characters
Number of words with accented characters/number of all words
N-gram perplexity of the original segment including OOV
N-gram perplexity of tags of the segment include OOV
Number of tokens
Number of word punctuation marks/number of sentence punctuation marks
Number of skip tags
N-gram perplexity of stems of the segment include OOV
N-gram perplexity of POS tags of the segment include OOV

Table 7. Optimized 12 features for the Classification QE model

Feature
Number of accented characters
Number of words with accented characters/number of all words
Number of punctuation marks
Number of verbs/number of preverbs
Number of pronouns
N-gram perplexity of the original segment including OOV
Number of preverbs
Number of tokens
N-gram perplexity of tags of the segment including OOV
N-gram probability of tags of the segment
Number of verbs
Number of word punctuation marks/number of sentence punctuation marks

the members of the error class related to grammatical correctness. This seemed to be less relevant in the case of the Likert-model, which means that the majority of quality problems in monolingual text are not of grammatical nature in the strict sense of the word.

Table 8 shows the average Likert-scores of the error classes.

Table 8. Average Likert-scores of error classes

	Avg point
Lack of accents	1.16
Data is not from the target language	1.63
Corpus segmentation faults	1.74
Typos, orthographic or grammar mistakes	2.69
Errors (lack) of punctuation or omission of capital letters	3.20
Hardly parseable spoken or informal texts	3.28

6 Conclusion

According to our recent research, our quality estimation method is language-independent. The difficulty of transferring the method from one language to another lies in the availability of resources. As we saw in the experiments of Spanish, the size of the WordNet that we used for extracting features, determined the improvement of the results.

In the case of quality estimation of monolingual texts, the error types are different than in multilingual data. While machine translation outputs usually contain grammatical errors, the main quality problems of monolingual texts come from the writing habits of internet users, namely from the omission of accents and punctuation marks. The detection of the error types still needs improvement, however we achieved good results in the Likert-scale quality estimation of Hungarian monolingual data.

As general observation of our research, we can also mention the importance of feature optimization. Our results show that we can achieve the highest correlations using a reduced feature set, which saves us resources. The optimal feature set is task-dependent. The most relevant features were different in the cases of multilingual and monolingual texts and also in Likert and classification models.

References

1. Beck, D., Shah, K., Cohn, T., Specia, L.: SHEF-Lite: when less is more for translation quality estimation. In: Proceedings of the Workshop on Machine Translation (WMT), August 2013
2. Csendes, D., Csirik, J., Gyimóthy, T., Kocsor, A.: The szeged treebank. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 123–131. Springer, Heidelberg (2005). https://doi.org/10.1007/11551874_16
3. Gonzalez-Agirre, A., Laparra, E., Rigau, G.: Multilingual central repository version 3.0. In: Calzolari, N., Choukri, K., Declerck, T., Dogan, M.U., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S. (eds.) LREC, pp. 2525–2529. European Language Resources Association (ELRA) (2012)
4. Halácsy, P., Kornai, A., Németh, L., Sas, B., Varga, D., Váradi, T., Vonyó, A.: A Hunglish korpusz és szótár. In: III. Magyar Számítógépes Nyelvészeti Konferencia. Szegedi Egyetem (2005)

5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009). http://doi.acm.org/10.1145/1656274.1656278
6. Indig, B., Vadász, N., Kalivoda, Á.: Decreasing entropy: how wide to open the window? In: Martín-Vide, C., Mizuki, T., Vega-Rodríguez, M.A. (eds.) TPNC 2016. LNCS, vol. 10071, pp. 137–148. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49001-4_11
7. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 2007, pp. 177–180. Association for Computational Linguistics, Stroudsburg (2007). http://dl.acm.org/citation.cfm?id=1557769.1557821
8. Laki, L.J., Yang, Z.G.: Combining machine translation systems with quality estimation. In: Computational Linguistics and Intelligent Text Processing, Budapest, Hungary (2017)
9. Novák, A., Tihanyi, L., Prószéky, G.: The MetaMorpho translation system. In: Proceedings of the Third Workshop on Statistical Machine Translation, StatMT 2008, pp. 111–114. Association for Computational Linguistics, Stroudsburg (2008). http://dl.acm.org/citation.cfm?id=1626394.1626405
10. Oravecz, C., Váradi, T., Sass, B.: The Hungarian Gigaword Corpus. In: Chair, N.C.C., et al. (eds.) Proceedings of the 9th International Conference on Language Resources and Evaluation. ELRA, Reykjavik, May 2014
11. Orosz, G., Novák, A.: PurePos 2.0: a hybrid tool for morphological disambiguation. In: RANLP 2013, pp. 539–545 (2013)
12. Prószéky, G.: Industrial applications of unification morphology. In: Proceedings of the Fourth Conference on Applied Natural Language Processing, pp. 213–214. Association for Computational Linguistics, Stuttgart, October 1994. http://www.aclweb.org/anthology/A94-1046
13. Recski, G., Varga, D.: A Hungarian NP Chunker. The Odd Yearbook. ELTE SEAS Undergraduate Papers in Linguistics, Budapest (2009)
14. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the Association for Machine Translation in the Americas, pp. 223–231 (2006)
15. Specia, L., Shah, K., de Souza, J.G., Cohn, T.: QuEst - A translation quality estimation framework. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 79–84. Association for Computational Linguistics, Sofia, August 2013. http://www.aclweb.org/anthology/P13-4014
16. Yang, Z.G., Laki, L.J.: Quality estimation for English-Hungarian machine translation. In: 7th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznan, Poland, pp. 170–174 (2015)
17. Yang, Z.G., Laki, L.J., Siklósi, B.: HuQ: an English-Hungarian corpus for quality estimation. In: Proceedings of the LREC 2016 Workshop - Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem
18. Yang, Z.G., Laki, L.J.: πRate: a task-oriented monolingual quality estimation system. Int. J. Comput. Linguist. Appl. **8** (2017)
19. Yang, Z.G., Laki, L.J., Siklósi, B.: Quality estimation for English-Hungarian with optimized semantic features. In: Computational Linguistics and Intelligent Text Processing, Konya, Turkey (2016)

Leveraging the Advantages of Associative Alignment Methods for PB-SMT Systems

Baosong Yang and Yves Lepage(✉)

IPS, Waseda University, Kitakyushu, Japan
yves.lepage@waseda.jp

Abstract. Training statistical machine translation systems used to require heavy computation times. It has been shown that approximations in the probabilistic approach could lead to impressing improvements (Fast align). We show that, by leveraging the advantages of the associative approach, we achieve similar, even faster, training times, while keeping comparable BLEU scores. Our contributions are of two types: of the engineering type, by introducing multi-processing both in sampling-based alignment and hierarchical sub-sentential alignment; of modeling type, by introducting approximations in hierarchical sub-sentential alignment that lead to important reductions in time without affecting the alignments produced. We test and compare our improvements on six typical language pairs of the Europarl corpus.

1 Introduction

Sub-sentential alignment, computed based on word associations, is the core of the training process in Phrase-based Statistical Machine Translation (PB-SMT). These two processes are crucial for the accuracy of translation, but they are also very time-consuming.

The IBM models [2] and the grow-diag-final-and heuristic are the most popular approach. They have been integrated as the GIZA++ tool [15], or MGIZA++ [6] for a parallel implementation, in the PB-SMT toolkit Moses[1]. A log-linear re-parameterisation of IBM Model 2 has been implemented in Fast align[2] [4]. It led to much faster training times.

IBM models are probabilistic models, so that the optimisation process requires the knowledge of the entire parallel corpus to estimate the parameters [14]. On the contrary, *associative* methods, as characterised in [5], do not

This paper is a part of the outcome of research performed under a Waseda University Grant for Special Research Projects (Project number: 2015A-063).

Y. Lepage—Thanks to Chonlathorn Kwankajornkiet from Chulalongkorn University, Thailand, for her contribution in implementing the C core component of Cutnalign during a training period at IPS, Waseda University.

[1] http://www.statmt.org.

[2] http://github.com/clab/fast_align.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 214–228, 2018.
https://doi.org/10.1007/978-3-319-93782-3_16

rely on a global alignment model, but use local maximisation so that each sentence pair can be processed independently. Various criteria may be used like Dice coefficient, cosine, mutual information [5] or likelihood ratio [3,16].

Sampling-based multilingual alignment, introduced in [13], and implemented as Anymalign[3], is an associative method for the computation of word associations. The method repeatedly draws random (mainly small) sub-corpora from the parallel corpus and obtains occurrence distributions of word pairs (or short word sequence pairs) within each sub-corpus so as to ultimately produce a word association table.

Bilingual hierarchical sub-sentential alignment, introduced in [12], and implemented as Cutnalign[4], is an associative method to compute sub-sentential alignments. It processes parallel sentences using a recursive binary segmentation of the alignment matrix. It yields performance comparable with that of state-of-the-art methods [7].

Figure 1 describes the training process which combines these two associative methods: it replaces GIZA++ and the grow-diag-final-and heuristic: Cutnalign uses word associations produced by Anymalign as input, and outputs sub-sentential alignments. The relevant script in Moses[5] then extracts phrases from sub-sentential alignments.

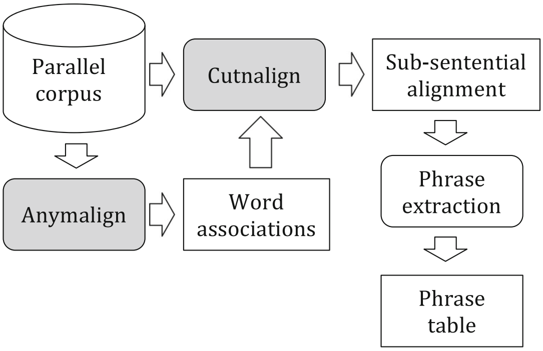


Fig. 1. Combination of two associative methods, Anymalign and Cutnalign, to obtain phrase tables from a parallel corpus

We present various types of improvements in the current implementations of the two above-mentioned associative methods that make them competitive with recent probabilistic approaches.

- improvement of the engineering type: we exploit the essence of associative methods and introduce multi-processing in both sampling-based alignment

[3] https://anymalign.limsi.fr/.

[4] Thanks to the authors for providing the source code.

[5] `train-model.perl --first step 4` .

and hierarchical sub-sentential alignment so as to trivially accelerate the overall alignment process. To compare with baseline systems, which are implemented in C/C++, in a more fair way, we also reimplement the core component of Cutnalign in C;

- improvement of the modelling type: we propose approximations to accelerate some decisions and a method to reduce the search space in hierarchical sub-sentential alignment so that additional speed-ups are obtained.

The rest of this paper is organized as follows: Sect. 2 rapidly describes the engineering improvements obtained by eliminating unnecessary computations and by introducing multi-processing. Section 3 explains and justifies in practice three practical and empirical improvements in hierarchical sub-sentential alignment. Section 4 gives an incremental evaluation of our work and a comparison with state-of-the-art methods on a series of machine translation experiments.

2 Acceleration Thanks to Re-engineering

2.1 Elimination of Unnecessary Computations

The first implementation of Cutnalign made use of a different matrix than the sentence pair matrix to accelerate computation. In that matrix, each cell contains the sum of all translation strengths of the block starting in the top left corner of the sentence pair matrix up to that cell, i.e., each cell (i, j) contains $W(X_{0\dots i}, Y_{0\dots j})$. As the process is recursive, the computation of W for many blocks inside the matrix is required. For a block $(X_{i_1\dots i_2}, Y_{j_1\dots j_2})$ extending from i_1 to i_2 and from j_1 to j_2, its corresponding $W(X_{i_1\dots i_2}, Y_{j_1\dots j_2})$ is easily computed as $W(X_{0\dots i_2}, Y_{0\dots j_2}) - W(X_{0\dots i_1}, Y_{0\dots j_1})$, thus saving computation time.

For generalisation purposes and readability of coding, the code introduced the factor $W(X_{0\dots 0}, Y_{0\dots 0})$ to be subtracted when computing W. This was hidden in one elegant general case by the use of general indices, that unfortunately could take the value 0 at some point during computation. As $W(X_{0\dots 0}, Y_{0\dots 0})$ is equal to 0, unnecessary subtractions by 0 were performed.

We isolated and rewrote nine different sub-cases (to the detriment of the aesthetics of the code) so as to eliminate such unnecessary subtractions. Much to our surprise, this led to an acceleration by a factor of approximately 40 times. The line labelled S in Table 2 reports such improvements.

2.2 Multi-processing

With the ubiquity of multi-processor systems, any software tool should allow optimal use of computer resources whenever possible. Associative methods make it possible by construction.

Word Associations. Anymalign draws random sub-corpora from the training corpus, and computes the occurrence distribution profiles for all words over all sentence pairs in each sub-corpus. Consequently, the process for each sub-corpus is independent. Thanks to this characteristic, no data needs to be transferred for synchronisation, thus avoiding any time consumption overhead usually observed when I/O operations are extensively used. The sizes of the sub-corpora are randomly drawn according to a specific distribution. Consequently, sampling of sizes can also be performed independently in different sub-processes, without affecting the general behavior in any way. Multi-processing is thus done by having each sub-process randomly drawing sub-corpora sizes, drawing sub-corpora of the given sizes, and computing word associations. After the master process has received an interruption[6], word associations and their associated frequencies are output by each sub-process and passed over to the master process which sums up the frequencies of each word association produced by each sub-process and computes association scores.

Experiments show that only very small, and insignificant differences in associations output exist between the mono-processing and multi-processing versions. They are due to differences in sampling.

Table 1 gives the BLEU scores obtained when allotting 15 min to Anymalign on one core, and 5 min on four cores. No significant difference is observed.

Table 1. No significant difference in BLEU scores is observed when using the mono-processor (original) or the multi-processor (M) versions of Anymalign to compute word associations. The number of word associations differs by 11% = (575,641 − 517,274)/517,274. The data used are described in Sect. 4.1.

Anymalign	Time (min)	# of word assoc.	BLEU (%)
Original	15	517,274	34.0 ± 0.8
M	5	575,641	34.1 ± 0.8

Hierarchical Sub-sentential Alignment. Cutnalign is easily parallelised by observing that the sub-sentential alignment process for each different sentence pair is independent from the other ones. The first two lines in Table 2 show the times obtained on increasing amounts of sentence pairs using the same word associations output by Anymalign. Using 4 cores divides the time by 3. Experiments have shown that using 4 cores divides the time by 3.

By design, introducing multi-processing as described above does not affect the quality of the final results, because the parallelised and non-parallelised implementations are theoretically equivalent. We checked that sub-sentential alignments outputs in both implementations are exactly the same.

[6] Anymalign is an anytime process, and should be given a timeout.

Table 2. Times (in minutes) for different versions of Cutnalign and different numbers of sentence pairs. The numbers in parentheses give the speed-up. The first line is the original implementation. M means a multi-processor version (see Sect. 2.2, 4 cores here). S means a version avoiding unnecessary subtractions of zeros (see Sect. 2.1). Same data as for Table 1

# of sentence pairs	2,500	5,000	7,500	10,000
Original	62 (×1)	141 (×1)	212 (×1)	288 (×1)
M	19 (× 3)	45 (×3)	64 (×3)	95 (×3)
S	1.5 (×41)	4 (×35)	5 (×42)	7 (×41)
M+S	0.5 (×124)	1 (×141)	1.5 (×141)	2 (×144)

3 Two Approximations in Hierarchical Sub-sentential Alignment

The original sub-sentential alignment method proposed in [12] can be explained in three main steps.

First, it builds a sentence pair matrix for a given sentence pair where the translation strength between a source word s and a target word t is computed as the product of the two association scores $p(s|t)$ and $p(t|s)$. In their proposal, as well as in this paper, the association scores are computed by Anymalign. Figure 2 illustrates such a sentence pair matrix. Notice that the content of the cells in the sentence pair matrix is bidirectional by construction.

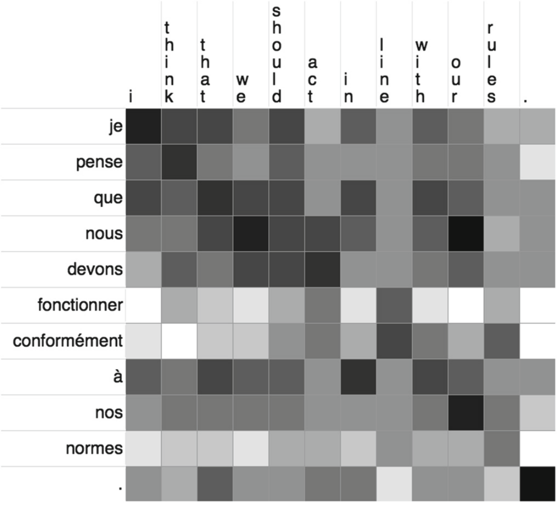


Fig. 2. Translation strengths in a French–English sentence pair matrix. Cells are grayed from 0.0 (white) to 1.0 (black) on a logarithmic scale.

Then, the method searches for the best alignment by computing the best segmentation of the sentences into sub-blocks recursively. This is done by computing the optimal bi-clustering of a bipartite graph, as suggested in the work of [17] for document clustering. For this purpose, a score named cut (see Eq. 1) is computed that sums up all the cells in the two sub-blocks of a block in the sentence pair matrix (see Fig. 3). In the definition of cut, $W(X,Y)$ is the sum of all translation strengths between all source and target words inside a sub-block (X,Y).

$$\mathrm{cut}(X,Y) = W(X,\overline{Y}) + W(\overline{X},Y) \tag{1}$$

In order to make the partition as dense as possible, [17] use a normalized variant of the score named Ncut (see Eq. 2)[7]. The best segmentation minimizes this variant over all $\mathrm{Ncut}(X,Y)$ and $\mathrm{Ncut}(\bar{X},Y)$, thus making simultaneously the decision of where to split and in which direction.

$$\begin{aligned}\mathrm{Ncut}(X,Y) = {} & \frac{\mathrm{cut}(X,Y)}{\mathrm{cut}(X,Y)+2\times W(X,Y)} \\ & + \frac{cut(\bar{X},\bar{Y})}{\mathrm{cut}(\bar{X},\bar{Y})+2\times W(\bar{X},\bar{Y})}\end{aligned} \tag{2}$$

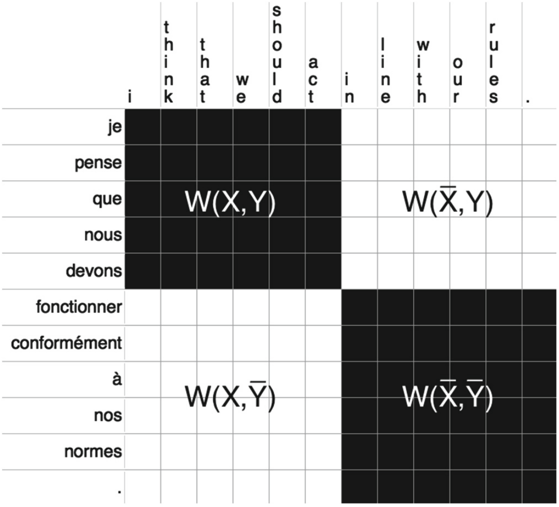


Fig. 3. Illustration of the segmentation of sentences $S = X.\overline{X}$ and $T = Y.\overline{Y}$. Here the block we start with is the entire matrix. Splitting horizontally and vertically into two parts gives four sub-blocks. There are two possible directions of segmentation: *linear* with the two sub-blocks in black, or *cross* with the two sub-blocks in white. The process is repeated recursively in the selected direction.

[7] Notice that, by definition: $\mathrm{Ncut}(X,Y) = \mathrm{Ncut}(\bar{X},\bar{Y})$ and $\mathrm{Ncut}(X,\bar{Y}) = \mathrm{Ncut}(\bar{X},Y)$. The same holds for cut.

Finally, as the method recursively segments the matrix, an alignment between the pair of sentences is obtained when no block remains to segment.

3.1 Decision on the Direction First

When splitting a block inside the sentence pair matrix into two sub-blocks, the segmentation method makes two theoretically separate decisions:

- which location, i.e., where to split, e.g., after (devons, act) in Fig. 3, and
- which direction, linear or cross, i.e., choosing either the black segmentation or the white one in Fig. 3.

The original approach consists in making the two decisions simultaneously, by selecting the max over all possible $\text{Ncut}(X, Y)$ and $\text{Ncut}(\bar{X}, Y)$. For a block of size $N \times M$, there are $2 \times N \times M$ Ncuts to compute. The original implementation of Cutnalign adopts this approach.

Our approach will separate the two decisions. We will first decide the direction and then the location. In practice, the use of cut instead of Ncut allows to make the decision on the direction without much difference in the final segmentation result. This leads to a reduction in computation because the computation of Ncut requires the computation of cut: making the decision in advance on cut avoids the computation of Ncut for the other direction. In this way, only half of the Ncuts, i.e., $N \times M$, are computed. As only one location inside a block is selected afterwards, possibly incorrect decisions on directions do remain unseen, and the final segmentation is not affected by them.

Table 3 reports the ratio of difference in final segmentation between the original approach and our approach on 350,000 French-English sentence pairs. It is only 0.3% in total. Differences start to appear only after the third level of segmentation and occur only once in 10,000 cases on that level. These figures show that the use of cut, instead of Ncut, for the decision on the direction does not significantly affect the final segmentation results. cut is enough to determine direction, while the introduction of its variant, Ncut, is justified to select the most dense pairs of sub-blocks, so as to lead to better alignments. As for time, a reduction of around 1/3 of computation time is observed. This will also be visible in Table 6 where the versions of Cutnalign denoted A use cut instead of Ncut for the decision on direction.

Figure 2 visualised a sentence pair matrix before sub-sentential alignment. Following intuition, the higher the translation strength between two words, the more they are prone to participate in the final sub-sentential alignment. Figure 4 shows this. Experiments on 347,614 French–English sentence pair matrices. Showed, that, in average, in each sentence, less than 3% of the word pairs have a translation strength higher than 0.1. More than 75% of these word pairs belong to the final sub-sentential alignment. We will now exploit this trend to reduce the search space in a sentence pair matrix.

Table 3. Percentage of segmentation differences when using cut instead of Ncut to decide for direction, on 347,614 French–English sentence pairs. The first column is the level of segmentation. The two middle columns give the average length of sub-blocks in source and target at that level. The last column gives the percentage of differences in the final segmentation when using approximate calculation; the reference is the final segmentation obtained using the original method.

Segmentation level	Avg length of block		Differences (%)
	in source	in target (in words)	
0	31	28	0.00
1	21	19	0.00
2	15	14	0.00
3	11	10	0.01
4	8	7	0.02
5	7	6	0.04
6	6	5	0.07
7	5	5	0.09
>7	4	3	0.11

3.2 Reduction of the Search Space

So as to decide the direction and the location for splitting into two sub-blocks, cuts are computed at *each* point inside a block. We propose to compute a kind of mask on the sentence pair matrix so as to restrict in advance the choice for splitting points at any level during segmentation.

In a preprocessing phase, all cells with a translation strength higher than a threshold are identified. We call them *peak cells*. As an illustration, consider the 5 black cells with a translation strength higher than 0.1 in the top matrix on the left of Fig. 5 (same as Fig. 2): (je, i), (nous, we), (nous, our), (nos, our) and (., .).

The next phase, the reduction phase, processes the matrix step by step. At the beginning of the first step, the domain is the entire sentence pair matrix and the search space is empty. In each step, the following operations are performed.

Firstly, the smallest rectangle with the largest number of peak cells is determined. The reason to select such a rectangle follows the intuition behind the introduction of Ncut: by approximation, the smallest rectangle with the largest number of peak cells should lead to the extraction of the densest sub-blocks. Necessarily, such a rectangle is delimited by some peak cells, which are added to the search space. The top matrix on the right in Fig. 5 shows the rectangle obtained in the first iteration step. It is the outer rectangle visualized by dotted lines. It is delimited by the peak cells (je, i) and (., .). The bottom matrix shows the one obtained in the second iteration step (outer rectangle again). It is delimited by the peak cells: (nous, we), (nous, our) and (nos, our). In all generality, peak cells do not necessarily lie in the corners; Fig. 5 is a particular case.

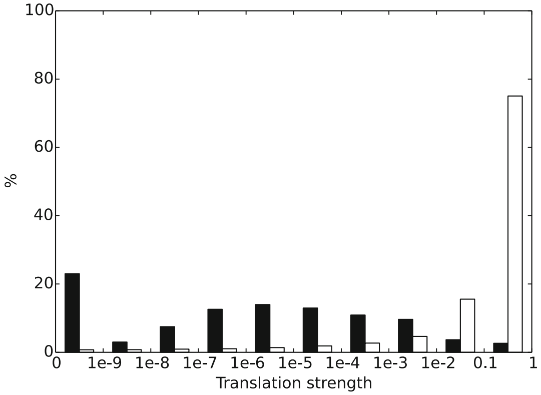


Fig. 4. Percentage of cells in the matrix with a translation strength inside the bin (*black bars*) and percentage of word pairs actually participating in the final sub-sentential alignment (*white bars*). The bars are drawn for each logarithmic bin on the horizontal axis: $[0, 10^{-9})$, $[10^{-9}, 10^{-8})$, ... $[0.1, 1]$.

Then, the next domain for the next step of iteration is determined by leaving out the cells in the contour which are not peak cells. In the two matrices on the right of Fig. 5, such new domains are the *inner* rectangles delimited by dotted lines. This leaves out the cells containing a grey cross in the figure. This can be done with some confidence because, by construction, sub-blocks extracted from such locations will leave out many well aligned word pairs and cannot be expected to yield a promising Ncut. On the contrary, one can expect that sub-blocks determined by splitting on positions in the new domain will be denser in well aligned word pairs.

Finally, the corner regions between two successive smallest rectangles are added to the search space (see bottom left matrix in Fig. 5), because the positions inside these regions have a good chance to provide a higher number of well aligned word pairs when splitting into sub-blocks.

The new domain is passed to the next iteration step. The iteration process stops when the smallest rectangle contains one or zero peak cell. In this case, the search space is not reduced and used as is.

The final reduced search space is thus made out of all the peak cells, all the corner regions between two successive smallest rectangles and the last inner rectangle. It usually takes the rough shape of a cross extending over the sentence pair matrix. This reduced search space is then passed over to the general sub-sentential alignment process which will no more be allowed to consider any possible positions to split into sub-blocks, but will be confined to the positions in the reduced search space at any level. As a consequence, processing time will be reduced: for well-balanced cases, a reduction from a computation in $O(n^2)$ to $O(n \log n)$ is obtained. The experiments reported hereafter also show that the reduction in search space does not affect BLEU scores.

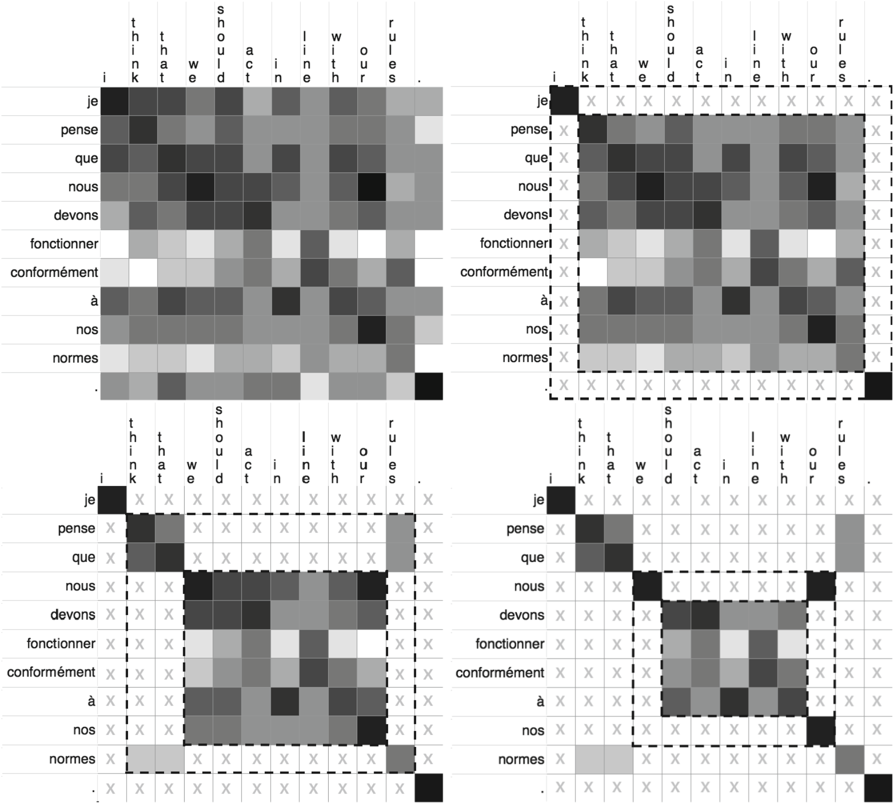


Fig. 5. Reduction of the search space in a French–English sentence pair matrix (same as Fig. 2) to find the sub-sentential alignment. Cells are grayed from 0.0 (white) to 1.0 (black) on a logarithmic scale to visualise translation strengths.

The procedure described above for reduction of space was first implemented in Python. Its re-implementation in C divides the processing time by 6 (see Table 6, last line).

4 Experiments

4.1 Overview: Tools and Data

We evaluate our work by building PB-SMT systems using the Moses toolkit [11], lexicalised reordering models [10] and the KenLM Language Modelling toolkit [8]. Accuracy relatively to translation references is assessed using BLEU. Baseline systems are built using GIZA++ or Fast align. For systems built using Anymalign and Cutnalign, the overall process to build translation tables has been illustrated in Fig. 1.

All the experiments mentioned in this paper use data from corresponding parts of the Europarl parallel corpus v3 [9], so that BLEU scores can be compared across language pairs, as the training, tuning and test sets correspond across languages. The training corpus is made of 347,614 sentences; 500 sentences are used for tuning; the test set contains 5,000 lines.

We use 3 language pairs in both directions involving 5 European languages: English (en), French (fr), Spanish (es), Portuguese (pt), Finnish (fi): fr–en as usual test languages, fi–en, i.e., agglutinative language vs. isolating language, and es–pt, as an example of close languages. Statistics on the data are given in Table 4.

Table 4. Statistics on the data used (M = million)

		en	fr	es	pt	fi
Train	Sentences			347,614		
	Word tokens	10.01M	11.62M	10,52M	10.35M	7.21M
	Word types	57,728	72,042	124,035	116,165	289,054
Tune	Sentences			500		
	Word tokens	14,697	17,132	15,440	15,348	10,580
	Word types	2,929	3,395	3,489	3,595	4,576
Test	Sentences			5,000		

4.2 Timeout and Max Length of Phrases for Anymalign

As Anymalign should be given a timeout, we first inquire the relationship between the timeout and translation accuracy. In addition, we also inquire the influence of the max length of the phrases output by Anymalign on translation accuracy.

Table 5 shows that 15 min is enough to get comparable translation accuracy. For all language pairs and timeouts, almost all the best scores are obtained for a max length of phrases set to 2. Drawing from these results, the experiments reported hereafter will adopt the following settings: Anymalign will be run with a max length of phrases set to 2. This will be denoted by Anymalign -i2. The original version of Anymalign will be given a time-out of 15 min, while its multi-processor version will be given a time-out of 5 min, according to the acceleration reported in Sect. 2.2.

4.3 Incremental Improvements

We incrementally evaluated the improvements presented in the previous sections on French–English data. In order to evaluate the difference in the final sub-sentential alignments obtained, we measured the alignment error rate (AER) [1] by reference to the results obtained using the original methods without any

Table 5. Translation accuracy (BLEU) against timeout and maximal length of phrases output by Anymalign for three language pairs. Here the test set contains 38,000 lines.

Language pair	Max length of phrases	BLEU for $\neq$ timeouts			
		15 min	30 min	60 min	120 min
fr–en	1	33.9	34.1	34.0	34.1
	2	34.0	34.3	34.1	34.2
	3	34.0	33.7	34.4	34.3
	4	33.7	33.8	34.2	34.1
pt–es	1	38.5	38.6	38.6	38.6
	2	38.5	38.7	38.7	38.8
	3	38.5	38.4	38.8	38.9
	4	38.4	38.5	38.6	38.8
fi–en	1	23.0	23.3	23.6	23.8
	2	23.3	23.2	23.8	24.2
	3	22.8	23.4	23.5	23.7
	4	22.7	23.2	23.6	23.9

improvement. As seen in Table 6, the multi-processing implementation of Anymalign delivers more word alignments than the mono-processor implementation in 3 times less time. In total, we could divide the training time by 750 without affecting the BLEU scores. Differences in alignements are observed but positively impact the results.

4.4 Comparison with Fast align

We compare the integration of all improvements with the fastest probabilistic state-of-the-art alignment method: Fast align. We run it with default settings in two directions, source to target and target to source, to produce alignments from which a phrase table is extracted using the grow-diag-final-and heuristic. For Anymalign, we use the options -i 2 -t 300, i.e., we set a preferred length of up to 2 words in associations, and a timeout of 5 min.

The results of the experiments are presented in Table 7. Our improvements allow the associative methods to beat Fast align in time. In addition, as much smaller phrase tables are extracted by our method, lower times for decoding are observed. Alignments produced with our improvements yield slightly lower scores than those obtained with Fast align on French–English and Spanish–Portuguese in both directions, but with no statistically significant difference in each case as the confidence intervals show. Unfortunately, on Finnish–English, in both directions, our BLEU scores are significantly lower. This may come from an insufficient timeout for Anymalign, 5 min, chosen for the sake of consistency across all experiments reported in this section.

Table 6. Incremental gains in time on French–English data. The max length of phrases output by Anymalign is set to 2 in all experiments. M denotes a multi-processing version (number of cores used: 4). For Cutnalign, S avoids unnecessary subtraction of zeros; A uses cut instead of Ncut to make the decision on direction of segmentation (Sect. 3.1); R implements reduction of search space (Sect. 3.2, threshold for translation strength set to 0.5); C uses re-implementation in C of core component of Cutnalign. The alignment time is the time for Anymalign plus the time for Cutnalign. In total a speed-up by 750 has been obtained (4,515/6).

Anymalign -i2 + Cutnalign	Alignment time (min)	AER (%)	BLEU (%)
original + original	15 + 4,500	–	34.0 ± 0.8
original + M	15 + 1,594	0.0	34.0 ± 0.8
original + M+S	15 + 31	0.0	34.0 ± 0.8
M + M+S	5 + 32	0.0	34.1 ± 0.8
M + M+S+A	5 + 19	5.4	34.2 ± 0.8
M + M+S+R	5 + 15	8.8	34.0 ± 0.8
M + M+S+A+R	5 + 6	11.8	34.1 ± 0.8
M + M+S+A+R+C	5 + 1	11.8	34.1 ± 0.8

Table 7. Comparison of BLEU scores and alignment times in 6 language pairs with different aligners

Language pair	Aligner	Align. time (min)	BLEU (%)
pt-es	MGIZA++	150	39.1 ± 0.8
	Fast align	17	38.9 ± 0.8
	M + M+S+A+R+C	**7**	38.8 ± 0.8
es-pt	MGIZA++	140	37.1 ± 0.8
	Fast align	17	36.9 ± 0.8
	M + M+S+A+R+C	**7**	36.6 ± 0.8
en-fr	MGIZA++	150	36.3 ± 0.7
	Fast align	17	36.1 ± 0.7
	M + M+S+A+R+C	**7**	36.0 ± 0.7
fr-en	MGIZA++	170	34.5 ± 0.8
	Fast align	17	34.5 ± 0.8
	M + M+S+A+R+C	**7**	34.1 ± 0.8
fi-en	MGIZA++	120	26.1 ± 0.8
	Fast align	14	25.0 ± 0.8
	M + M+S+A+R+C	**6**	23.9 ± 0.8
en-fi	MGIZA++	110	16.3 ± 0.8
	Fast align	14	16.7 ± 0.8
	M + M+S+A+R+C	**6**	15.7 ± 0.8

5 Conclusion

We presented multi-processing implementations of the multilingual sampling-based alignment method [13] and of the hierarchical sub-sentential alignment method [12], two associative methods which, by essence allow for this. We introduced two approximations in the hierarchical sub-sentential alignment method: we modified how to decide the direction of split and we reduced the search space. The removal of some unnecessary computations, and the re-implementation of core components in C were also introduced. We obtained considerable gains in time so that the combination of these two associative methods becomes competitive with probabilistic methods.

The new multi-processing version of Anymalign divides the computation times for word associations by 3 on a 4-core computer. Elimination of unnecessary computations of special cases in Cutnalign divides the computation times of sub-sentential alignments by more than 40 times in comparison with the original implementation described in [12]. Combined with multi-processing, this leads to a speed-up of roughly 140 times on a 4-core computer. The latest version of Cutnalign, which also includes approximations in decisions and reduction of the search space, and C implementation of core components runs approximately 4,500 times faster than the original implementation. The combination of the two new versions of Anymalign and Cutnalign result in an overall alignment process that can be faster than Fast align while delivering comparable results.

As for comparison with probabilistic methods, some may argue that the comparison of a probabilistic method running on one processor with an associative method running on 4 cores is unfair. We claim on the contrary that it *is* fair because associative methods intrinsically cater for this at no expense of the quality of their results. What would be unfair is precisely to forbid associative methods to make use of their inherent advantages.

Our main task in the near future is to get rid of the threshold manually set for the reduction of the search space in the hierarchical sub-sentential alignment method. We want to find a way to automatically determine a threshold based on an inspection of the distribution of translation strengths in sentence pair matrices.

As a final note, it is worth mentioning that both Anymalign and Cutnalign are by essence bidirectional methods. They compute the bidirectional parameters or the sub-sentential alignments in one pass, on the contrary to MGIZA++ or Fast align which have to be run in both directions and on the contrary to the first step in grow-dial-final-and which builds two matrices in both directions before merging.

References

1. Ayan, N.F., Dorr, B.J.: Going beyond AER: an extensive analysis of word alignments and their impact on MT. In: Proceedings of COLING/ACL, pp. 9–16 (2006)
2. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2), 263–311 (1993)
3. Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Comput. Linguist. **19**(1), 61–74 (1993)
4. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of HLT-NAACL, pp. 644–648 (2013)
5. Gale, W.A., Church, K.W.: Identifying word correspondences in parallel texts. In: Proceedings of the Workshop on Speech and Natural Language, vol. 91, pp. 152–157 (1991)
6. Gao, Q., Vogel, S.: Parallel implementations of word alignment tool. In: Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pp. 49–57 (2008)
7. Gong, L., Max, A., Yvon, F.: Improving bilingual sub-sentential alignment by sampling-based transpotting. In: Proceedings of IWSLT, pp. 243–250 (2013)
8. Heafield, K.: Kenlm: faster and smaller language model queries. In: Proceedings of the 6th Workshop on Statistical Machine Translation, pp. 187–197 (2011)
9. Koehn, P.: Europarl: a parallel corpus for statistical machine translation. In: Proceedings of Machine Translation Summit, vol. 5, pp. 79–86 (2005)
10. Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D., White, M.: Edinburgh system description for the 2005 IWSLT speech translation evaluation. In: Proceedings of IWSLT, pp. 68–75 (2005)
11. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R.: Moses: open source toolkit for statistical machine translation. In: Proceedings of ACL (Poster sessions), pp. 177–180 (2007)
12. Lardilleux, A., Yvon, F., Lepage, Y.: Hierarchical sub-sentential alignment with Anymalign. In: Proceedings of EAMT 2012, pp. 279–286 (2012)
13. Lardilleux, A., Yvon, F., Lepage, Y.: Generalizing sampling-based multilingual alignment. Mach. Transl. **27**(1), 1–23 (2013)
14. Levenberg, A., Callison-Burch, C., Osborne, M.: Stream-based translation models for statistical machine translation. In: Proceedings of HLT-NAACL, pp. 394–402 (2010)
15. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
16. Smaïli, K., Jamoussi, S., Langlois, D., Haton, J.P.: Statistical feature language model. In: Proceedings of ICSLP, pp. 1357–1360 (2004)
17. Zha, H., He, X., Ding, C., Simon, H., Gu, M.: Bipartite graph partitioning and data clustering. In: Proceedings of International Conference on Information and Knowledge Management, pp. 25–32 (2001)

Information and Data Extraction

Events Extractor for Polish Based on Semantics-Driven Extraction Templates

Jolanta Cybulka(✉) and Jakub Dutkiewicz

Poznań University of Technology, 3B Piotrowo Street, Poznań, Poland
{jolanta.cybulka, jakub.dutkiewicz}@put.poznan.pl

Abstract. The paper presents a certain paradigm of extracting events from Polish free texts. We call it *semantics-driven* because the extraction templates are generated from the specification of a domain knowledge that is expressed in the form of a well-founded ontology. The considered method is equipped with the supporting tool that has two components: the first one is domain-dependent and serves to generate extraction templates on the basis of an ontology. The second part is linguistic and domain-independent and may be used whenever templates are supplied, not necessarily via the generator. We checked the quality performance of our generator on a basis of a case study.

Keywords: Event extraction · Well-founded ontology
Semantics-driven event extraction for Polish

1 Extraction of Events from Free Texts

The Web is populated with enormous amounts of free-text repositories in which the valuable factual information is hidden. This hidden knowledge should be precisely automatically revealed in order to fit the different user's information needs, for example Machine Translation, Question Answering, Text Summarization etc. (see [9]). Such a task is known as Information Extraction (IE) and, in general, relies on obtaining instances of predefined types of entities (including relationships and events) of some domain of interest from free-texts. Apart from entities themselves, their arguments or characteristics are also extracted. The obtained target instances are structured (in some knowledge representation format) as the opposite to unstructured source natural language texts. The process of IE implies the necessity to consider at least two issues: on the one hand it is the representation of domain's semantics (i.e. the predefined types of entities and their roles in knowledge representation) and on the other hand - the identification of possible syntactic representations of entities in unstructured texts. Both issues have their specificities but the second is particularly difficult to address, especially if we deal with highly inflected languages such as Polish. Also, the entity description may not appear in only one sentence but in many sentences and this brings about further difficulties in natural language analysis.

Event extraction (EE) is a kind of information extraction (IE) and relies on obtaining from free-texts a predefined types of facts concerning events: we aim to detect what happened and what were the arguments/parameters of it. In such a case the extraction is domain-dependent because we need the explicit representation of

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 231–245, 2018.
https://doi.org/10.1007/978-3-319-93782-3_17

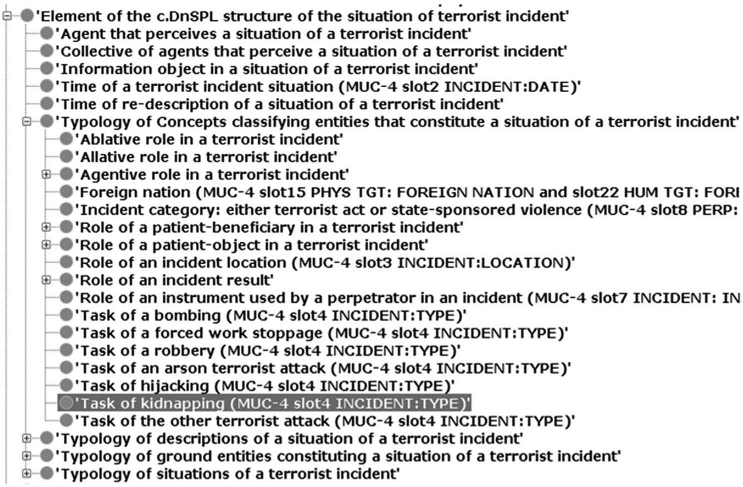
knowledge concerning events in some domain of interest. Once the domain semantics is established one may create event extraction templates. In our work we do it automatically generating a knowledge frame and extraction templates on the basis of a well-founded ontology (see Sect. 2). The generated templates are parsed by a linguistic, domain-independent part of the extractor. We briefly describe the extraction process and introduce the basic data structures used in this process (see Sect. 3). We evaluate the extraction providing a case study in Sect. 4.

2 Specifying Reader's Knowledge and Generation of Event Extraction Templates

The semantics-driven extraction of events from free-texts requires having a domain knowledge to be explicitly specified. Additionally, such a specification should reflect semantic features of words and phrases that express events and their parameters. One may try to use directly a general purpose semantic lexicon of verbs, like VerbNet [7], which specifies non-contextual, "objective" world knowledge, but sometimes contextual, "subjective" knowledge of a text reader is of better value. In such cases the suitable domain ontology is useful, that contains the specification of semantics of syntactical predicate-arguments structure. Indirectly, the ontological concepts may be linked to lexicalizations provided by language resources, for example these that are contained in LLOD (Linguistic Linked Open Data, [6]). To us the requirements concerning the specification of a (contextual) domain semantics may be ideally fulfilled by using a well-founded ontology. We use a (bilingual, Polish-English) capsular c.DnSPL ontology [1] which is based on the foundational ontological pattern of *constructive descriptions and situations* of [4].

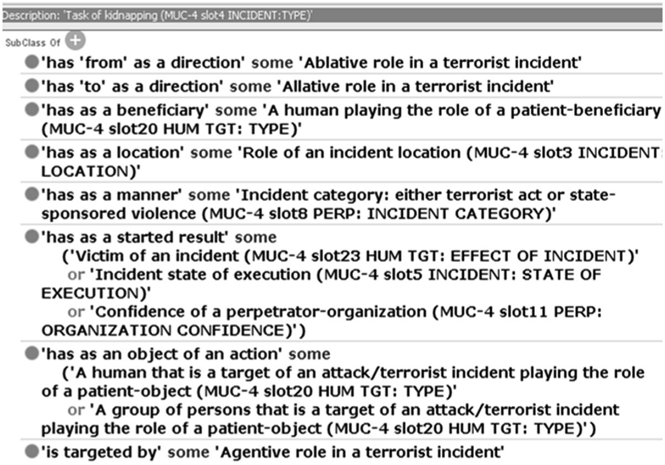
A capsule specializes (in terms of subsumption) the foundational pattern representing a "situation" that forms a conceptual equivalent of some domain. The capsule skeleton is universal and it has 9 components, from which we describe here only those that are necessary to understand our ideas. For example, let us consider a domain of terrorist incidents as it is understood in MUC-4 ([10]) knowledge frame. To represent it we created a capsule (Fig. 1) named *c.DnSPL relation of the situation of a terrorist incident according to MUC-3 and MUC-4* and two typologies of ontological entities: *Typology of ground entities* constituting a situation of a terrorist incident and their equivalent *Typology of Concepts classifying entities that constitute a situation of a terrorist inciden*t. The latter are results of perceiving of the former by some *Agent that perceives a situation of a terrorist incident*. Among the *Typology of Concepts classifying…* the *perdurants* called tasks are of special interest because they represent events. Let us consider the ontological nature of a kidnapping event, reified as a *Task of kidnapping* (*MUC-4 slot4 INCIDENT:TYPE*).

To serve as a semantic basis for extracting such events from free-texts the *perdurant* should have parameters such as: a perpetrator, a victim, a location in time and place, and optionally a beneficiary, source and target places, a manner, a result etc. These parameters are modelled by using *thematic roles*, respectively a role of an: *agent*, *patient-object*, *location*, *patient-beneficiary*, *ablative location*, *allative location*, *manner*, *result* (see Fig. 2).



- 'Element of the c.DnSPL structure of the situation of terrorist incident'
  - 'Agent that perceives a situation of a terrorist incident'
  - 'Collective of agents that perceive a situation of a terrorist incident'
  - 'Information object in a situation of a terrorist incident'
  - 'Time of a terrorist incident situation (MUC-4 slot2 INCIDENT:DATE)'
  - 'Time of re-description of a situation of a terrorist incident'
  - 'Typology of Concepts classifying entities that constitute a situation of a terrorist incident'
    - 'Ablative role in a terrorist incident'
    - 'Allative role in a terrorist incident'
    - 'Agentive role in a terrorist incident'
    - 'Foreign nation (MUC-4 slot15 PHYS TGT: FOREIGN NATION and slot22 HUM TGT: FORI
    - 'Incident category: either terrorist act or state-sponsored violence (MUC-4 slot8 PERP:
    - 'Role of a patient-beneficiary in a terrorist incident'
    - 'Role of a patient-object in a terrorist incident'
    - 'Role of an incident location (MUC-4 slot3 INCIDENT:LOCATION)'
    - 'Role of an incident result'
    - 'Role of an instrument used by a perpetrator in an incident (MUC-4 slot7 INCIDENT: IN
    - 'Task of a bombing (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of a forced work stoppage (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of a robbery (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of an arson terrorist attack (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of hijacking (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of kidnapping (MUC-4 slot4 INCIDENT:TYPE)'
    - 'Task of the other terrorist attack (MUC-4 slot4 INCIDENT:TYPE)'
  - 'Typology of descriptions of a situation of a terrorist incident'
  - 'Typology of ground entities constituting a situation of a terrorist incident'
  - 'Typology of situations of a terrorist incident'

Fig. 1. The exemplary capsule of terrorist incidents (as seen in Protégé editor).



Description: 'Task of kidnapping (MUC-4 slot4 INCIDENT:TYPE)'

SubClass Of

- 'has 'from' as a direction' some 'Ablative role in a terrorist incident'
- 'has 'to' as a direction' some 'Allative role in a terrorist incident'
- 'has as a beneficiary' some 'A human playing the role of a patient-beneficiary (MUC-4 slot20 HUM TGT: TYPE)'
- 'has as a location' some 'Role of an incident location (MUC-4 slot3 INCIDENT: LOCATION)'
- 'has as a manner' some 'Incident category: either terrorist act or state-sponsored violence (MUC-4 slot8 PERP: INCIDENT CATEGORY)'
- 'has as a started result' some ('Victim of an incident (MUC-4 slot23 HUM TGT: EFFECT OF INCIDENT)' or 'Incident state of execution (MUC-4 slot5 INCIDENT: STATE OF EXECUTION)' or 'Confidence of a perpetrator-organization (MUC-4 slot11 PERP: ORGANIZATION CONFIDENCE)')
- 'has as an object of an action' some ('A human that is a target of an attack/terrorist incident playing the role of a patient-object (MUC-4 slot20 HUM TGT: TYPE)' or 'A group of persons that is a target of an attack/terrorist incident playing the role of a patient-object (MUC-4 slot20 HUM TGT: TYPE)')
- 'is targeted by' some 'Agentive role in a terrorist incident'

Fig. 2. The (logical) description of *Task of kidnapping* concept.

The considered thematic roles are concepts used to classify the real domain entities represented as *Typology of ground entities …*. For example, a civilian may play the role of a victim-patient-object while a terrorist organization may play the role of a perpetrator-agent. As it was mentioned, all these concepts should be carefully lexicalized. Having such a knowledge scheme in mind one may expect that the considered ontological capsule specifies the semantics of the italicized part of a NJKP[1] sentence "*Krystynę Starczewską porwano z ulicy do pałacu Mostowskich*, pokazano nakaz

[1] NJKP − national corpus for Polish, http://nkjp.pl/index.php?page=0&lang=1.

ekstradycji narzeczonego córki i zaproponowano targ – współpraca za pozostawienie Nama w spokoju.", which means "*Krystyna Starczewska was kidnapped from the street to Mostowski Palace*, …". The verb *porwać/kidnap* is here in impersonal form, so there is no an instance classified as *Agentive role in a terrorist incident* but the proper name of a woman *Krystyna Starczewska* (in accusative; it may be an example of a concept *Human* that acts for an instance of *Civilian*) suggests she plays the *Role of a patient-object in a terrorist incident*, which is specialized by *A human that is a target of an attack ...* (*MUC-4 slot20 HUM TGT: TYPE*). Also, the prepositional phrases *z ulicy/from the street* and do *pałacu Mostowskich/to Mostowski Palace* are expressions of ablative and allative locations, respectively.

Consider then another NJKP sentence "Jeden z nich w rozmowie lekko odwrócił się od stołu, *a wtedy orzechówka porwała mu z talerzyka ogromny kawał kiełbasy i uciekła*." where the italicized part says: "*…the spotted nutcracker grabbed from his plate a big piece of sausage* …". In Polish, *porwać* is polysemous and also means *to grab*. It seems obvious that this fact is semantically inappropriate and should not be extracted due to unfitness of verb's parameters-roles: *spotted nutcracker* cannot be classified as *Agentive role in a terrorist incident* (i.e. person, organization etc.), *sausage* is not considered to be a kidnapped victim (i.e. person, people etc.), also *a plate* cannot be an ablative location for people. Thus the semantics specified in the ontology allows us to filter the inadequate facts. But, as it was said before, the semantics may be "dressed" in many syntactic forms. The question arises, how to provide them? To specify the possible syntactic expressions of the underlined semantics we use role approximations proposed in [5]. With the help of them we built equivalents between thematic roles and syntactic theta-roles (parameters of verbs in valence structures). In the first column of Table 1 we list 8 roles used in c.DnSPL ontology, to which we assign theta-roles, assuming that the verb is in active voice (we have also analogous equivalents in cases of passive voice, impersonal form and a gerund). Theta-roles are represented using a notation from Walenty[2], the Polish valence dictionary (we assume that the used abbreviations, as *subj*, *obj*, *np* and *prepnp* are widely known; the same concerns grammatical cases). The syntactic expression of the agentive role is simplified for we assume the noun phrase to be in nominative. The same concerns patient-object - it should be in accusative.

The input to the event extractor consists of two parts: a knowledge frame and a set of extraction templates. The main part of a knowledge frame is a set of pairs: a slot and a corresponding list of semantic types – concepts taken from the ontology. Thus, the frame abbreviates the semantics of an event. Let us look at the frame representing a kidnapping (see Fig. 3). The agentive role (*RolaAgentywnaWPrzebieguIncydentu*) forms the first slot and the semantic constraints imposed on role's syntactic realizations say that they should be: a terrorist organization (*OrganizacjaTerrorystyczna*) or a person (*Osoba*). The other slots correspond to the following roles: patient-object, patient-beneficiary, ablative, allative and locative. Every *extraction template* belongs to one of four groups, from which three are singled out according to the form of a verb (here verbs are the so-called *anchoring phrases*): we have then active (PL ACT),

[2] Walenty, http://zil.ipipan.waw.pl/Walenty.

Table 1. Thematic roles and their syntactic equivalents.

Thematic role	Theta-role
Agentive	subj{np(str/nom)}
Patient-beneficiary	obj{np(dat)}, {prepnp(dla,gen)}, {prepnp(wobec,gen)}, {prepnp (przeciw,dat)}
Patient-object	obj{np(str/acc)}
Instrumental	{np(inst)}
Allative	{prepnp(do,gen)},{prepnp(ku,dat)}, {prepnp(między,acc)}, {prepnp(na, acc)},{prepnp(nad,acc)}, {prepnp(pod,acc)},{prepnp(po,acc)}, {prepnp (pomiędzy,acc)},{prepnp(ponad,acc)}, {prepnp(poza,acc)},{prepnp(przed, acc)}, {prepnp(w,acc)}, {prepnp(za,acc)}
Ablative	{prepnp(dzięki,dat)}, {prepnp(od,gen)}, {prepnp(spod,gen)}, {prepnp (spośród,gen)}, {prepnp(wskutek,gen)},{prepnp(z,gen)}, {prepnp (zza,gen)}
Locative	{prepnp(koło,gen)}, {prepnp(poniżej,gen)}, {prepnp(wokół,gen)}, {prepnp (wsród,gen)}, {prepnp(u,gen)},{prepnp(między,inst)}, {prepnp(nad,inst)}, {prepnp(pod,inst)}, {prepnp(pomiędzy,inst)}, {prepnp(ponad,inst)}, {prepnp(przed,inst)}, {prepnp(za,inst)}, {prepnp(na,loc)}, {prepnp(po, loc)}, {prepnp(poza,loc)},{prepnp(przy,loc)}, {prepnp(w,loc)}
Perlative	{prepnp(bez,gen)}, {prepnp(poprzez,acc)}, {prepnp(przez,acc)},{prepnp(z, inst)}

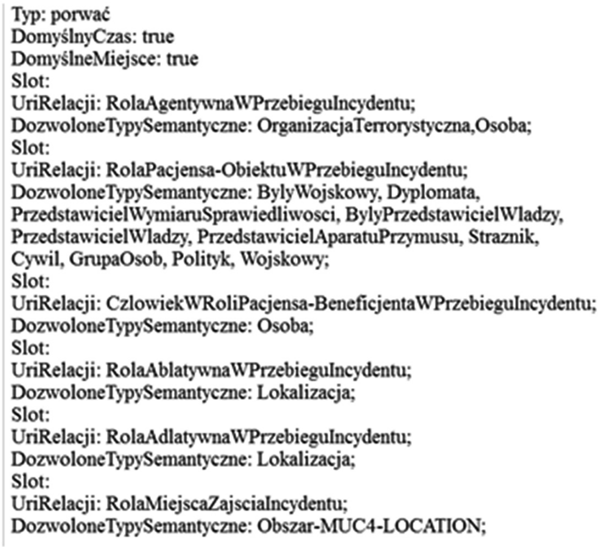
passive (PL PASS) and impersonal (PL IMPS) templates. The fourth type of templates is anchored by the gerund noun phrase (PL GERUND). Let us look at the kidnapping template anchored by the verb *porwać* in active voice (Fig. 4). We have a template *header* and a *list of elements* specifying syntactic parameters of the anchoring verb. The header starts with the template name (*Nazwa szablonu*), an associated knowledge frame (*Rama wiedzy*), a version of a template (*Wersja szablonu*), a specification of a verbal anchor and its voice (*Kotwica* and *Typ frazy*). Analyzing the ontological description of a *Task of kidnapping…* (Fig. 2) we see that the semantics of a kidnapping *perdurant* is specified by using 8 thematic roles, from which 6 are contained in Table 1. The roles are ontological concepts that are linked to the *perdurant* by carefully distinguished, formal foundational relations. For example, the considered event is linked to the *Agentive role in…* through the relation *is targeted by*. Following this relation the first *Element* was generated: the one connected with the *Agentive role in…* (*RolaAgentywnaWPrzebieguIncydentu*) that in free-texts is represented by the sentence subject (a noun phrase *NG* in *nominative* case (*Przypadki*) without prepositions (*Przyimki*)).

The other *Elements*, similarly to the knowledge frame, correspond to: patient-object, patient-beneficiary, allative, ablative and locative roles. The method of generation can be characterized by the following scheme:

Input: Ontology in OWL

Output: Knowledge frame, set of extraction templates

Method:



```
Typ: porwać
DomyślnyCzas: true
DomyślneMiejsce: true
Slot:
UriRelacji: RolaAgentywnaWPrzebieguIncydentu;
DozwoloneTypySemantyczne: OrganizacjaTerrorystyczna,Osoba;
Slot:
UriRelacji: RolaPacjensa-ObiektuWPrzebieguIncydentu;
DozwoloneTypySemantyczne: BylyWojskowy, Dyplomata,
PrzedstawicielWymiaruSprawiedliwosci, BylyPrzedstawicielWladzy,
PrzedstawicielWladzy, PrzedstawicielAparatuPrzymusu, Straznik,
Cywil, GrupaOsob, Polityk, Wojskowy;
Slot:
UriRelacji: CzlowiekWRoliPacjensa-BeneficjentaWPrzebieguIncydentu;
DozwoloneTypySemantyczne: Osoba;
Slot:
UriRelacji: RolaAblatywnaWPrzebieguIncydentu;
DozwoloneTypySemantyczne: Lokalizacja;
Slot:
UriRelacji: RolaAdlatywnaWPrzebieguIncydentu;
DozwoloneTypySemantyczne: Lokalizacja;
Slot:
UriRelacji: RolaMiejscaZajsciaIncydentu;
DozwoloneTypySemantyczne: Obszar-MUC4-LOCATION;
```

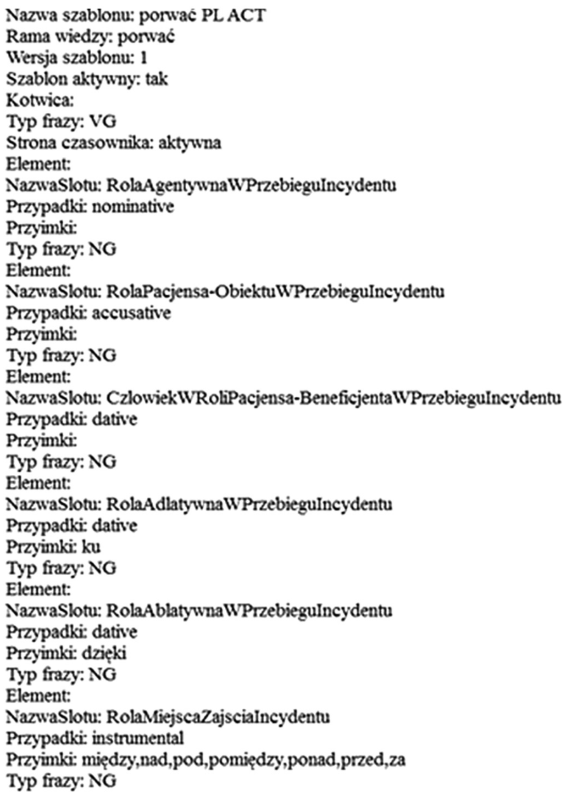
Fig. 3. The knowledge frame reflecting the kidnapping event (with the verb's lexicalization *porwać*).

1. Parse the ontology and check if it is a capsular c.DnSPL − if not, stop.
2. For each capsule do

   list its *perdurants* (i.e. tasks)

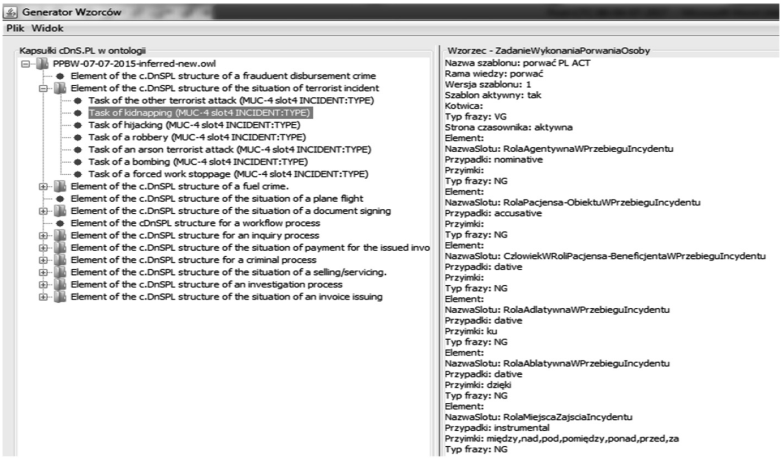
3. For a user chosen *perdurant* do
   (a) read-in a verb in infinitive
   (b) generate active voice templates
   (c) generate passive voice templates
   (d) generate impersonal form templates
   (e) generate a gerund template
   (f) generate knowledge frame.

Considering all variants of theta-roles, maximally, i.e. if all 8 roles are used in a task description, we have 162 templates for one verb form plus one GERUND template. In the description of kidnapping we have 6 roles and 163 templates in all. The left panel of the screenshot from Fig. 5 shows the capsules and *perdurants* of the parsed ontology, where the task of kidnapping is highlighted. The right panel contains the generated templates and a knowledge frame, after providing the lexicalization of the verb (here: *porwać*).



```
Nazwa szablonu: porwać PL ACT
Rama wiedzy: porwać
Wersja szablonu: 1
Szablon aktywny: tak
Kotwica:
Typ frazy: VG
Strona czasownika: aktywna
Element:
NazwaSlotu: RolaAgentywnaWPrzebieguIncydentu
Przypadki: nominative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: RolaPacjensa-ObiektuWPrzebieguIncydentu
Przypadki: accusative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: CzlowiekWRoliPacjensa-BeneficjentaWPrzebieguIncydentu
Przypadki: dative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: RolaAdlatywnaWPrzebieguIncydentu
Przypadki: dative
Przyimki: ku
Typ frazy: NG
Element:
NazwaSlotu: RolaAblatywnaWPrzebieguIncydentu
Przypadki: dative
Przyimki: dzięki
Typ frazy: NG
Element:
NazwaSlotu: RolaMiejscaZajsciaIncydentu
Przypadki: instrumental
Przyimki: między,nad,pod,pomiędzy,ponad,przed,za
Typ frazy: NG
```

Fig. 4. The exemplary template to extract the kidnapping event (with the possible verb's lexicalization *porwać*).



```
PPBW-07-07-2015-inferred-new.owl
  Element of the c.DnSPL structure of a frauduent disbursement crime
  Element of the c.DnSPL structure of the situation of terrorist incident
    Task of the other terrorist attack (MUC-4 slot4 INCIDENT:TYPE)
    Task of kidnapping (MUC-4 slot4 INCIDENT:TYPE)
    Task of hijacking (MUC-4 slot4 INCIDENT:TYPE)
    Task of a robbery (MUC-4 slot4 INCIDENT:TYPE)
    Task of an arson terrorist attack (MUC-4 slot4 INCIDENT:TYPE)
    Task of a bombing (MUC-4 slot4 INCIDENT:TYPE)
    Task of a forced work stoppage (MUC-4 slot4 INCIDENT:TYPE)
  Element of the c.DnSPL structure of a fuel crime.
  Element of the c.DnSPL structure of the situation of a plane flight
  Element of the c.DnSPL structure of the situation of a document signing
  Element of the cDnSPL structure for a workflow process
  Element of the c.DnSPL structure for an inquiry process
  Element of the c.DnSPL structure of the situation of payment for the issued invo
  Element of the c.DnSPL structure for a criminal process
  Element of the c.DnSPL structure of the situation of a selling/servicing.
  Element of the c.DnSPL structure of an investigation process
  Element of the c.DnSPL structure of the situation of an invoice issuing
```

```
Nazwa szablonu: porwać PL ACT
Rama wiedzy: porwać
Wersja szablonu: 1
Szablon aktywny: tak
Kotwica:
Typ frazy: VG
Strona czasownika: aktywna
Element:
NazwaSlotu: RolaAgentywnaWPrzebieguIncydentu
Przypadki: nominative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: RolaPacjensa-ObiektuWPrzebieguIncydentu
Przypadki: accusative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: CzlowiekWRoliPacjensa-BeneficjentaWPrzebieguIncydentu
Przypadki: dative
Przyimki:
Typ frazy: NG
Element:
NazwaSlotu: RolaAdlatywnaWPrzebieguIncydentu
Przypadki: dative
Przyimki: ku
Typ frazy: NG
Element:
NazwaSlotu: RolaAblatywnaWPrzebieguIncydentu
Przypadki: dative
Przyimki: dzięki
Typ frazy: NG
Element:
NazwaSlotu: RolaMiejscaZajsciaIncydentu
Przypadki: instrumental
Przyimki: między,nad,pod,pomiędzy,ponad,przed,za
Typ frazy: NG
```

Fig. 5. The screenshot with generated templates.

3 Specification of Event Extraction Process

As it was mentioned, event extraction is a process of textual data analysis with the goal of knowledge acquisition, in the context of the domain semantics specified by an ontology. The extractor uses specific data structures. The architecture of the extractor is shown in Fig. 6.

There are three main data structures used in our EE process:

1. *Event Frames*
2. *Event Templates* and
3. *Event Instances*.

Event Frames and *Event Templates* are domain-dependent and may be either generated on the basis of the ontology or they may be created manually. They are to be instantiated before the extraction process commences, while *Event Instances* form the output of the extractor. *Event Frame* is a semantic description of the event. A singular frame specifies a concept that is hidden under the anchoring word in the event, such as *kidnapping*, as well as pairs: a thematic role and its semantics (semantic classes), such as: an agent role – being for example a *human* concept. Various lexicalizations of the concepts should be necessarily included in the ontology. *Event Templates* specify the valence structure of sentences, which are supposed to be the crucial input source of the extraction. For example, we may specify that the role of an agent should be a nominal phrase while the role of a patient should be a noun phrase with the main noun in the accusative case.

Event instances describe the details of the instantiated extracted events. The extraction starts with the tokenizing and tagging process. Tokenizing is handled by the TaKIPI tool [8]. Then, the tokens are chunked into phrases. The following part of the EE process is executed with the Spejd shallow parsing tool for Polish [11] according to the specified grammar. The chunking process generates three main types of chunks: noun phrases (NP), verb phrases (VP) and prepositional phrases (PP), as well as several technical chunks, such as: complex sentence separator, (which implies that the potential nominal phrase of the latter homogenous part of the sentence is equal to the nominal phrase of the first homogenous part of the sentence (CSnf)) and complex sentence separator (which implies that the potential nominal phrase of the latter homogenous part of the sentence is equal to the word, which appears directly before the separator (CSf)). The chunks consist of tokens, as well as the identifier of the semantically main word in the chunk and the syntactically main word in the chunk (see Fig. 7), the exemplary sentence is taken from NJKP.

Once the chunking process is done, the generated XML output is serialized into the C# application. Data is incorporated into the extractor mechanism. The extractor compares the structure of the serialized data with all of the *Event Frames* and *Event Templates*. If the sentence happens to contain a word that is lexicalized by one of the anchoring words within all of the extraction frames, the sentence is passed for the further examination. In the next step, the extractor tries to match the valence structure with all of the corresponding *Event Templates*. On top of that all of the thematic roles must correspond to the lexicalizations. Details of this process are discussed in [3]. The

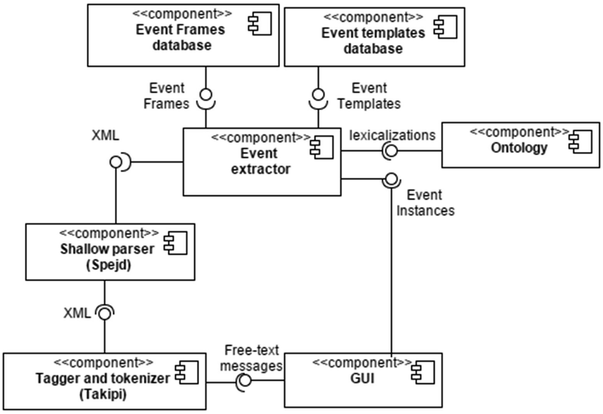


Fig. 6. Architecture of the event extraction process expressed via the UML component diagram.

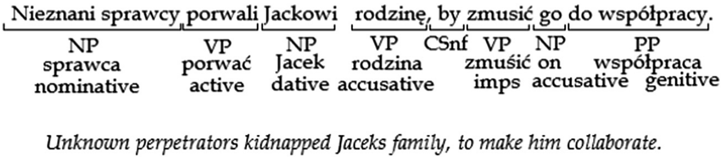

Nieznani sprawcy porwali Jackowi rodzinę, by zmusić go do współpracy.
NP
sprawca
nominative
VP
porwać
active
NP
Jacek
dative
VP
rodzina
accusative
CSnf
VP
zmuśić
imps
NP
on
accusative
PP
współpraca
genitive
Unknown perpetrators kidnapped Jaceks family, to make him collaborate.

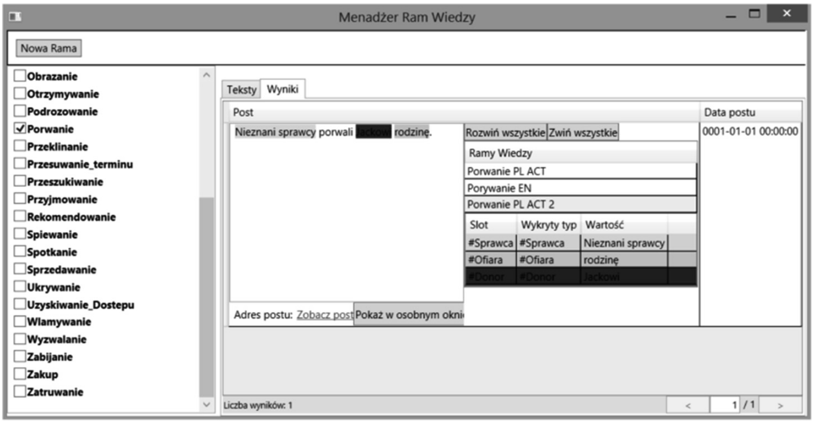
Fig. 7. Chunking of the sentence.

final result of the extraction is visible in Fig. 8. The ontology is the core piece of the extraction process. It contains all the data related to the events semantics, as well as the entirety of the lexicalizations. If the sentence happens to have a word which does not appear within the lexicalizations, the word is passed to the supporting NER system. If the word is still not recognized, it is treated as if it was valid and corresponding to the correct class. Based on the ontology, it is possible to generate all of the *Event Frames* and *Event Templates*. In the next section, we are going to introduce an alternative, manual process of generation of *Event Frames* and *Event Templates*.

4 Event Extraction Templates at Work – a Case Study

The procedure of manual template creation is based on the valence structures of the verb representing kidnapping that are specified in Walenty. The most basic *Event Frame* for the *kidnapping* event should consist of three thematic roles:

- Event Frame name: *kidnapping event*
- Anchoring word: *kidnapping*
- Agent class: *human* (a concept that acts for a *Person* in the c.DnSPL ontology)
- Patient class: *human*
- Beneficiary class: *human*.



- ☐ Obrazanie
- ☐ Otrzymywanie
- ☐ Podrozowanie
- ☑ Porwanie
- ☐ Przeklinanie
- ☐ Przesuwanie_terminu
- ☐ Przeszukiwanie
- ☐ Przyjmowanie
- ☐ Rekomendowanie
- ☐ Spiewanie
- ☐ Spotkanie
- ☐ Sprzedawanie
- ☐ Ukrywanie
- ☐ Uzyskiwanie_Dostepu
- ☐ Wlamywanie
- ☐ Wyzwalanie
- ☐ Zabijanie
- ☐ Zakup
- ☐ Zatruwanie

Post	Data postu
Nieznani sprawcy porwali Jackowi rodzinę.	0001-01-01 00:00:00

Adres postu: Zobacz post

Ramy Wiedzy
- Porwanie PL ACT
- Porywanie EN
- Porwanie PL ACT 2

Slot	Wykryty typ	Wartość
#Sprawca	#Sprawca	Nieznani sprawcy
#Ofiara	#Ofiara	rodzinę
#Donor	#Donor	Jackowi

Liczba wyników: 1

1 / 1

Fig. 8. Extraction process results.

Figure 9 depicts the screen of a tool supporting the manual creation of an *Event Frame*.

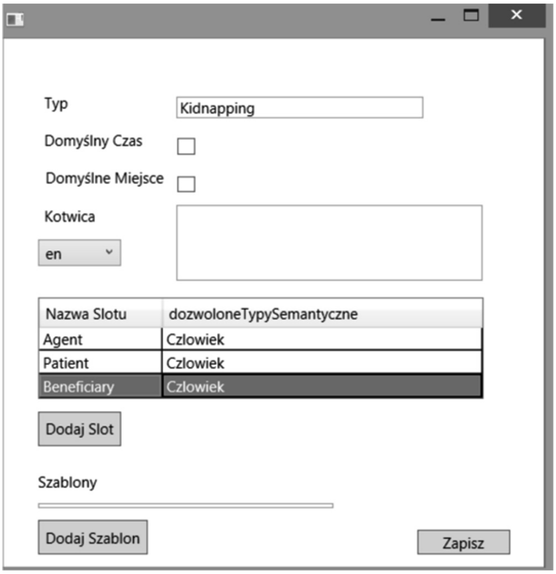
The tool allows one to set up all of the mentioned frame elements (*Typ* – a name, *Kotwica* – an anchoring word, *Nazwa slotu* – a thematic role, *dozwoloneTypySemantyczne* – the allowed ontological concepts), as well as to choose whether the default thematic roles of time (*Domyślny czas*) and space (*Domyślne miejsce*) should be included into it or not. The exactly one point in time and place is specified by the default roles. It should be noted, that events do not happen in exactly one point in time and that they often are specified inaccurately or relatively, as for example the word "yesterday" in the slightly modified NJKP sentence, translated into English as: "*Unknown perpetrators kidnapped Jacek's family yesterday*."

Using the tool one can also specify a set of lexicalizations, which will be added in to the ontological resources. Each *Event Frame* may relate to many *Event Templates*. Let us specify two *Event Templates* for an exemplary frame for *kidnapping*. At first let us consider the valence structure of the active voice of the anchoring verb:

- Related Frame: kidnapping event
- Anchoring word: active verb phrase
- Agent: noun phrase; Case: nominative
- Patient: noun phrase; Case: accusative
- Beneficiary: noun phrase; Case: dative – the valence structure in Polish differs from the English one

and then the structure for the passive voice one:

- Related Frame: kidnapping event
- Anchoring word: passive verb phrase
- Agent: noun phrase; Case: accusative
- Patient: noun phrase; Case: nominative
- Beneficiary: noun phrase; Case: dative



Typ: Kidnapping

Domyślny Czas □

Domyślne Miejsce □

Kotwica

en

Nazwa Slotu	dozwoloneTypySemantyczne
Agent	Czlowiek
Patient	Czlowiek
Beneficiary	Czlowiek

Szablony

Fig. 9. Creating an *Event Frame*.

On the basis of this specification the *Event Template* may be created (see Fig. 10).

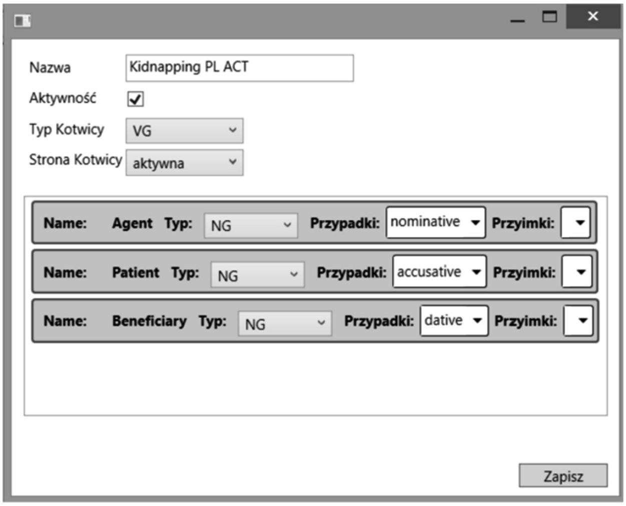
The tool allows one to specify all of the previously mentioned features as well as the set of prepositions (*Przyimki*) suitable for the valence structure of verb. Once the operation of creating *Event Templates* and *Event Frames* is done, the data is stored in the ontology. The defined set of structures enables to extract knowledge from the following two sentences:

> "*Nieznani sprawcy porwali Jackowi rodzinę*." ("*Unknown perpetrators kidnapped Jacek's family*.")
> "Rodzina Jacka została porwana przez nieznanych sprawców." ("*Jacek's family was kidnapped by unknown perpetrators.*")

This basic scenario works properly in every similar case of a simple sentence. Let us consider the more complicated (NJKP-based) sentence:

> "*Jeden z nich w rozmowie lekko odwrócił się od stołu, a wtedy orzechówka porwała z talerzyka ogromny kawał kiełbasy i uciekła.*"
> ("*One of them slightly turned back out of table, and then a nutcracker grabbed a huge piece of sausage from the plate and ran away.")*

First of all, the phrase "*Jeden z nich*", "*One of them*" is never going to be picked up as the *Agent*, because the latter part of the sentence contains a noun phrase with the main noun in nominative case, i.e. "*orzechówka*" (a *nutcracker*). However, there are several possible outcomes of the extraction process. The results depend both on the vocabulary stored within lexicalizations and on the structure of the event templates. If the words used in the sentence are stored within the set of lexicalizations, the event extractor is not going to make a mistake. Else, if there are no such words stored within



Nazwa: Kidnapping PL ACT
Aktywność: ☑
Typ Kotwicy: VG
Strona Kotwicy: aktywna

Name: Agent Typ: NG Przypadki: nominative Przyimki:
Name: Patient Typ: NG Przypadki: accusative Przyimki:
Name: Beneficiary Typ: NG Przypadki: dative Przyimki:

Fig. 10. Creating an Event Template.

lexicalizations, the extractor assumes, that the used word is a correct one. This approach is weak to the homonymic words, such as "*orzechówka*". The existence of such a word under only one of the semantic classes is most likely going to be confusing in the extraction process. The possible results of extraction from the considered sentence, with regard to the homonymic nature of the word "*orzechówka*" are listed in Table 2.

The outcomes denoted with the asterisk (see Table 2) incorrectly generate *Event Instances* and should be separated from the others. They illustrate the crucial role of lexicalizations assigned to ontological concepts. This examples also provide an additional information upon further inspection. The acquired in this way information may be used for the vocabulary enrichment. Methods of the vocabulary enrichment are crucial in the process of the accuracy improvement. There is another problem in this particular sentence. The phrase, which describes the patient, consists of nouns in accusative and genitive cases. The phrases in which the other word is in accusative, make it very hard to recognize the semantically most important word. For example:

"*Rodzina* (nominative - semantically the most important word) *Jacka* (genitive)"
("*Jacek's Family*") and
"*Kawał* (nominative) *kiełbasy* (genitive - semantically the most important word)"
("*a huge piece of sausage*")

The extractor assumes, that the first word in the phrase is semantically the most important word. In this particular case it does not interfere with the results, however it is a problem, which should be resolved to improve the accuracy of the extraction.

Another example exposes one of the weaknesses of the described approach. The considered sentence is as follows (taken from NJKP):

"*Gwiazda wieczoru, Helena Vondračkova wystąpiła dopiero około 00:30 i od razu porwała widownię do tańca.*"

Table 2. Possible outcomes of the extraction process.

Event Template details	Vocabulary details	Result of the extraction
Patient: either a *human* or a *food* *Agent*: either a *human* or an *animal*	no "orzechówka" (a *nutcracker*) lexicalization under the *animal* class, but there is an "orzechówka" (*whale liqueur*) under another class	no *Event Instance* extracted
Patient: either a *human* or a *food* *Agent*: either a *human* or an *animal*	"orzechówka" (a *nutcracker*) lexicalization under the *animal* class	*Event Instance* extracted correctly
Agent: a *human*	"orzechówka" (a *nutcracker*) lexicalization under the *animal* (assuming it subsumes a *human*) class	no *Event Instance* extracted
Patient: a *human* *Agent*: an *animal*	"orzechówka" (*whale liqueur*) lexicalization under the *liqueurs* class	*Event Instance* extracted incorrectly*
Agent: a *human*	no "orzechówka" (a nutcracker) lexicalization in the ontology	*Event Instance* extracted incorrectly*

> *"The star of the evening, Helena Vondračkova performed only at 00:30 and at once 'kidnapped' the audience to dance."*

The idiomatic structures are a problem in our approach. The phrase "*to snatch to dance*" (literally in Polish "*to kidnap to dance*") is a commonly used idiom. In this case, the extraction process generates the *Event Instance* with "*Helena Vondračkova*" as the agent and "*the audience*" as the patient. Both lexicalizations might correctly appear under the human class in the ontology. At first glance, this extraction is correct, however it does not satisfy the intention of the *Event Frame*. The extractor is supposed to find the information about the kidnappings. Generating separate *Event Frames* and ontological classes dedicated for the idiomatic meanings is a solution to this problem. Then, if the extractor did generate an idiomatic *Event Instance*, it should not generate any other *Event Instances*. But such a solution, however, generates a lot of redundancy in the ontology. That problem is to be investigated further.

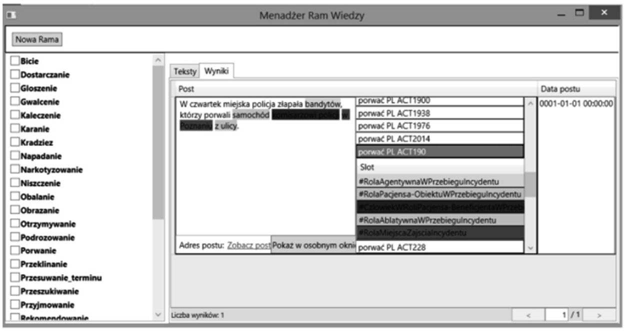
We have run several experiments with sets of *Event Templates* generated automatically as described in Sect. 2. This approach provided us with the additional information. Creating *Event Templates* manually usually covers several, most commonly used valence structures, while the before-mentioned approach generates all of the possible templates. There are 53 *Event Templates* for active voice of the kidnapping verb. Let us consider the following sentence (NJKP-based):

> "*Nieznani sprawcy porwali Krystynę Starczewską z ulicy*." ("*Unknown perpetrators kidnapped Krystyna Starczewska from the street.*")

With the set of templates presented at the beginning of the section, the extractor generates an *Event Instance* with two thematic roles. The considered *Event Instance* is partially correct, as it does express the meaning of the sentence, but it omits the ablative role "from the street". Obviously, if we were more careful, we would put that role in the template, as well as many other thematic roles (see Sect. 2). However, by doing so, we do not neither provide nor receive any additional information about the valence structure of the sentence. Creating templates for all of the valence structures is extremely time consuming. With 53 generated templates we created 53 Event Instances, out of which 32 cover the ablative role "from the street", and only one *Event Instance* contains 3 thematic roles, which correspond to 3 slots in the *Event Template*. We must note, however, that the time of the extraction process is particularly prolonged by tokenizing and parsing processes, not due to applying many templates.

We have performed similar tests for various sentences with the active and passive voice structures. We achieved similar results in every case. Another example of the *Event Instance* extracted from the complex sentence is depicted in Fig. 11. One of the many extracted *Event Instances* covers all of the thematic roles. In this particular case, there is no word "*samochód*" (a *car*) within the lexicalizations. It makes the extractor assume, that a *car* is a correctly used word, which is unknown to the system. If there was a word "*samochód*" within the set of lexicalizations, the extraction would be aborted.

> "*W czwartek miejska policja złapała bandytów, którzy porwali samochód komisa-rzowi policji w Poznaniu z ulicy*" (*"On Thursday the city police captured bandits, who sized a car from the street, which belonged to the Poznań police-officer."*)



W czwartek miejska policja złapała bandytów, którzy porwali samochód komisarzowi policji w Poznaniu z ulicy.

0001-01-01 00:00:00

porwać PL ACT1900
porwać PL ACT1938
porwać PL ACT1976
porwać PL ACT2014
porwać PL ACT190

Slot
#RolaAgentywnaWPrzebieguIncydentu
#RolaPacjensa-ObiektuWPrzebieguIncydentu
#CzlowiekWRoliPacjensa-BeneficjentaWPrzeb
#RolaAblatywnaWPrzebieguIncydentu
#RolaMiejscaZajsciaIncydentu

porwać PL ACT228

Fig. 11. Extracted Event Instance.

5 Final Remarks

In this paper we provide a view on EE process with regard to the generation of *Event Frames* and *Event Templates*. We firmly believe that the automated process of template generation provides data which will be useful in the process of vocabulary (which is assigned to ontological concepts) enrichment, valence structures studies and, ultimately, accuracy improvement in the process of event extraction. We believe, that the research is going to be useful, not only in the event extraction, but also in the synthetic text generation. As it was mentioned, the process of generation of extraction templates and frames strongly relies on the domain semantics expressed in the form of an ontology. Creation of c.DnSPL ontologies is not an easy task thus recently we developed a tool to support this process [2]. In the future we plan to support the process of extraction by using rich language resources contained in LLOD network [6]. Also, we are working on the method and tool enabling to assign the proper and rich lexicalizations to ontological concepts. There are at least two problems to be solved: finding a suitable model of lexicalizations and populating the model with relevant lexical data. The lexical "seed" for an ontological concept, in the simplest solution, may be given by the user by means of a dedicated tool – then, the tool should find proposals of other lexicalizations using, for example, some of the above mentioned LLOD language resources and their translations into Polish.

References

1. Cybulka, J.: The OWL version of c.DnSPL ontology. http://users.man.poznan.pl/jolac/PPBW-22-07-2015-inferred-new.owl (20 MB). Accessed 07 July 2017
2. Cybulka, J.: Supporting the creation of some class of well-founded OWL-DL ontologies. Comput. Methods Sci. Technol. **23**(5), 57–64 (2017)
3. Dutkiewicz, J., Falkowski, M., Nowak, M., Jędrzejek, C.: Semantic extraction with use of frames. In: Przepiórkowski, A., Ogrodniczuk, M. (eds.) NLP 2014. LNCS (LNAI), vol. 8686, pp. 208–215. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10888-9_22
4. Gangemi, A., Lehmann, J., Catenacci, C.: Norms and plans as unification criteria for social collectives. http://drops.dagstuhl.de/opus/volltexte/2007/910. Accessed 07 July 2017
5. Jaworski, W., Przepiórkowski, A.: Syntactic approximation of semantic roles. In: Przepiórkowski, A., Ogrodniczuk, M. (eds.) NLP 2014. LNCS (LNAI), vol. 8686, pp. 193–201. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10888-9_20
6. McCrae, J.P., et al.: The open linguistics working group: developing the linguistic linked open data cloud. In: 10th International Conference on Language Resources and Evaluation, Portorož, Slovenia, pp. 23–28, May 2016. http://www.lrec-conf.org/proceedings/lrec2016/pdf/851_Paper.pdf. Accessed 5 July 2017
7. Palmer, M.: VerbNet − A Class-Based Verb Lexicon. http://verbs.colorado.edu/~mpalmer/projects/verbnet.html. Accessed 05 July 2017
8. Piasecki, M.: Polish tagger TaKIPI: rule based construction and optimisation. TASK Q. **11** (1–2), 151–167 (2007)
9. Piskorski, J., Yangaber, R.: Information extraction: past, present and future. In: Poibeau, T., et al. (eds.) Multi-source, Multilingual Information Extraction and Summarization. NLP, pp. 23–49. Springer, Cham (2013). https://doi.org/10.1007/978-3-642-28569-1_2. ISBN 978-3-642-28568-4
10. Proceedings of the 4th Conference on Message Understanding, MUC4 1992, McLean, Virginia. Association for Computational Linguistics, USA (1992). ISBN 1-55860-273-9
11. Przepiórkowski, A.: Powierzchniowe przetwarzanie języka polskiego. Akademicka Oficyna Wydawnicza EXIT, Warszawa (2008). (in Polish)

Understanding Questions and Extracting Answers: Interactive Quiz Game Application Design

Volha Petukhova(✉), Desmond Darma Putra, Alexandr Chernov, and Dietrich Klakow

Spoken Language Systems Group, Saarland University, Saarbrücken, Germany
{v.petukhova,d.darma,a.chernov,d.klakow}@lsv.uni-saarland.de

Abstract. The paper discusses two key tasks performed by a Question Answering Dialogue System (QADS): user question interpretation and answer extraction. The system represents an interactive quiz game application. The information that forms the content of the game is concerned with biographical facts of famous people's life. The process of a question classification and answer extraction is performed based on a domain-specific taxonomy of semantic roles and relations computing the Expected Answer Type (EAT). Question interpretation is achieved performing a sequence of classification, information extraction, query formalization and query expansion tasks. The expanded query facilitates the search and retrieval of the information. The facts are extracted from Wikipedia pages by means of the same set of semantic relations, whose fillers are identified by trained sequence classifiers and pattern matching tools, and edited to be returned to the player as full-fledged system answers. The results (precision of 85% for the EAT classification of both in questions and answers) show that the presented approach fits the data well and can be considered as a promising method for other QA domains, in particular when dealing with unstructured information.

1 Introduction

Question-Answering (QA) applications have gained steady growing attention over past decades. Three major approaches can be observed. The first one is the Information-Retrieval (IR) based QA system consisting of three main components: question processing, passage retrieval, and answer ranking [5]. The second paradigm is a knowledge-based QA system as used by Apple Siri[1], Amazon Alexa[2], Wolfram Alpha[3], etc. Such systems, first, build a query representation and then map it to structured data like ontologies, gazetteers, etc. The third approach combines these two methods.

[1] http://www.apple.com/ios/siri/.

[2] https://developer.amazon.com/alexa.

[3] www.wolframalpha.com.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 246–261, 2018.
https://doi.org/10.1007/978-3-319-93782-3_18

We aim at building an end-to-end Question Answering Dialogue System (QADS) that provides an interactive guessing game where players have to ask questions about attributes of an unknown person in order to guess his/her identity. The system adopts a statistical approach by employing state-of-the-art supervised machine-learning algorithms run on features such as n-grams, POS (Part-of-Speech), Named Entity (NE), syntactic chunks, etc. The main differences between our QA system and those of the others, in general, are that our domain is rather closed, and that the content that the system operates on is mainly unstructured free texts. What is more important, our system is an interactive QADS where the answers are returned to the user not as extracted information chunks or slot fillers, but are rather full-fledged dialogue utterances.

The core module of the QADS is the Dialogue Engine which consists of four main components: interpretation module, dialogue manager, answer extraction module and utterance generation module. The Dialogue Manager (DM) takes care of the overall communication between the user and the system. It gets as an input from the interpretation module a dialogue act representation. Mostly it is about a question which is uttered by the human player. Questions are classified identifying their communicative function (e.g. Propositional, Check, Set and Choice Questions) and semantic content in accordance to the ISO 24617-2 dialogue act standard [17]. Semantic content is determined based on Expected Answer Type (EAT). To extract the requested information, a taxonomy is designed comprising 59 semantic relations to cover the most important facts in human life, e.g. birth, marriage, career, etc. The extracted information is mapped to the EAT, and both the most relevant answer and a strategy for continuing the dialogue are computed. The Dialogue Manager then passes the extracted information for further system response generation, where the DM input is transformed into a dialogue utterance (possibly multimodal one).

For a closed domain as ours, restricted to personal biographical facts, it is possible to narrow down the knowledge available to the system. For example, structured knowledge bases can be used, e.g. Freebase[4]. They are, however, not complete to achieve sufficient coverage of factual information required for our game. Therefore, the content that the system operates on is a bigger collection of unstructured free texts, namely Wikipedia articles[5]. This impacts search and retrieval tasks. As a consequence, the output of the question understanding module should be a rather comprehensive query capturing various semantic information concerning events in question, entities involved in this event and their properties, and type of relations between entities and possibly between events, EAT. The EAT is augmented with question focus word(-s) to determine the main event in question. The EAT together with focus word(-s), are formalized in a query which, on its turn, is expanded to cover as many natural language variations as possible. To extract the requested information, information is mapped to the EAT and focus word. The answer extraction module operates on unstructured unprocessed data as input, i.e. Wikipedia articles, and its

[4] http://www.freebase.com/.

[5] http://www.wikipedia.org.

design based on trained classifiers and post-processing tools to extract semantic relation automatically with reasonably high accuracy.

The paper is structured as follows. Section 2 gives an overview of previous approaches to QA system design. Section 3 defines semantic relations as a framework for this study. Section 4 describes the annotated data. Classification results using semantic relations in questions (Sect. 5) and for answer extraction (Sect. 6) are presented. We also outline performed experiments describing features, algorithms and evaluation metrics that have been used. We discuss how the query is generated and expanded, and how the full answer extraction procedure is designed. Section 6 concludes the reported study and outlines future research.

2 Question Answering: Related Work

A breakthrough in QA has been made by [5] when designing an end-to-end open-domain QA system. This system achieved the best result in the TREC-8 competition[6] with accuracy of 77.7%. The system consists of three modules such as question processing, paragraph indexing and answer processing. First, the question type, question focus, question keyword and expected answer type are specified. Further, the search engine is used to retrieve the relevant documents and filter candidate paragraphs. Subsequently, the answer processing module identifies the answer in the paragraph using lexico-semantic information (POS, Gazetteers, WordNet and Named Entities), and after scoring candidates using word similarity metric returns the highest ranked answer.

In 2010, Watson, a DeepQA system of IBM Research [3], won a Jeopardy quiz challenge. This system incorporates content acquisition, question analysis, hypothesis generation, etc. For the hypotheses generation, it relies on named entity detection, triple store and reverse dictionary look-up to generate candidate answers which are then ranked based on confidence scores.

The most recent work comes from the TAC KBP slot filling task [8] aimed at finding fillers for each identified empty slot, e.g. for person (e.g. date_of_birth, age, etc.) and/or for a organization (e.g. member_of, founded_by, etc.). Pattern matching, trained classifiers and Freebase are used in [1,2] to find the best filler. The best system performance achieved in terms of F-score is 37.28% [18,20].

The TAC KBP approach differs from TREC tasks in that the former focuses on entities such as person or organization, while the later has a broader focus (person, organization, location, etc.). Secondly, TAC KBP slot filling has determined 41 slots that need to be filled, while in TREC, the information that needs to be found depends on the question asked. Finally, in terms of questions, TAC questions are defined by a topic and a list of slots that needs to be filled, while in TREC they vary from simple factoids to more complex questions.

Analysing the above mentioned studies, we concluded that computing the Expected Answer Type (EAT), classification, focus word extraction, query generation and expansion and pattern matching are important steps to robust

[6] http://trec.nist.gov/pubs/trec8.

question classification and answer extraction. Since our task, domain and data differ as mentioned above, the following extensions were performed:

– the TAC KBP 2013 relations set was enriched to compute EAT;
– different syntactic and semantic parsers for better coverage of relevant phenomena were applied;
– different classifiers and classification procedures were evaluated to determine the EAT in questions and answer candidates, and to establish the answer's boundaries;
– the EAT information was enriched with query focus word(-s) and expanded with synonyms to enable efficient and accurate answer extraction;
– matching patterns to capture the defined relations were designed;
– ranked answer candidates were post-processed and redundancies removed.

3 Semantic Framework: Relations

To understand a question and to find a correct answer to this question semantic roles are often used. A semantic role is a relational notion describing the way a participant is involved in an event or state [16], typically providing answers to questions such as"who" did "what" to "whom", and "when", "where", "why", and "how". Several semantic role annotation schemes have been developed in the past, e.g. FrameNet [29], PropBank [28] and Lirics [30]. Along with semantic roles, relations between participants are also relevant for our domain, e.g. the relation between Agent and Co-Agent (or Partner) involved in a 'work' event may be a COLLEAGUE_OF relation.

Depending on the domain and task, QA systems may require different kinds of question and answer type taxonomies. For instance, Singhal et al. (2000) designed a very simple taxonomy based on the correspondence between question words and expected answer types. For instance, according to this taxonomy, questions containing *who* or *whom* answers of the type *person*. For more ambiguous question words like *what* or *which* the type of a question was identified by the head noun.

Moldovan et al. (2000) define 9 question classes (e.g. *'what'*, *'who'*, *'how'*) and 20 sub-classes (e.g. *'basic what'*, *'what-who'*, *'what-when'*). Additionally, expected answer type is determined, e.g. *person*, *money*, *organization*, *location*. Finally, a focus word or a sequence of words is identified in the question, which disambiguates it by indicating what the question is looking for (see [5] for an overview of defined classes for 200 of the most frequent TREC-8 questions).

Li and Roth (2002) proposed another question classification scheme, also based on determining the expected answer type. This scheme is a layered hierarchical two-level taxonomy. The first level represents coarse classes like *Date*, *Location*, *Person*, *Time*, *Definition*, *Manner*, *Number*, *Title*, *Organization*, *Reason*, etc. The second level comprises 50 fine-grained classes like *Description*, *Group*, *Individual* and *Title* for the upper-level class of *Human*. Using a hierarchical classifier they tried to get an increase in performance, but experimental results showed that the gained difference with a flat classifier was not statistically significant.

The TAC KBP slot filling task [8] aimed at finding fillers for each identified empty slot, e.g. for a person (e.g. date_of_birth, age, etc.) and/or for an organization (e.g. member_of, founded_by, etc.).

To decide on the set of relations to investigate, we analysed already available and collected new dialogue data. As a starting point, we analysed recordings of the famous US game 'What's my line?' that are freely available on Youtube[7]. However, the latter differs from our scenario: during the TV-show participants may ask only propositional questions with expected 'yes' or 'no' answers; our game allows any question type from the user. Therefore, we collected data in pilot dialogue experiments, where one participant was acting as a person whose name should be guessed and the other as a game player. 18 dialogues were collected of total duration of 55 min comprising 360 system's and user's speaking turns. To evaluate the relation set and to train classifiers, we performed large scale gaming experiments in a Wizard of Oz setting, see next section.

Pilot experiments showed that all players tend to ask similar questions about gender, place and time of birth or death, profession, achievements, etc. To capture this information we defined 59 semantic relations proposing a multi-layered taxonomy: a high level, coarse annotation comprising 7 classes and a low-level, fine-grained annotation, comprising 52 classes, see [24] for more details. This includes the HUMAN DESCRIPTION class defined for acts about an individual such as age, title, nationality, etc.; HUMAN RELATIONS for family relations; HUMAN GROUPS for relations between colleagues, friends, etc.; EVENTS & NON-HUMAN ENTITIES class for awards, products of human activities, etc.; EVENT MODIFIERS for specifying manner, reasons, etc.; the TIME class to capture temporal information like duration, frequency, etc.; and the LOCATION class to capture spatial event markers for places where events occur. Table 1 presents the subset of about the 30 most frequently occurring relations with an indication of what concepts can be found in existing schemes for annotating semantic relations and semantic roles. We also provide relative frequencies of the annotated questions and answers in the data. It should be noted here that the majority of the concepts defined here are domain-specific, i.e. tailored to our quiz game application. The approach could however be adapted for designing comparable annotation schemes for other domains; this has for example been done for the food domain (see [31]).

Each relation has two arguments and is one of the following types:

- RELATION(Z, ?X), where Z is the person in question and X the entity slot to be filled, e.g. CHILD_OF(einstein, ?X);
- RELATION(E$_1$, ?E$_2$) where E$_1$ is the event in question and E$_2$ is the event slot to be filled, e.g. REASON(death, ?E$_2$); and
- RELATION(E, ?X) where E is the event in question and X the entity slot to be filled, e.g. DURATION(study, ?X).

The slots are categorized by the entity type which we seek to extract information about. However, slots are also categorized by the content and quantity of their fillers [8].

[7] https://www.youtube.com/channel/UChPE75Fvvl1HmdAsO7Nzb8w.

Slots are labelled as *name*, *value*, or *string* based on the content of their fillers. *Name* slots are required to be filled by the name of a person, organization, or geo-political entity (GPE). *Value* slots are required to be filled by either a numerical value or a date, e.g. *December 7, 1941*, *42*, *12/7/1941. String* slots are basically a "catch all", meaning that their fillers cannot be neatly classified as names or values.

Slots can be as *single-value* or *list-value* based on the number of fillers they can take. While single-value slots can have only a single filler, e.g. date of birth, list-value slots can take multiple fillers having more than one correct answer, e.g. employers.

4 Data: Collection and Annotations

In order to validate the proposed EAT annotation scheme empirically and to build an end-to-end QADS, two types of data are required: (1) dialogue data containing player's questions that are more realistic than youtube games and larger than our pilots; and (2) descriptions containing answers to player's questions about the guessed person.

To collect question data we explored different possibilities. There is some question data publicly available, e.g. approximately 5500 questions are provided by the University Illinois[8] annotated according to the scheme defined in [10]. However, not all of this data can be used for our scenario. We filtered out about 400 questions for our purposes. Since this dataset is obviously too small, we generated questions automatically using the tool provided by (Heilman and Smith, 2009) from the selected Wikipedia articles and filtered them out manually: grammatically broken questions were fixed and repetitions deleted. Additionally, synonyms from WordNet[9] were used to generate different variations of questions for the same class.

We collected game data in *Wizard of Oz* experiments on a larger scale then pilot ones. Here again one participant was acting as a Wizard simulating the system's behaviour (2 English native speakers: male and female) and the other as a game player (21 unique subjects: undergraduates of age between 19 and 25, who are expected to represent our target audience). 338 dialogues were collected of total duration of 16 hours comprising about 6.000 speaking turns, see [23].

The final question set consists of 1069 questions. Table 1 illustrates the distribution of question and answer types in terms of the EAT.

Additionally, a focus word or words sequence specifying the main event in a question, usually a verb or eventive noun, is extracted from the question to compute the EAT and formulate the query. For example,

(1) Question: When was his first album released?
Assigned semantic relation: TIME
Focus word sequence: first album released

[8] http://cogcomp.cs.illinois.edu/page/resources/data.

[9] http://wordnet.princeton.edu/.

Table 1. Question and answer types in terms of defined semantic relations and their distribution in data (relative frequency in %; (① means that the relation is also defined in TAC KBP slot filling task; ② in TREC-08 QA task; ③ in TREC 2002 QA task, i.e. annotation scheme proposed by [10]; and ④ in LIRICS semantic role set)).

RELATION	Questions (%)	Answers (%)	RELATION	Questions (%)	Answers (%)
ACTIVITY_OF	10.2	4.0	LOC_BIRTH	2.3	5.0
AGE_OF① ②	3.0	2.1	AWARD	4.4	2.5
LOC_RESIDENCE	1.7	3.2	MEMBER_OF①	2.4	1.8
CHILD_OF①	1.5	3.6	COLLEAGUE_OF	1.0	1.7
NATIONALITY①	1.2	3.1	CREATOR_OF	6.1	8.5
OWNER_OF	2.0	1.1	PARENT_OF①	1.3	3.7
DURATION④	1.3	1.8	EDUCATION_OF①	3.7	4.2
RELIGION	2.5	0.7	EMPLOYEE_OF①	1.6	2.2
SIBLING_OF①	0.9	2.3	SPOUSE_OF①	1.4	1.9
FAMILY_OF	1.6	-	FOUNDER_OF①	1.9	1.2
TIME② ③ ④	8.0	14.6	TIME_BIRTH	2.1	2.8
TIME_DEATH	1.6	1.0	LOCATION ② ③ ④	4.7	5.6
TITLE① ③	11.1	14.2	LOC_DEATH	1.7	0.8
PART_IN	-	3.6	CHARGED_FOR	4.2	-
GENDER	1.7	-	NAME	1.9	-

EAT: TIME_release(first_album)
Query: TIME_release(first_album) :: (E, ?X) :: QUALITY(VALUE) :: QUANTITY(SINGLE)

Answers were retrieved from 100 selected English Wikipedia articles containing 1616 sentences (16 words/sentence on average), 30.590 tokens (5.817 unique tokens). Descriptions are annotated using complex labels consisting of an IOB-prefix (**I**nside, **O**utside, and **B**eginning) and the EAT tag to learn the exact answer boundaries. We mainly focus on labeling nouns and noun phrases. For example:

(2) *Gates graduated from* ***Lakeside School*** *in 1973.*

The word *Lakeside* in (2) is labeled as the beginning of an EDUCATION_OF relation (B-EDUCATION_OF), and *school* is marked as inside of the label (I-EDUCATION_OF). Table 1 illustrates the distribution of the most frequently occurring answer types based on the identified semantic relation.

Since the boundaries between semantic classes are not always clear, we allowed multiple class labels to be assigned to one entity. For example:

(3) *Living in Johannesburg, he became involved in anti-colonial politics, joining the ANC and becoming a founding member of its* ***Youth League****.*

Here, *Youth League* is founded by a person (FOUNDER_OF relation), but the person is also a member of the *Youth League*. There are also some overlapping segments detected as in example (4):

(4) *He served as* ***the commander-in-chief of the Continental Army*** *during the American Revolutionary War.*

The entity *commander-in-chief of the Continental Army* in (4) is marked as TITLE, while *the Continental Army* is recognized as MEMBER_OF. Both of these relations are correct, since if a person leads an army he/she is also a member of it. To assess the reliability of the defined tagset, the inter-annotator agreement was measured in terms of the Cohen's kappa [19]. For this, 10 randomly selected descriptions and all 1069 questions were annotated by two trained annotators. The obtained *kappa* scores were interpreted as annotators having reached good agreement (averaged for all labels, kappa = .76).

5 Question Classification

5.1 Classification Design: Classifiers, Features and Evaluation

We defined the question classification task as a machine-learning task, for which we built Support Vector Machines (SVM, [4]) based classifiers. In all experiments, linear kernel function (linearSVC) was used. We performed stratified 5-fold cross-validation multi-class and multi-label classification experiments applying cascade classification procedure. This implies that the first set of classifiers was trained to classify coarse labels (7 top classes) and coarse class predictions were added as features to perform fine-grained classification (cascading).[10]

We conducted a series of experiments to assess: (1) the features' importance for the defined task; and (2) classifiers performance in the cascade setting.

Since the system's goal is to provide the player with a correct answer, and if no answer was found to acknowledge the fact by generating negative feedback utterances like "*Sorry, I do not have this information*"[11], the classifier precision has been considered in evaluation. The trained classifiers performance was compared to the baseline. The baseline was computed based on a single feature, namely, *bag-of-words* when training the Naive Bayes classifier. The baseline classifier was implemented using Multinomial Naive Bayes algorithm from scikit-learn [26] achieving the precision of 56%.

Features computed from the data include *bag-of-words, bigrams, trigrams* and *part-of-speech (POS) tags* using the Stanford CoreNLP tools[12]. In our experiments *surface word forms* and *lemmas* of *focus words* as the most salient words

[10] Another classification procedure is known as *hierarchical* classification. Hierarchy of classifiers consists of classifier#1 deciding to which coarse class a question belongs and transfers this information to the corresponding classifier trained specifically to predict this particular question type.

[11] To make the game more entertaining, the system can always play with strategies to turn a negative situation in a system's favour. For example, if no answer was found, the system may ask the player to ask another question claiming that the previous one was not eligible for whatever reasons or the answer to it would lead to quick game end, or alike.

[12] http://nlp.stanford.edu/downloads/corenlp.shtml.

were used as features. Apart from that, we applied combinations of all the above mentioned features with coarse class labels to predict fine classes.

To extract focus words, we implemented an algorithm that preserves the main nominal phrase with the predicate, corresponding prepositions and conjunctions while removing everything else. The algorithm excludes stop words and stop phrases (from predefined lists), as well as some parts of speech (based on the Penn Tree Bank tagset[13] we remove existential *there*, interjections, interrogative pronouns and possessive endings), auxiliary verbs, and interrogative pronouns. Questions from the real dialogue data were manually annotated with focus words, which allowed to test this algorithm. It was able to extract focus words with the accuracy of 94.6%.

5.2 Experimental Results

As for features, the best results were obtained by the model trained on *unigrams+bigrams of lemmas*. In most cases models based on *unigrams+bigrams* demonstrated significantly better results than *unigram, bigram* or *trigram models*. It means that the word order is important to classify questions correctly. Our classifier outperformed the baseline (X^2 (1, n = 2403) = 293.181, $p < .05$). Table 2 summarizes results from all experiments.

Adding coarse class labels as additional features did not result in a significantly higher precision. The classifier that predicts coarse class labels achieved the average precision of 90%. However, it was not enough to make the predicted coarse class labels useful as features for the fine-grained classification.

As for separate classes, questions of the most prevailing classes were identified with the highest precision: TITLE - 85%, CREATOR_OF - 81%, NAME - 89%.

As expected, the classifier achieved the best results by using lexical clues, i.e. the presence or absence of certain words is a strong feature to determine to which

Table 2. Precision of the question classifier for fine classes (CP - coarse class labels predicted by the classifier).

Experiment 1						Experiment 2					
Features	n-grams range					Features	n-grams range				
	1, 1	1, 2	2, 2	2, 3	3, 3		1, 1	1, 2	2, 2	2, 3	3, 3
Words	0.81	0.81	0.72	0.72	0.68	+CP	0.82	0.81	0.77	0.76	0.70
POS	0.27	0.40	0.40	0.46	0.44	+CP	0.41	0.60	0.59	0.60	0.56
Lem	0.80	**0.82**	0.74	0.73	0.71	+CP	0.81	**0.82**	0.78	0.78	0.75
Words+POS	0.80	0.81	0.71	0.71	0.68	+CP	0.81	0.81	0.77	0.76	0.70
Lem+POS	0.80	0.82	0.73	0.73	0.71	+CP	0.81	0.82	0.78	0.78	0.75
Focus	0.75	0.74	0.63	0.59	0.31	+CP	0.79	0.77	0.68	0.65	0.44
FocusLem	0.76	0.76	0.65	0.61	0.39	+CP	0.79	0.79	0.71	0.68	0.48

[13] http://www.cis.upenn.edu/~treebank/.

class or classes a question belongs. Unfortunately, when a question contains words shared by questions belonging to different classes, it caused prediction errors. Extensive error analysis and learnability experiments were performed, see [22].

5.3 Query Generation and Expansion

Query generation is the last data processing operation that is performed by the question interpretation module. The query is generated according to the predefined set of rules. It captures the results of the question classification process as well as the extracted focus words and transfers this information to the next module.

Table 3. Example of an expanded query.

Question	What do you do as a job?
Focus words	do as job
Expanded focus	do [make, perform, cause, practice, act], as, job [activity, occupation, career, employment, position]
EAT	Title_do(do as job)
Query	(Z, E, ?X) :: Title_do(Z, doAs, ?job) :: QUALITY(String) :: QUANTITY(List) :: FOCUS(do as job)
Expanded query	(Z, E, ?X) :: Title_do(Z, doAs, ?job) :: QUALITY(String) :: QUANTITY(List) :: FOCUS(do [make, perform, practice, act], as, job [activity, occupation, career, employment, position])

The query generation processes, the semantic representation of its components in particularly, partially based on the Discourse Representation Theory (DRT) [27]. It incorporates semantic information that is necessary to find the correct answer. Table 3 demonstrates an example of such a query.

In natural language the same message may have a number of realizations. So far, our QA system misses many answers when the answer is expressed by different lexical units. To solve this problem, we used WordNet[14] synonyms to elaborate the extracted question focus words.

6 Answer Extraction

Figure 1 depicts the answer extraction procedure. The process starts with splitting the data into training and test sets, 80% and 20% respectively. Subsequently, features are extracted for both sets and two sequence classifiers are applied. Additionally, a pattern matching tool is used to predict the outcome based on regular expressions. All predictions are then post-processed to return the final answer.

[14] http://wordnet.princeton.edu.

6.1 Classifiers, Features and Evaluation

Two well-known sequence classifiers such as Conditional Random Field (CRF) [6] and Support Vector Machine (SVM) [9] are trained.[15]

The selected set of features includes *word* and *lemma* tokens as two basic features for classifiers; *POS* tags from the Stanford POS tagger [14]; *NER* tags from three different NER tools: Stanford NER [13], Illinois NER [11], and Saarland NER [15]; *chunking* using OpenNLP[16] to determine the NP boundaries; *key word* to determine the best sentence candidate for a particular relation, e.g. marry, marriage, husband, wife, spouse for the SPOUSE_OF relation; *capitalization* to detect relations between NEs.

To assess the system performance standard evaluation metrics are used, precision (P), recall (R) and F-score (F1), using the tool developed by [12]. In particular, precision is important, since it is worse for the game to give the wrong answer than to say it cannot answer a question.[17] A classifier prediction is considered as correct if both the IOB-prefix and the relation tag fully correspond to those in the referenced annotation.

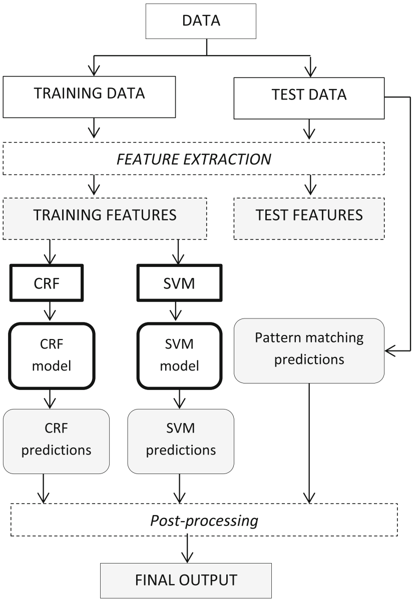


Fig. 1. Answer extraction pipeline.

Table 4. Overall system performance *) applied only to 12 most frequently occurred relations. P stands for precision; R for recall; F1 for harmonic mean.

Classifier	Baseline			System 1			System 2			System 3		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
CRF++	0.56	0.34	0.42	0.68	0.52	0.59	0.82	0.55	0.66	0.85	0.54	0.66
CRFs_AP	0.33	0.29	0.30	0.54	0.53	0.53	0.71	0.57	0.63	0.74	0.56	0.64
CRFs_LBFGS	0.37	0.65	0.44	0.67	0.52	0.58	0.82	0.53	0.65	0.85	0.53	0.65
SVM-HMM	0.59	0.28	0.38	0.53	0.51	0.52	0.72	0.47	0.57	0.75	0.47	0.58
Pattern*	-	-	-	-	-	-	0.74	0.62	0.67	0.77	0.63	0.69

[15] We used two CRF implementations from CRF++ (http://crfpp.googlecode.com/svn/trunk/doc/index.html) and CRFsuite [7] with Averaged Perceptron (AP) and Limited-memory BFGS (L-BFGS) training methods.

[16] http://opennlp.apache.org/.

[17] WoZ experiments participants indicated that 'not-providing' an answer was entertaining, giving wrong information, by contrast, was experienced as annoying.

6.2 Pattern Matching

Our pattern matching system handles 12 relations (See Table 6). These manually defined regular expressions seem to work well with certain relations. For example, regular expression like `born in (.*)` would match TIME_BIRTH or LOC_BIRTH relations. Subsequently, NER disambiguates between a DATE or GPE entities.

6.3 Post-processing Procedures

The process of extracting relations does not stop after the classifiers and pattern matching tools are applied. Certain post-processing is required in order to select the best result for each relation, e.g. based on confidence scores. This step also involves eliminating relations that do not link the person in question and chunk expansion.

Relations that are not concerned with the person in question were removed. For example:

(5) *Her mother, Kathy Hilton is* ***a former actress****, and her father, Richard Howard Hilton, is* ***a businessman****.*

In (5), the classifier marks *a former actress* and *a businessman* as the TITLE. However, this relation does not link the person in question, but her mother and father. In other words, we omitted the TITLE relation from the same sentence that contains CHILD_OF and PARENT_OF relations.

Table 5. CRF++ performance on System 3. P stands for precision; R for recall; F1 for harmonic mean.

Relation	P	R	F1	Relation	P	R	F1
ACCOMPLISHMENT	0.73	0.44	0.55	NATIONALITY	0.92	0.73	0.81
AGE_OF	0.95	0.76	0.84	OWNER_OF	0.76	0.40	0.48
AWARD	0.80	0.62	0.70	PARENT_OF	0.79	0.54	0.63
CHILD_OF	0.74	0.58	0.65	PART_IN	0.25	0.05	0.08
COLLEAGUE_OF	0.78	0.32	0.43	RELIGION	0.60	0.16	0.24
CREATOR_OF	0.64	0.17	0.26	SIBLING_OF	0.92	0.69	0.78
DURATION	0.97	0.64	0.76	SPOUSE_OF	0.76	0.42	0.52
EDUCATION_OF	0.84	0.65	0.72	SUBORDINATE_OF	0.81	0.19	0.31
EMPLOYEE_OF	0.77	0.19	0.28	SUPPORTEE_OF	1.00	0.40	0.54
FOUNDER_OF	0.65	0.26	0.36	MEMBER_OF	0.65	0.14	0.21
LOC	0.77	0.33	0.45	TIME	0.90	0.83	0.86
LOC_BIRTH	0.94	0.84	0.89	TIME_BIRTH	0.92	0.89	0.90
LOC_DEATH	0.90	0.55	0.67	TIME_DEATH	0.94	0.79	0.86
LOC_RESIDENCE	0.86	0.55	0.66	TITLE	0.84	0.66	0.74

There is also a special treatment for the TITLE relation which often requires chunk expansion when more information in form of complex possessive constructions is available. For example:

(6) *She later became* ***managing director of info service***.

The output from our classifier for (6) has *managing director* as TITLE, while the correct chunk is *managing director of info service*. Therefore, we expand the relevant chunk in order to cover the full NP with embedded NPs inside.

7 Experimental Setup and Results

In our 5-fold cross-validation classification experiments, classifiers were trained and evaluated in 3 different settings: (1) *System 1* where classification is based on automatically derived features such as n-grams for word and lemma (trigrams), POS, NER tags, chunking and capitalization; the joint classification on all relations was performed; (2) *System 2*: pattern matching and classification on the same features as System 1 applied for each relation separately; (3) *System 3*: the post-processed output of *System 2*.

Table 6. Pattern matching performance. P stands for precision; R for recall; F1 for harmonic mean.

Relation	P	R	F1	Relation	P	R	F1
AGE_OF	0.85	0.79	0.82	MEMBER_OF	0.46	0.43	0.42
CHILD_OF	0.87	0.87	0.87	PARENT_OF	0.86	0.78	0.82
DURATION	0.90	0.68	0.77	SIBLING_OF	0.93	0.85	0.88
EMPLOYEE_OF	0.53	0.16	0.23	SPOUSE_OF	0.79	0.63	0.70
FOUNDER_OF	0.74	0.71	0.72	SUBORDINATE_OF	0.72	0.61	0.65
LOC_DEATH	0.40	0.23	0.28	TIME_DEATH	0.29	0.23	0.26

All systems show the gains over the baseline systems. The later is obtained when training classifiers on *word token* features only. To indicate how good statistical classifiers generally are on relation recognition, consider the performance of distant supervision SVM[18] with precision of 53.3, recall of 21.8 and F-score of 30.9 (see [20]) on the TAC KBP relations. However, we emphasize that our task, relation set, application and data are different from those of TAC KBP. It would be useful in the future to test how well our proposed systems would behave on a different dataset.

As it can be observed from Table 4, the CRF++ classifier achieves the best results in terms of precision and F-score. Although the running time was not

[18] Distant supervision method is used when no labeled data is available, see [21].

measured, the classification runs faster comparing to SVM-HMM. System 2 outperforms the System 1 (6–11% increase in F-score). When training on each relation in isolation, features weights can be adjusted more efficiently not affecting other relations classification. Moreover, this allows assigning multiple relations to the same entity more accurately while avoiding high data sparseness opposed to training on complex multi-classs labels. *Key word* features have been observed as having the highest information gain. Pattern matching is proven to be a powerful and straightforward method, see Table 6.

While in general System 3 gains a small increase in F-score (around 0.6–2%) compared to System 2, it increases the precision for many relations. More detailed results from CRF++ on System 3 can be seen in Table 5.

8 Conclusions and Future Work

We proposed a data-oriented approach for question classification and answer extraction from unstructured textual data based on determining semantic relations between entities computing the Expected Answer Type. Our results showed that the relations that we have defined help the system to understand user's questions and to capture the information, which needs to be extracted from the data.

Having analysed misclassified EATs, we drew several conclusions. First, the classifier confuses semantically similar classes. Second, the classifier has difficulty to identify EATs for under-represented classes. Third, questions simultaneously belonging to several classes were often misclassified.

The easiest way to achieve a higher precision is probably to increase the number of instances for the under-represented classes. Of course, it is impossible to force the users to ask only certain types of questions. However, new instances can be generated based on the designed patterns using bootstrapping techniques and user's behaviour simulations.

Some of the relations were found using classification tools and not with pattern matching (and vice versa). In the future, both techniques should be combined. Observed inter-annotator agreement indicated that some relations need to be re-defined. Finally, we will test how generic the proposed approach is by testing it on the TAC and TREC datasets. Moreover, since some relations, in particular of RELATION(E$_1$, ?E$_2$) and RELATION(E, ?X) types, are very close to semantic roles, there is a need to analyse semantic role sets (e.g. ISO semantic roles [32]) and study the possible overlaps.

From the QADS development point of view, we need to evaluate the system in real settings. For this, question classifiers need to be re-trained on the actual and potentially erroneous ASR output. While testing/evaluating the QADS system, additional data will be produced, saved and used to enrich the training set.

Acknowledgments. The research reported in this paper was carried out within the DBOX Eureka project under number E! 7152.

References

1. Min, B., Li, X., Grishman, R., Ang, S.: New York University 2012 system for KBP slot filling. In: Proceedings of the 5th Text Analysis Conference (TAC 2012) (2012)
2. Roth, B., Chrupala, G., Wiegand, M., Singh, M., Klakow, D.: Saarland University spoken language systems at the slot filling task of TAC KBP 2012. In: Proceedings of the 5th Text Analysis Conference (TAC 2012), Gaithersburg, Maryland, USA (2012)
3. Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J., Nyberg, E., Prager, J., Schlaefer, N., Welty, C.: Building Watson: an overview of the DeepQA Project. AI Mag. **3**(31), 59–79 (2010)
4. Cortes, C., Vapnik, V.: Support vector networks. Mach. Learn. **20**(3), 273–297 (1995)
5. Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., Rus, V.: The structure and performance of an open-domain question answering system. In: Proceedings of the Association for Computational Linguistics, pp. 563–570 (2000)
6. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML 2001, pp. 282–289 (2001)
7. Okazaki, N.: CRFsuite: a fast implementation of Conditional Random Fields (CRFs) (2007). http://www.chokkan.org/software/crfsuite/
8. Ellis, J.: TAC KBP 2013 slot descriptions (2013). http://surdeanu.info/kbp2013/TAC_2013_KBP_Slot_Descriptions_1.0.pdf
9. Joachims, T., Finley, T., Yu, C.: Cutting-plane training of structural SVMs. Mach. Learn. **77**(1), 27–59 (2009)
10. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the COLING 2002, pp. 1–7. Association for Computational Linguistics (2002)
11. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of CoNLL 2009, pp. 147–155. Association for Computational Linguistics (2009)
12. Tjong Kim Sang, E., Buchholz, S.: Introduction to the CoNLL-2000 shared task: chunking. In: Proceedings of the 2nd Workshop on Learning Language in Logic and ConLL 2000, pp. 127–132. Association for Computational Linguistics (2000)
13. Finkel, J., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of ACL 2005, pp. 363–370. Association for Computational Linguistics (2005)
14. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of NAACL 2003, pp. 173–180. Association for Computational Linguistics (2003)
15. Chrupala, G., Klakow, D.: A named entity labeler for German: exploiting wikipedia and distributional clusters. In: Proceedings of LREC 2010, pp. 552–556. European Language Resources Association (ELRA) (2010)
16. Jackendoff, R.S.: Semantic Structures. MIT Press, Cambridge (1990)
17. ISO: Language resource management - Semantic annotation framework - Part 2: Dialogue acts. ISO 24617–2. ISO Central Secretariat, Geneva (2012)
18. Surdeanu, M.: Overview of the TAC2013 knowledge base population evaluation: English slot filling and temporal slot filling. In: Proceedings of the TAC KBP 2013 Workshop. National Institute of Standards and Technology (2013)

19. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**, 37–46 (1960)
20. Roth, B., Barth, T., Wiegand, M., Singh, M., Klakow, D.: Effective slot filling based on shallow distant supervision methods. In: Proceedings of the TAC KBP 2013 Workshop. National Institute of Standards and Technology (2013)
21. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint ACL/IJCNLP Conference, pp. 1003–1011 (2009)
22. Chernov, V., Petukhova, V., Klakow, D.: Linguistically motivated question classification. In: Proceedings of the 20th Nordic Conference on Computational Linguistics (NODALIDA), pp. 51–59 (2015)
23. Petukhova, V., Gropp, M., Klakow, D., Eigner, G., Topf, M., Srb, S., Moticek, P., Potard, B., Dines, J., Deroo, O., Egeler, R., Meinz, U., Liersch, S.: The DBOX corpus collection of spoken human-human and human-machine dialogues. In: Proceedings of the 9th Language Resources and Evaluation Conference (LREC) (2014)
24. Petukhova, V.: Understanding questions and finding answers: semantic relation annotation to compute the expected answer type. In: Proceedings of the Tenth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA 2010), Reykjavik, Iceland, pp. 44–52 (2014)
25. Heilman, M.: Automatic factual question generation from text. Ph.D. thesis, Carnegie Mellon University, USA (2011)
26. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
27. Kamp, H., Reyle, U.: From discourse to logic. Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. In: Studies in Linguistics and Philosophy, vol. 42. Kluwer, Dordrecht (1993)
28. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: an annotated corpus of semantic roles. Comput. Linguist. **31**(1), 71–106 (2002)
29. ICSI: FrameNet (2005). http://framenet.icsi.berkeley.edu
30. Petukhova, V., Bunt, H.: LIRICS semantic role annotation: design and evaluation of a set of data categories. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Paris. ELRA (2008)
31. Wiegand, M., Klakow, D.: Towards the detection of reliable food-health relationships. In: Proceedings of the NAACL-Workshop on Language Analysis in Social Media (NAACL-LASM), pp. 69–79 (2013)
32. Bunt, H., Palmer, M.: Conceptual and representational choices in defining an ISO standard for semantic role annotation. In: Proceedings of the Ninth Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9), Potsdam, pp. 41–50 (2013)

Exploiting Wikipedia-Based Information-Rich Taxonomy for Extracting Location, Creator and Membership Related Information for ConceptNet Expansion

Marek Krawczyk[1], Rafal Rzepka[2](✉), and Kenji Araki[2]

[1] Future Processing, ul. Bojkowska 37A, 44-100 Gliwice, Poland
mkrawczyk2@future-processing.com

[2] Hokkaido University, Kita-ku, Kita 14, Nishi 9, Sapporo, Japan
{rzepka,araki}@ist.hokudai.ac.jp

Abstract. In this paper we present a method for extracting IsA assertions (hyponymy relations), AtLocation assertions (informing of the location of an object or place), LocatedNear assertions (informing of neighboring locations), CreatedBy assertions (informing of the creator of an object) and MemberOf assertions (informing of group membership) automatically from Japanese Wikipedia XML dump files. We use the Hyponymy extraction tool v1.0, which analyses definition, category and hierarchy structures of Wikipedia articles to extract IsA assertions and produce information-rich taxonomy. From this taxonomy we extract additional information, in this case AtLocation, LocatedNear, CreatedBy and MemberOf types of assertions, using our original method. The presented experiments prove that both methods produce satisfactory results: we were able to acquire 5,866,680 IsA assertions with 96.0% reliability, 131,760 AtLocation assertion pairs with 93.5% reliability, 6,217 LocatedNear assertion pairs with 98.5% reliability, 270,230 CreatedBy assertion pairs with 78.5% reliability and 21,053 MemberOf assertions with 87.0% reliability. Our method surpassed the baseline system in terms of both precision and the number of acquired assertions.

Keywords: Common sense knowledge · Knowledge extraction
ConceptNet

1 Introduction

The effectiveness of systems dealing with textual-reasoning tasks depends on the scope of large-scale general knowledge bases they utilize. Just to enumerate few examples of such bases we could mention Cyc [1], YAGO [2] and ConceptNet [3]. In this paper we will focus on the last of the three - ConceptNet, a knowledge representation project that provides a large semantic graph describing general

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 262–274, 2018.
https://doi.org/10.1007/978-3-319-93782-3_19

human knowledge. ConceptNet was designed to contain knowledge collected by Open Mind Common Sense project's website [4]. Further releases incorporated knowledge from similar websites and online word games which automatically collect general knowledge in several languages. Current goal of ConceptNet is to expand the knowledge base with data mined from Wiktionary[1], a multilingual, web-based free content dictionary, and Wikipedia[2], a free-access, free content Internet encyclopedia. This open-source knowledge base is used for many applications such as topic-gisting [5], affect-sensing [6], dialog systems [7] and so on. Manual expansion of the knowledge base would be a long and labor-intensive process, as seen in nadya.jp[3], an online project aiming at gathering knowledge by using a game with a purpose [8]. Since its launch in 2010 it was able to introduce little over 43,500 entries to the ConceptNet. It is therefore evident that we need to employ automatic methods to gather new data.

Projects such as NELL [9] or KNEXT [10] aim at extracting semantic assertions from unstructured text data found on the Internet. Alternatively we could transfer information from the existing semi-structured sources into a knowledge base. As a considerable amount of human validation has already been involved in the process of creating such sources, the reliability of information gathered this way would be considerably higher. Wikipedia is probably the best example of open-source, large-scale information pools. Apart from previously mentioned YAGO, DBpedia project also aims at transferring knowledge gathered in Wikipedia into more formalized, digitally processable form [11]. English part of DBpedia has already been merged to ConceptNet, however the Japanese part has not been transferred yet, leaving this part of the knowledge base at the size of roughly 1/10 of the English language domain. The problem with using DBpedia repository is that the information gathering algorithms used to prepare the knowledge base were designed for multilingual input processing and therefore introduce a considerable amount of noise. As the knowledge gathered in ConceptNet is in considerable proportion language-specific, it is vital to widen the scope of Japanese part independently.

The current paper elaborates on efforts of [12]. We extended the scope of acquired assertions as well as explored possibilities of deriving commonsense knowledge from instance related information triplets.

2 Hyponymy Relation as IsA Relation

In our approach we use the Hyponymy extraction tool v1.0[4], an open-source program for extracting hyponymy relation pairs from Wikipedia's XML dump files. The tool has been developed specifically to process Japanese language entries. It consists of four modules, three of which deal with extraction of hyponymy pairs from different parts of Wikipedia content: definition, category and hierarchy

[1] http://www.wiktionary.org/.

[2] http://www.wikipedia.org/.

[3] http://nadya.jp/.

[4] http://alaginrc.nict.go.jp/hyponymy/.

structures [13]. The program utilizes Pecco library[5] (SVM-like machine learning tool) to assess the plausibility level of the extracted hyponymy relation pairs and boost the precision and recall of the system [14]. The extracted hyponymy pairs may be transferred to ConceptNet as two concepts related to each other by IsA relationship (Table 1 lists examples of the extracted pairs). According to [15] these pairs are not informative enough to be useful for NLP tasks such as Question Answering, however they do fall into the scope of ConceptNet, a domain representing commonsense and general knowledge. They are simple enough not to interfere with the ConceptNet's usage flexibility, yet informative enough to introduce new and valuable input to the knowledge base.

Table 1. Examples of extracted 'IsA' relationship pairs.

Hypernym	Hyponym
kouen. (park)	*Motomiya-kouen* (Motomiya Park)
kougu (tool)	*baisu* (vice)
Werudaa Bureemen-no senshu (Werder Bremen player)	Klaus Allofs
Nihon-no SF shousetsu (Japanese SF novel)	*Maikai Suikoden* (Hell's Water Margin)

[a] All Japanese language phrases are transliterated and written in italics.

3 Extracting Other Relations

The fourth module of the Hyponymy extraction tool v1.0 generates intermediate concepts of hyponymy relations using the output of the first three modules [15]. The tool executes the following procedure: first it acquires basic hyponymy relations from Wikipedia using the method proposed by [14]. Next, it augments each acquired hypernym with the title of the Wikipedia article from which the basic hyponymy relation was extracted and consolidates the basic hypernym with the newly generated augmented hypernym (so called 'T-INTER'). Finally it generates additional intermediate concept ('G-INTER') by generalizing the enriched hypernym. As a result, it acquires four-level, information-rich hyponymy relations.

Examples of augmented hyponymy relations include: *tojo-jinbutsu* (character) – *SF eiga no tojo-jinbutsu* (character of SF movie) – *WALL-E no tojo-jinbutsu* (character of WALL-E) – M.O; *seihin* (product) – *kigyo no seihin* (product of a company) – *Silicon Graphics no seihin* (product of Silicon Graphics, Inc.) – IRIS Crimson; *sakuhin* (work) – *America no shosestu-ka no sakuhin* (work of American novelist) – *J.D. Salinger no sakuhin* (work of J.D. Salinger) – A boy in

[5] http://www.tkl.iis.u-tokyo.ac.jp/~ynaga/pecco/.

France; *machi* (town) – *England no shu no machi* (town in a county in England) – *East Sussex no machi* (town in East Sussex) – Uckfield. As we can see from the examples, the generated augmented hypernyms are too specific to be incorporated into ConceptNet directly. However some additional information about their corresponding hyponyms may be extracted from them, such as information concerning location, neighboring locations, creator, membership and so on. Knowledge about location, creator and membership may be directly transferred into ConceptNet through already built-in AtLocation, LocatedNear, CreatedBy and MemberOf relations. It should be noted that according to the ConceptNet documentation[6] CreatedBy relation relates to processes, however inspection of the existing CreatedBy assertions show that they include creations and their authors as well. The remaining part of the acquired information related to the hyponyms may be represented by a more general RelatedTo relation.

The procedure of acquiring additional information is presented in Fig. 1 and exemplified in Fig. 2. First (Step 1), we scan the G-INTER using our handcrafted primary rule base in search of tags referring to locations, creators or members, for example {city}[7], {district}, {cartoonist}, {writer}, {member} and so on. In the case of acquiring LocatedNear pairs, we confirm that the basic hypernym contains a marker indicating physical proximity (such as the Chinese character meaning ‘neighboring’). Next (Step 2), we filter the basic hypernym through a secondary rule base to exclude items that would introduce noise. For example, we can extract information about the birthplaces of famous people; however this does not mean that we can build an AtLocation kind of relationship between the person and his or her birthplace. If so, hypernyms indicating people are excluded from the analysis of location. When analysing LocatedNear pairs we filter out ambiguous items. If the basic hypernym is positively assessed by the secondary rule base, then (Step 3) we assume that the phrase acquired by deleting the basic hypernym from the G-INTER is a valid location, creator or member tag. Using the example from Fig. 2, we check that ‘adjacent municipality’ is a valid tag to describe a nearby location. In the next stage (Step 4) we compare the validated location, creator or member tag with the content of the T-INTER. This way, using the previous example, we can extract the knowledge that the municipality we refer to is *Tomi-shi*. Finally (Step 5), we join the newly acquired information to the base hyponym with a proper relationship tag to extract a new relation, for example *Komoro-shi*-LocatedNear-*Tomi-shi*.

The effectiveness of the method mainly depends on the number and nature of introduced rules to both primary and secondary rules base. Our method is still work in progress and at this stage we used 58 primary rules and 16 secondary rules, which allowed us to extract assertions concerning location, neighboring locations and creators. The manually crafted rules have been created using heuristics after the analysis of the input data. The reason why we chose this kind of approach is because the information units contain Chinese characters indicating a type of location, a city, province, school or a creator. We use

[6] https://github.com/commonsense/conceptnet5/wiki/Relations.

[7] Curly brackets were used to mark the tags’ representations.

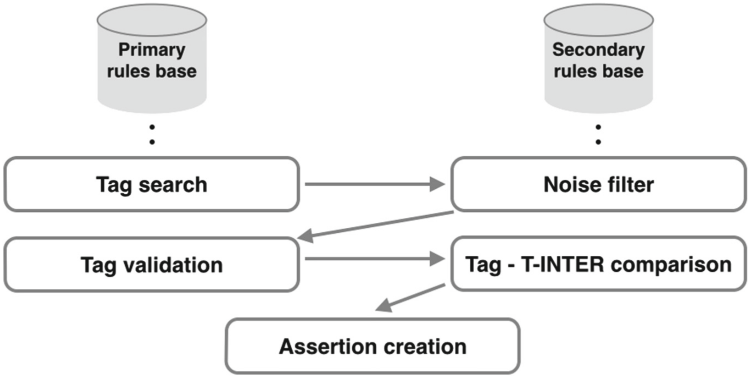


Fig. 1. Flowchart of our proposed method.

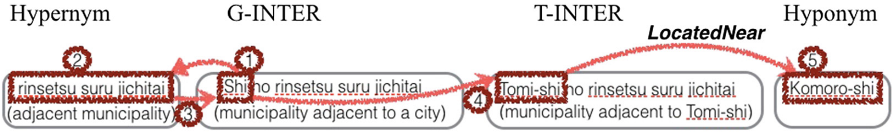

Hypernym
G-INTER
T-INTER
Hyponym
LocatedNear
1
2
3
4
5
rinsetsu suru jichitai
(adjacent municipality)
Shi no rinsetsu suru jichitai
(municipality adjacent to a city)
Tomi-shi no rinsetsu suru jichitai
(municipality adjacent to Tomi-shi)
Komoro-shi

Fig. 2. Procedure of our proposed method exemplified on the extracted relation.

the rules to detect these characters, and this way we are able to get the named entities referring to locations and creators. Because of the qualities of Japanese language writing system these rules are often very simple, containing a single character, but still effective for detecting language units we want to extract. For example secondary rules used for detecting people include suffix '∼sha', which describes different professions. For English such shortcut would be harder to apply, and therefore person detection would require a much larger rules base covering a long list of names of professions and appropriate suffixes (like '∼er', '∼or' or '∼ist').

As our experiments revealed, extracting creator information is more complex and creates some challenges. While extracting location and member-related information, the introduced rules may be simple and straightforward. In the case of creators, the rules not only have to cover the qualities of the writing system, but also take into consideration the importance of particular roles while creating a given piece of work. For example our annotators indicated that a number of professionals taking part in the creation of films may not be considered as the creators of these films. Actors, actresses and voice actors, even if they make a great contribution to the work, should not be labeled as its creators. Further experiments have shown that similarly animators, animation directors, sound directors, and storyboard creators, according to the annotators, do not qualify to be included in the common sense CreatedBy assertions. The question whether all these roles should be indeed excluded from the creator category is open to discussion. If we changed our perspective and considered that not only one person or role is to be credited as the creator of a given piece of work, then we could assess some of these roles as correct in the CreatedBy assertions. The problem of different opinions on this matter would however remain. As the algorithm bases on keywords, it is unable to distinguish, for example, between director and sound director. Such distinction would be possible if we employed an additional, concept-based knowledge base.

In future we would like to investigate the possibility of combining heuristics with automated rule discovery methods in order to achieve higher precision and recall. The number and reliability level of the data acquired with our method is presented in the Evaluation section.

4 Evaluation

To verify the reliability level declared by Sumida [14] and evaluate our proposed method for obtaining additional relations we used the 2014-11-04 version of the Japanese Wikipedia dump data. We ran the definition, category and hierarchy modules of the Hyponymy extraction tool v1.0 at 93% precision rate using the biggest available training set, and obtained 6,014,194 hypernym-hyponym pairs. The number of unique hyponymy pairs was 5,866,680, which indicates that 147,514 pairs have been extracted by more than one module. The 93% reliability level declared by the authors of the method has been verified by three human annotators, whose task was to evaluate a sample of the data and decide whether the extracted pairs (a) represent a correct hyponymy relation, (b) represent related concepts, but not in a hyponymy relation, or (c) represent unrelated concepts. The annotators assigned 1, 0.5 and 0 points respectively to 300 randomly selected assertions. We decided to assign 0.5 points to related concepts as they may be used to create correct assertions (see Future Work section). If two or more annotators assessed an item as belonging to one category, their decision was regarded as the evaluation output. In cases where their decisions varied (which happened 10 times), the first author decided the score. The procedure follows a modified Sumida *et al.* [14] evaluation method.

Table 2 presents the evaluation results. 283 pairs were assessed as representing a correct hyponymy relation, 10 pairs as related concepts, but not in a hyponymy relation and 7 as unrelated concepts. This results in 96.0% precision value of the tested sample, which surpasses the 93% declared by Sumida *et al.* The level of overall agreement between annotators was 86.9%, and the Kappa value[8] was 0.80, which indicates that the annotation judgement was in substantial agreement [16].

Running the fourth ‘extended’ module of the Hyponymy extraction tool v1.0 on the same Wikipedia dump data resulted in obtaining 2,738,211 basic hypernym–G-INTER–T-INTER–basic hyponym sets. By applying our method for extracting additional information, we were able to produce 131,760 pairs

[8] To measure the agreement level between judges, we used Randolph’s free marginal multirater kappa instead of Fleiss’ fixed-marginal multirater kappa, due to high agreement low kappa paradox.

Table 2. Evaluation results for IsA relations.

Correct hyponymy	Related concepts	Unrelated concepts	Precision	Total number of pairs
0.943 (283/300)	0.033 (10/300)	0.023 (7/300)	0.960	5,866,680

representing AtLocation relation, 6,217 pairs representing LocatedNear relation, 270,230 pairs representing CreatedBy relation and 21,053 pairs representing MemberOf relation. For comparison, nadya.jp, the baseline system, has provided only 8,706 AtLocation relations and no LocatedNear, CreatedBy or MemberOf relations in four years of its operation. In the case of AtLocation pairs, we evaluated 100 pairs[9] randomly selected from our method's output and 100 pairs randomly selected from nadya.jp's AtLocation assertions [8]. While evaluating LocatedNear, CreatedBy and MemberOf relations, a comparison with the baseline was not possible, as ConceptNet 5.3 does not yet contain any LocatedNear, CreatedBy or MemberOf pairs in its Japanese language section. These assertions were therefore evaluated independently. The evaluation procedure follows the previously applied one: 1 point being applied to correct AtLocation, LocatedNear, CreatedBy or MemberOf assertions, 0.5 point to related concepts, but not in the evaluated relation, and 0 points to unrelated concepts. In 15 cases the annotators' evaluation was inconsistent, and therefore the first author decided the score.

Table 3 shows the evaluation results of our AtLocation pairs generation method in comparison with the baseline system. 88 pairs generated by our method were evaluated as representing a correct AtLocation relation, 11 pairs as related concepts, but not in an AtLocation relation, and 1 as unrelated concepts. This results in a 93.5% precision value. In the case of the baseline system, 64 pairs were evaluated as correct AtLocation assertions, 20 as related concepts, but not in an AtLocation relation, and 16 as unrelated concepts. The precision value for the baseline system is 74.0%. The level of overall agreement between

Table 3. Evaluation results for AtLocation relations in comparison with the nadya.jp baseline.

	Correct AtLocation	Related concepts	Unrelated concepts	Precision	Total number of pairs
Proposed	0.880 (88/100)	0.110 (11/100)	0.010 (1/100)	0.935	131,760
Baseline	0.640 (64/100)	0.200 (20/100)	0.160 (16/100)	0.740	8,706

$p < 0.001$, t-score = 4.6291

[9] We adjusted the number of evaluated pairs to balance the proportion between the total number of pairs and the test sample.

annotators was 73.6% and the Kappa value was 0.60, which indicates that the annotation judgment was in moderate agreement. Examples of the extracted AtLocation assertions are presented in Table 4.

Table 4. Examples of generated AtLocation assertions.

Tomato Ginkou (Tomato Bank)	AtLocation	*Okayama-shi* (Okayama city)
Outao hoikuen (Outao nursery)	AtLocation	*Sakai-shi* (Sakai city)
Sandifukku (Sandy Hook)	AtLocation	*Eriotto-gun* (Elliott County)
Hoteru Kadoya (Kadoya Hotel)	AtLocation	Tochigi-shi (Tochigi city)

Table 5 contains the evaluation result of the generated LocatedNear relations. 97 pairs were evaluated as correct LocatedNear pairs, 3 as related concepts and none as unrelated concepts, which results in 98.5% precision. The level of overall agreement between annotators was 86.6% and the Kappa value was 0.80, which indicates that the annotation judgment was in substantial agreement. Examples of the extracted LocatedNear assertions are presented in Table 6.

Table 7 contains the evaluation result of the generated CreatedBy relations. 60 pairs were evaluated as correct CreatedBy pairs, 37 as related concepts and 3 as unrelated concepts, which results in 78.5% precision. The level of overall agreement between annotators was 71.6% and the Kappa value was 0.57, which indicates that the annotation judgment was in moderate agreement. Examples of the extracted CreatedBy assertions are presented in Table 8.

Table 5. Evaluation results for LocatedNear relations

Correct LocatedNear	Related concepts	Unrelated concepts	Precision	Total number of pairs
0.970 (97/100)	0.030 (3/100)	0.000 (0/100)	0.985	6,217

The analysis of the relatively low precision score of the assessed CreatedBy assertions revealed the following: in 24 cases it was the annotators' opinion that actors, voice actors, animators, storyboard creators or sound directors cannot be considered as creators of works they contribute to. Although it would be valid to include such persons in the RelatedTo kind of relationship with the work they helped to create, defining them as creators would go against common sense. This is a valid observation and it will be taken into consideration when re-designing

Table 6. Examples of generated LocatedNear assertions.

Ougoe-machi (Ougoe city)	LocatedNear	*Ono-machi* (Ono city)
Iseri-gawa (Iseri river)	LocatedNear	*Konoha-gawa* Konoha river
Daiting	LocatedNear	Monheim
Kumotori-yama (Mount Kumotori)	LocatedNear	*Karamatsuo-yama* (Mount Karamatsuo)

Table 7. Evaluation results for CreatedBy relations.

Correct CreatedBy	Related concepts	Unrelated concepts	Precision	Total number of pairs
0.600 (60/100)	0.370 (37/100)	0.030 (3/100)	0.785	270,230

Table 8. Examples of generated CreatedBy assertions.

Dark Horse	CreatedBy	George Harrison
Kaze (Wind)	CreatedBy	*Kubota Koutarou*
Manuke-na Oukami (Sheep Wrecked)	CreatedBy	Michael Lah
The Point of View	CreatedBy	Alan Crosland

and expanding the rule base for the next version of the algorithm. There were also cases of assertions assessed as invalid due to errors passed from the output of the Hyponymy extraction tool to the proposed method. Table 9 contains examples of assertions that were assessed as erroneous by the annotators.

Table 9. Examples of erroneous CreatedBy assertions.

Road 88	CreatedBy	*Tomita Yasuko* (actress)
Kaiketsu Zorori (Incredible Zorori)	CreatedBy	*Yamada Etsuji* (sound director)
Kishin Douji Zenki (Zenki)	CreatedBy	*Hayashi Akemi* (animator)
Human (incomplete name error)	CreatedBy	Nicholson Baker

Table 10 contains the evaluation result of the generated MemberOf relations. 76 pairs were evaluated as correct MemberOf pairs, 22 as related concepts and 2 as unrelated concepts, which results in 87.0% precision. The level of overall agreement between annotators was 80.6% and the Kappa value was 0.71, which indicates that the annotation judgment was in substantial agreement. Examples of the extracted MemberOf assertions are presented in Table 11.

Table 10. Evaluation results for MemberOf relations.

Correct MemberOf	Related concepts	Unrelated concepts	Precision	Total number of pairs
0.760 (76/100)	0.220 (22/100)	0.020 (2/100)	0.870	21,053

Table 11. Examples of generated MemberOf assertions.

Henning Schmitz	MemberOf	*Kurafutowaaku* (Kraftwerk)
Dir.F	MemberOf	*Suiyoubi no Kanpanera* (Wednesday Canpanella)
Oono Satoshi	MemberOf	*Arashi*
Nishimura Akihiro	MemberOf	*Nikkan Giin Renmei* (Japan-Korea Parliamentarians' Union)

In the 13 cases the annotators decided that the generated MemberOf assertion refer to the former member of relative group, and therefore assigned it as the related concepts. The question whether these pairs should be considered as representing concepts in MemberOf relation is currently under discussion. If we would consider that the status of a member, once granted, is not temporary, then the precision rate of the tested sample would be higher, reaching 93.5%.

The results show that IsA relation pairs generated by the definition, category and hierarchy of the Hyponymy extraction tool v1.0, as well as AtLocation, LocatedNear and MemberOf relation pairs extracted by our proposed method may be incorporated into ConceptNet. Considering the number of the newly acquired assertions as well as reliability of the data in comparison with the resources already present in the knowledge base, such operation would be beneficial for ConceptNet. CreatedBy relation pairs could also be added after the revision of introduced rules and a substantial increase of the precision rate.

5 Generalizing over Assertions

Wikipedia contains a lot of information about instances of certain concepts, such as Salvador Dali as an instance of a painter. Filling up ConceptNet with instances is a valid task, as it is very hard to establish the boundaries of commonsense knowledge - facts obvious for one group of people in large proportion overlap with knowledge of another group, but there is always a discrepancy. This issue raises a question: would it be possible to come to more general conclusions on the basis of the numerous instances? In order to solve this problem we created and performed an initial test of the following method: we took each of the additional information lists (representing LocatedAt, LocatedNear and CreatedBy relations) and analyzed each assertion one by one. For both concepts in the assertion we found their hypernyms in the generated IsA relations list. Next we generated assertions representing all possible combinations between concept's A hypernyms and concept's B hypernyms. We have repeated the process for all assertions in the additional information list and calculated the generated hypernym assertions' occurrence frequency. As predicted the assertions with the highest occurrence frequency represent general, commonsense observations. This is true for AtLocation and CreatedBy lists, but it is not the case when processing the LocatedNear list because of the relatively low number of LocatedNear assertions. It became apparent that the higher number of initial assertions increases the probability of generating meaningful general assertions. See Table 12 for the examples of generated general assertions. The procedure requires further development in terms of the method of frequency calculations and automatic filtering of non-general assertions.

Table 12. Examples of generated general assertions.

toshi oyobi machi (city and town)	AtLocation	*gun* (province)
shougakkou (elementary school)	AtLocation	*machi* (city)
douro (road)	AtLocation	*machi* (city)
sakuhin (work)	CreatedBy	*zonmei jinbutsu* (living person)
anime sakuhin (anime)	CreatedBy	*anime kankeisha* (people involved in making anime)
shutsuen sakuhin (performance art)	CreatedBy	*bunkajin* (cultural figure)

6 Conclusion

In this paper we presented a method for automatic acquisition of common sense knowledge triplets from the Japanese Wikipedia. It allowed us to mine IsA, AtLocation, LocatedNear, CreatedBy and MemberOf assertions with precision estimated at the levels of 96.0%, 93.5%, 98.5%, 78.5% and 87.0% respectively.

We also demonstrated the possibility of formulating common sense assertions on the basis of generated instances data. As the Japanese part of the current ConceptNet 5.3 consists of 1,071,046 assertions, a contribution of 6,295,940 new assertions would be significant. It would mean an almost sixfold increase and could potentially make ConceptNet applicable to many Japanese language analysis problems. Moreover, as Wikipedia is a constantly expanding source, we could acquire more assertions simply by applying our method to the updated Wikipedia XML dump files.

7 Future Work

In order to extend the functionality of our proposed method, we intend to update the primary and secondary rules, which would allow the system to increase its precision and the scope of extracted information. We would also like to explore the possibility of using a machine learning algorithm for automatic rule generation combined with the already present heuristics. Such a combination could potentially be more effective in increasing precision and recall, as well as finding new rules to extract even more relations.

We also plan to create an interface for the evaluation of the method's output by Japanese native speakers, which would allow us to utilize the pairs representing related concepts.

References

1. Lenat, D.B.: CYC: a large-scale investment in knowledge infrastructure. Commun. ACM **38**(11), 33–38 (1995)
2. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706 (2007)
3. Liu, H., Singh, P.: ConceptNet? A practical commonsense reasoning tool-kit. BT Technol. J. **22**(4), 211–226 (2004)
4. Singh, P., Lin, T., Mueller, E.T., Lim, G., Perkins, T., Zhu, W.L.: Open mind common sense: knowledge acquisition from the general public. In: On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE, pp. 1223–1237 (2002)
5. Speer, R.H., Havasi, C., Treadway, K.N., Lieberman, H.: Finding your way in a multi-dimensional semantic space with Luminoso. In: Proceedings of the 15th International Conference on Intelligent User Interfaces, pp. 385–388 (2010)
6. Cambria, E., Hussain, A., Havasi, C., Eckl, C.: SenticSpace: visualizing opinions and sentiments in a multi-dimensional vector space. In: Knowledge-Based and Intelligent Information and Engineering Systems, pp. 385–393 (2010)
7. Korner, S.J., Brumm, T.: RESI - a natural language specification improver. In: IEEE International Conference on Semantic Computing, pp. 1–8 (2009)
8. Nakahara, K., Yamada, S.: Development and evaluation of a web-based game for common-sense knowledge acquisition in Japan. Unisys Technol. Rev. **107**, 295–305 (2011)

9. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka Jr., E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: AAAI, vol. 5, p. 3 (2010)
10. Schubert, L.: Can we derive general world knowledge from texts? In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 94–97 (2002)
11. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, pp. 1–8 (2011)
12. Krawczyk, M., Rzepka, R., Araki, K.: Extracting ConceptNet knowledge triplets from Japanese Wikipedia. In: Proceedings of the 21st Annual Meeting of the Association for Natural Language Processing, pp. 1052–1055 (2015)
13. Sumida, A., Torisawa, K.: Hacking Wikipedia for hyponymy relation acquisition. In: IJCNLP, vol. 8, pp. 883–888 (2008)
14. Sumida, A., Yoshinaga, N., Torisawa, K.: Boosting precision and recall of hyponymy relation acquisition from hierarchical layouts in Wikipedia. In: LREC (2008)
15. Yamada, I., Hashimoto, C., Oh, J., Torisawa, K., Kuroda, K., De Saeger, S., Tsuchida, M., Kazama, J.: Generating information-rich taxonomy from Wikipedia. In: 4th International Universal Communication Symposium (IUCS), pp. 97–104 (2010)
16. Randolph, J.J.: Free-Marginal Multirater Kappa (multirater K [free]): An Alternative to Fleiss' Fixed-Marginal Multirater Kappa (2005). Online Submission

Text Engineering and Processing

Lexical Analysis of Serbian with Conditional Random Fields and Large-Coverage Finite-State Resources

Mathieu Constant[1](✉), Cvetana Krstev[2], and Duško Vitas[2]

[1] ATILF UMR 7118, Université de Lorraine/CNRS, Nancy, France
Mathieu.Constant@univ-lorraine.fr
[2] University of Belgrade, Belgrade, Serbia

Abstract. This article describes a joint approach to lexical tagging in Serbian, combining three fundamental natural language processing tasks: part-of-speech tagging, compound and named entity recognition. The proposed system relies on conditional random fields that are trained from a newly released annotated corpus and finite-state lexical resources used in an existing symbolic Serbian tagging system. Experimental results show that a joint strategy is more robust than pipeline ones and that the use of lexical resources has a significant positive impact on tagging, in particular on out-of-domain texts.

1 Introduction

Lexical tagging is a key preprocessing stage for Natural Language Processing (NLP) applications as it maps a sequence of tokens into a sequence of tagged lexical units. Lexical units are semantic units that can be made of several tokens like multiword expressions. In this article, we present methods to integrate three fundamental NLP tasks related to lexical tagging and applied to Serbian: part-of-speech (POS) tagging, Named Entity (NE) recognition and compound[1] recognition.

Given a sequence of tokens in Serbian, our goal is to provide a tagged sequence of lexical units, a lexical unit being either a simple word, a compound or a multiword named entity. For instance,

- **Input:** Sutradan, 22. oktobra, na pitanje ser Fransisa Kromertija, Paspartu pogledavši svoj sat odgovori da je tri časa izjutra. I zaista, ovaj slavni sat, još udešen po griničkom meridijanu, koji je sada bio nekih sedamdeset i sedam stepeni na zapadu, morao je kasniti i kasnio je stvarno četiri časa.
  (The following day, 22 October, Passepartout consulted his watch in response

[1] For the purpose of this article, we define a compound as a contiguous sequence of tokens that has a non-compositional meaning. Compounds form a subclass of multiword expressions. We exclude multiword named entities from it.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 277–289, 2018.
https://doi.org/10.1007/978-3-319-93782-3_20

to a question from Sir Francis Cromarty, and replied that it was three in the morning. And indeed this wonderful watch was, quite logically, four hours slow, being still set to the Greenwich meridian, nearly 77° further west.)
- **Output:** Sutradan,_22._oktobra/NE ,/PONCT na/PREP pitanje/N ser/N Fransisa/X Kromertija/NE ,/PONCT Paspartu/ pogledavši/V svoj/PRO sat/N odgovori/V da/CONJ je/V tri_časa/NE izjutra/NE ./PONCT I/PAR zaista/ADV ,/PONCT ovaj/PRO slavni/A sat/N ,/PONCT još/ADV udešen/A po/PREP griničkom_meridijanu/N ,/PONCT koji/PRO je/V sada/ADV bio/V nekih_sedamdeset_i_sedam_stepeni/NE na/PREP zapadu/N ,/PONCT morao/V je/V kasniti/V i/CONJ kasnio/V je/V stvarno/ADV četiri_časa/NE ./PONCT

For example, *griničkom_meridijanu/N* annotates as a noun (N) the lexical unit *grinički meridijan* composed of the two tokens *grinički* and *meridijan*. The character _ delimits tokens in a multiword unit. Named entities are tagged with the symbol NE. Other symbols stand for POS tags.

In this article, we experiment for Serbian a supervised hybrid strategy, proposed by [6], integrating information coming from lexical resources into a statistical model trained on a reference annotated corpus. In particular, we integrate large-scale finite-state resources made of e-dictionaries and local grammars into linear-chain Conditional Random Fields (CRF), and we evaluate the produced system on out-of-domain texts.

The contributions of the article are the following:

- Release of new datasets for Serbian, with fine-grained annotations of NE and compounds, as well as POS tags, that were used for training and evaluating our system;
- Experimental comparisons between different orchestration strategies for the integration of the three tasks;
- Development of a new lexical tagger for Serbian.

2 Background

2.1 Compound and NE Recognition and POS Tagging

Our paper focuses on the combination of three fundamental tasks of NLP: POS tagging, compound recognition (CR) and Named Entity Recognition (NER). POS tagging and NER have been widely studied in the literature, while CR is getting growing interest in the community [2,6,25], but research on their combination is much less common. NER and POS tagging (as well as CR and POS tagging) are traditionally combined, as POS tagging is very frequently used to provide linguistic information to NER and CR in the form of features in statistical approaches, as shown in the data provided for the CONLL-2003 shared task on NER [22] and for the PARSEME shared task on the identification of verbal multiword expressions [19]. CR and NER can also be combined in different ways as shown in [25]: either CR is informed by NER, or NER is informed by

CR. Some have implemented limited joint strategies:[2] for instance, joint CR and POS tagging [4,21]. Furthermore, many studies have shown that statistical models could be efficiently trained by combining annotated corpora and pieces of information coming from linguistic/knowledge databases (ex. lexicons, gazeeters) for POS tagging [8], NER,[17], or CR [5].

Statistical approaches for POS tagging, NER and CR mainly rely on sequential models such as maximum entropy, conditional random fields, support vector machine. In addition, nowadays deep learning revolution has revived the interest in such traditional tasks that are more and more modeled with neural networks [3,15]. In this article, we do not rely on neural approaches as the data used is small, and such approaches are not known to perform well on small data. Instead, we use conditional random fields [14] that have been shown to perform well for our three tasks [4,17,25]. A neural approach could be considered for future work.

2.2 Lexical Tagging in Serbian

The lexical tagger presented in this paper builds on an existing Serbian symbolic system implemented via the Unitex platform [26]. This system relies on large-coverage and fine-grained e-dictionaries of simple and compound words, as well as on local grammars (cf. next section). In particular, the NE hierarchy in Serbian NER system consists of five top-level types: persons, organizations, locations, amounts, and temporal expressions, each of them having one or more levels of sub-types. For instance, locations have sub-types: hydronyms, oronyms, regions, cities, etc. The tagging strategy allows nesting, which means that a named entity can be nested within another named entity, e.g. an organization name can be nested within a person's role (or function) which is nested within a personal NE, like in the following example:

```
<pers>
  <role>
    knjizevnik i predsednik
    <org>Upravnog odbora</org>
    <org>Narodne biblioteke
      <top.dr>Srbije</top.dr>
    </org>
  </role>
  <persName.full>
    Mihailo Pantic
  </persName.full>
</pers>
```

(Writer and chairman of the Board of the National Library of Serbia Mihailo Pantić). However, for the purpose of our experiment we have kept only the NEs with the longest span.

[2] Note that these tasks can also be jointly combined with parsing: e.g. CR [6,10,18] or NER [9].

The Serbian NER system is a handcrafted rule-based system that relies on comprehensive lexical resources for Serbian described in Subsect. 3.1. For recognition of some types of named entities, e.g. personal names and locations, e-dictionaries and information within them is crucial; for others, like temporal expressions, local grammars in the form of Finite-State Transducers (FST) that try to capture a variety of syntactic forms in which a NE can occur had to be developed. However, for all of them local grammars were developed that use wider context to disambiguate ambiguous occurrences as much as possible. These local grammars were organized in cascades that further resolve ambiguities [16]. NER system was evaluated on a newspaper corpus and results reported in [12] showed that F-measure of recognition was 0.96 for types and 0.92 for tokens.

Note that the Serbian system we are relying on is not the only tagging system available for Serbian. For instance, [1] have experimented different statistical models and configurations for POS tagging and lemmatization. A TreeTagger was also used to POS tagging of the Corpus of Contemporary Serbian [24]. A POS semi-automatic tagger aimed for speech technologies that annotates accentuation as well is presented in [20]. None of them, however, takes into account compounds and/or named entities.

3 Data

3.1 Lexical Resources

The resources for NLP of Serbian consist of electronic dictionaries and local grammars. They are being developed using the finite-state methodology [7,11]. The role of e-dictionaries, covering both simple words and compounds, and dictionary finite-state transducers (FSTs) is text tagging. Each e-dictionary of forms consists of a list of entries supplied with their lemmas, morphosyntactic, semantic and other information. The forms are, as a rule, automatically generated from the dictionaries of lemmas containing the information that enables production of forms. Compounds are assigned the same POS as simple words; however, compound verbs are not covered yet. The system of Serbian e-dictionaries covers both general lexica and proper names and all inflected forms are generated from 135,000 simple forms and 13,000 compound lemmas. Approximately 28.5% of these lemmas represent proper names: personal, geopolitical, organizational, etc.

Dictionary FSTs are used for recognition and tagging of some open classes of compounds, multiword numerals written with digits, words and their combinations (e.g. *10 milijardi i 135 miliona* ‘10 billions and 135 millions’), and multiword nouns, adjectives and adverbs derived from numerals and written with digits (e.g. adjective *14-karatno (zlato)* ‘14-carat (gold)’), interjections with freely repeating parts (e.g. *Eeeeej* and *A-ha*), etc. [13]. The output format of these FSTs follows exactly the format of e-dictionaries; thus, from the recognized sequence an e-dictionary entry is formed and added to the used e-dictionaries. For instance, if the recognized sequence is the form *14-karatnog*, a dictionary FST produces a dictionary entry *18-karatnog, 18-karatni.A:adns2* which gives the form's lemma *18-karatni*, its POS *A* (adjective), and a set of morphosyntactic categories *adns2* – positive (a), definite (d), neuter gender (n), singular (s), genitive (2).

3.2 Annotated Corpus

For the experiment we used two texts: one for training and development and another one for testing. For training and development we used the Serbian translation of Verne's novel "Around the World in Eighty Days". The text was analyzed using Serbian lexical resources presented in previous sections in Unitex system.[3] The annotated text was prepared in two steps. First, the text was analyzed with e-dictionaries of simple words and then manually disambiguated [23]. In the next step, the text was analyzed with remaining resources (e-dictionaries of compounds, dictionary FSTs and NER system), and results were manually disambiguated and corrected where necessary. For instance, in our example *griničkom meridijanu* 'Greenwich meridian' was annotated by e-dictionary of compounds, *sedamdeset i sedam* 'seventy seven' was annotated by dictionary FSTs and *Sutradan, 22. oktobra* 'The following day, 22 October' was annotated by NER. Finally, both texts were automatically merged into one. The resulting text uses annotation codes applied in the Serbian system of e-dictionaries.

Table 1. The size of the training and testing texts. Tokens comprise words and punctuation marks. NEs can be both simple words and multiword units.

	Tokens	Simple	Compounds	NE
Verne	64,829	51,845	3,054	3,036
Švejk	2,953	3,104	108	192
Floods	4,272	3,232	237	395
History	5,193	4,859	471	531
Test	13,418	11,195	816	1,118

For testing, we prepared another text that comprises parts coming from three different sources: (i) the first chapter of the novel "The Good Soldier Švejk" (translation to Serbian) (referred to as *Švejk*); (ii) a few news articles dealing with floods in Serbia in 2014 (referred to as *Floods*); (iii) a few chapters of the History manual for elementary schools (referred to as *History*). First, the text was processed by e-dictionaries of simple words and compounds, dictionary FSTs, and at the end NER was applied. In the next step, all NE tags were manually checked and corrected. Finally, POS tags and lemmas of all simple words and compounds were manually disambiguated and necessary corrections

[3] The Unitex software system: http://unitexgramlab.org/.

Table 2. New entries added to e-dictionaries during text processing.

	Simple	Compounds
Verne	294	143
Švejk	16	1
Floods	36	33
History	8	36
Test	60	70

were done (e.g. missing tags for words not covered by e-dictionaries were added).[4] The size of the training and testing texts are presented in Table 1.

Processing of training and testing texts revealed that some entries were missing in dictionaries and they were added to them for future use (see Table 2). Entries added from the training text were used during the training phase, while entries added from the testing texts were not used in the testing phase, in order not to bias the experimental results.

4 Approach

Given a sequence of tokens, our goal is to provide a tagged sequence of lexical units: a lexical unit being either a simple word, a compound or a multiword named entity; a tag being either a POS or a NE class. This involves the integration of three different tasks: POS tagging, NE recognition and compound recognition. Each of the three intended tasks are usually considered as sequential tagging tasks. Indeed, multiword NE tagging and compound recognition can be seen as segmentation tasks (like chunking). By using a IOB-like scheme, it is equivalent to labeling simple tokens. Each token is labeled by a tag in the form B-X or I-X, where X is the label of the lexical unit the token belongs to. Prefix B indicates that the token is at the beginning of the lexical unit. Prefix I indicates an internal position. label O indicates an element that corresponds to a simple word.

The three different tasks on the same sentence should produce independently the first three annotations (columns NER, CR and POS) in Table 3. As depicted in Sect. 2, there are several possible orchestrations to reach our goal: either using a joint approach, or using a pipeline one. The joint approach consists in performing the three tasks in one step using a single sequential tagger (one model) by using a tagset, the labels of which combine the three annotations.[5] The corresponding output is provided in the last column of Table 3.

[4] The disambiguation was done by a special tool integrated into Unitex system that facilitates manual disambiguation http://tln.li.univ-tours.fr/Tln_Colloques/Tln_JUnitex2014/Communications/Vitas.pdf.

[5] Note that it does not correspond to a strict combination of the three types of annotations, as we do not tag the internal elements of the multiword lexical units.

Table 3. The annotation of the presented example

token	NER	CR	POS	JOINT
Sutradan	B-NE	O	ADV	B-NE
22	I-NE	O	NUM	I-NE
.	I-NE	O	PONCT	I-NE
oktobra	I-NE	O	N	I-NE
na	O	O	PREP	B-PREP
pitanje	O	O	N	B-N
ser	O	O	N	B-N
Frensisa	O	O	X	B-X
Komertija	B-NE	O	N	B-NE
,	O	O	PONCT	B-PONCT
Paspartu	O	O	N	B-N
pogledavši	O	O	V	B-V
svoj	O	O	PRO	B-PRO
sat	O	O	N	B-N
odgovori	O	O	V	B-V
da	O	O	CONJ	B-CONJ
je	O	O	V	B-V
tri	B-NE	O	NUM	B-NE
časa	I-NE	O	N	I-NE
izjutra	B-NE	O	ADV	B-NE
.	O	O	PONCT	B-PONCT
I	O	O	CONJ	B-CONJ
zaista	O	O	ADV	B-ADV
,	O	O	PONCT	B-PONCT
ovaj	O	O	PRO	B-PRO
slavni	O	O	A	B-A
sat	O	O	N	B-N
,	O	O	PONCT	B-PONCT
još	O	O	ADV	B-ADV
udešen	O	O	V	B-V
po	O	O	PREP	B-PREP
griničkom	O	B-N	A	B-N
meridijanu	O	I-N	N	I-N
,	O	O	PONCT	B-PONCT
koji	O	O	PRO	B-PRO
je	O	O	V	B-V
sada	O	O	ADV	B-ADV
bio	O	O	V	B-V
nekih	B-NE	O	ADV	B-NE
sedamdeset	I-NE	O	NUM	I-NE
i	I-NE	O	CONJ	I-NE
sedam	I-NE	O	NUM	I-NE
stepeni	I-NE	O	N	I-NE
na	O	O	PREP	B-PREP
zapadu	O	O	ON	B-N
,	O	O	PONCT	B-PONCT
morao	O	O	V	B-V
je	O	O	V	B-V
kasniti	O	O	V	B-V
i	O	O	CONJ	B-CONJ
kasnio	O	O	V	B-V
je	O	O	V	B-V
stvarno	O	O	ADV	B-ADV
četiri	B-NE	O	NUM	B-NE
časa	I-NE	O	N	I-NE
	O	O	PONCT	B-PONCT

The pipeline approach consists in applying sequentially different tagging tasks. In particular, we tested two possibilities:

- POS → SEG: POS tagging is first performed on the token sequence, and the predicted POS are then provided as an input to a standalone compound/NE recognition system
- SEG → POS: A standalone compound/NE recognizer provides a sequence of lexical units as an input of a POS tagger.

For each module of these different orchestrations, we used linear-chain Conditional Random Fields (CRF). They are discriminative probabilistic models introduced in [14] for sequential labeling and have been shown to be very accurate for segmentation tasks.

5 Experiments

5.1 Setup

The various CRF models used in our experiments were trained on 80% of the Verne Corpus. The remaining 20% were used as development (dev) dataset (e.g. for feature tuning). As mentioned in Sect. 3.2, the test set is composed of the texts *Svejk*, *Floods* and *History*. We therefore performed out-of-domain evaluation in the sense that the dataset used for training/dev belong to a domain different from the one used for testing. The models were trained and tested with the software *lgtagger* [4] that allows easy incorporation of information coming from lexical resources into CRF in the form of features.

For our experiments, we set two parameters: (a) orchestration strategy; (b) use of lexicon-based features. Parameter (a) offers three possible values: one joint strategy and two pipeline ones. Parameter (b) is binary-valued (NO LEX or LEX). In the latter case, the lexicon-based features are computed as follows. We first applied the Serbian e-dictionaries and cascades of FSTs described in Sect. 3.1 on the whole corpus presented in 3.2, in order to create a single lexicon containing all recognized forms. *lgtagger* uses this lexicon to construct a preliminary "naive" segmentation to be used as a source of features (for more details, see [5]).

Table 4. Overall scores on DEV and TEST datasets. We provide two kinds of evaluation: (a) lexical segmentation alone (SEG); (b) segmentation + tagging (TAG).

		DEV (in-domain)					
		NO LEX			LEX		
		R	P	F	R	P	F
JOINT	SEG	98.37	97.69	98.03	99.33	99.09	99.21
	TAG	95.07	94.41	94.74	97.36	97.12	97.24
POS → SEG	SEG	98.75	97.53	98.13	99.49	99.09	99.29
	TAG	95.18	94.00	94.58	97.44	97.04	97.24
SEG → POS	SEG	98.35	97.21	97.77	99.45	98.94	99.19
	TAG	94.87	93.77	94.31	97.36	96.86	97.11
		TEST (out-of-domain)					
		NO LEX			LEX		
		R	P	F	R	P	F
JOINT	SEG	95.58	91.46	93.48	97.61	95.65	96.62
	TAG	85.86	82.15	83.96	91.28	89.44	90.35
POS → SEG	SEG	96.15	91.26	93.64	97.55	94.96	96.24
	TAG	86.15	81.76	83.90	91.06	88.64	89.83
SEG → POS	SEG	95.86	90.91	93.32	97.72	95.59	96.64
	TAG	85.99	81.55	83.71	90.71	88.73	89.71

5.2 Results

Experimental results on development and test datasets are given in Table 4. Results are evaluated with the standard F-score (F) that is the harmonic mean of precision (P) and recall (R). Whereas all strategies reach comparable scores on the lexical segmentation task alone, it appears that the joint strategy is more robust on the tagging task on out-of-domain texts (test dataset). The experimental difference between the two tagging approaches is statistically significant with p-value < 0.01 computed from χ^2 score. This strategy has also the advantage of being easy to implement (a single model to train and to apply), although it is slightly slower to train than the ones used in pipeline strategies. For instance, the model used in the best joint strategy is trained in 1229s, instead of 484s for the longer training in a pipeline strategy on the same machine (Intel(R) Xeon(R) CPU E5640 @ 2.67 GHz 8 core).

One can observe that the use of lexicon-based features greatly improves the accuracy of the lexical tagger, especially for out-of-domain texts: a gain of 6.5 pts in terms of F-score as compared with 2.5 pts for in-domain text (dev dataset). It also appears that lexical resources have a significant impact on precision first (+7 pts on the joint system) and then on recall (+4.5 pts).

6 Discussions

This section is devoted to go deeper in the analysis of the results in order to have a better understanding what really happens with the joint system. We first make an error analysis on the development set, in order to obtain the main kinds of errors produced by the system. We then discuss how good unknown units are tagged.

6.1 Error Analysis

There were 182 differences between the reference text and the output text. The differences can be described in the following way:

- **POS** is wrongly attributed. There were 107 differences of this kind (see Table 5). Adjectives, particles and adverbs had the most wrongly attributed POS (23, 22 and 19, respectively). Verbs and conjunctions were assigned wrongly in most of the cases (27 and 17, respectively). Prepositions were always correctly tagged; numerals were never wrongly assigned. The most confusions were between pairs: adjective/verb (19), particle/conjunction (17), noun/verb (13). Many cases of adjective/verb confusion come from the past participle of a verb and an adjective derived from it, e.g. *zatvoren* ‘close/-closed’.
- **NE recognition.** A simple word NE was not recognized (instead a correct POS was assigned), or a simple word was wrongly recognized as a NE. There were 18 differences of this kind. Example: a time NE *uveče* ‘in the evening’ was assigned a POS *ADV*. The second example: the noun *Mongolija* ‘Mongolia’ (the name of a ship) was recognized as a toponym.

Table 5. A POS confusion: values in a column show POS tags that were erroneously attributed to a chosen POS tag, values in a row show to what POS a chosen POS tag was erroneously attributed.

	A	N	V	ADV	NUM	PREP	PRO	PAR	CONJ	INT	Total
A		1	9	2			1				13
N	2		2	3	1						8
V	10	11		2			1	1		2	27
ADV	9	1			1		3		1		15
NUM											0
PREP				3	2						5
PRO	2			6				7			15
PAR									4		4
CONJ				3				13		1	17
INT								1			1
X		2									2
Total	23	15	11	19	4	0	5	22	5	3	107

- **NE type.** A NE type was wrongly attributed or it was not attributed at all. There were 25 differences of this kind. Example: a money NE *pedeset i pet hilxada livara* ‘fifty five thousand pounds’ was recognized as an amount NE. The most differences included time, amount, money and measure NEs.
- **NE span.** A NE span was not correctly established. There were 26 differences of this kind. Example: a time NE *devet časova i trideset i sedam minuta* ‘nine o’clock and thirty seven minutes’ was recognized as two separate time NEs: *devet časova* and *trideset i sedam minuta*.
- **compound recognition** compound not recognized, or a simple word sequence wrongly recognized as a compound. There were 3 differences of this kind. Example: a compound *dobro delo* ‘a good deed’ was recognized as a sequence *dobro ADV delo N*, where a POS *ADV* is wrongly assigned (it should be *A*).
- **foreign words.** A foreign word assigned an incorrect POS. There were 3 of this kind. For instance, *of* in *Siti of Pariz* ‘City of Paris’ was assigned PONCT tag (for punctuation marks and special characters) instead X (unknown/foreign words).

6.2 Unknown Units

We have also explored results for unknown units in order to picture how the system is able to behave on unseen units. These results are displayed in Table 6. We have investigated three sets of unknown units: lexical units absent from the training corpus (UC), units absent from lexical resources (i.e. not in dictionary and not recognized by NE transducers) (UL), units absent from training corpus and lexical resources (U). Raws ALL (resp. MW) correspond to all lexical units (resp. the multiword lexical units). For each set, column *cov.* displays its coverage on the tested text; column F displays the lexical tagging F-score. The column "global F" indicates the overall F-scores on the TEST set.

One first striking observation is that the lexicon has a very high impact for the prediction of multiword lexical units: the recognition of multiword units absent from the lexicon is a disaster, reaching an accuracy lower than 10%. Furthermore, we can deduce from the results that the impact of the lexicon-based features for simple units is mitigated: the model tends to favor other features.

Table 6. Scores on unknown units on the TEST set with the joint strategy and the use of lexicon-based features.

	global	UC		UL		U	
	F	cov.	F	cov.	F	cov.	F
ALL	90.35	28.5	79.12	26.8	91.87	4.8	60.32
MW	63.72	90.6	61.39	37.4	7.45	37.1	6.49

7 Conclusions and Future Work

This paper describes three methods to integrate POS tagging, NE recognition and compound recognition into a lexical tagging system for Serbian: two pipeline strategies involving a POS tagger and a NE/compound recognizer; a joint strategy performing the three tasks at the same time. All strategies were based on CRF models trained from a new annotated corpus and existing lexical resources. The experimental results showed that the joint strategy appears to be the more robust to tag out-of-domain texts. The lexical resources showed to greatly improve the accuracy of the system, especially for multiword unit tagging. This paper opens new perspectives. In particular, a neural network could be experimented in order to get freed from feature-engineering.

Acknowledgments. This research was partly supported by the Serbian Ministry of Education and Science under grant #47003 and by the French National Research Agency (ANR) through the project PARSEME-FR (ANR-14-CERA-0001).

References

1. Agić, v., Ljubešić, N., Merkler, D.: Lemmatization and Morphosyntactic Tagging of Croatian and Serbian. In: Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing, pp. 48–57. Association for Computational Linguistics, Sofia, Bulgaria, August 2013
2. Blunsom, P., Baldwin, T.: Multilingual Deep Lexical Acquisition for HPSGs via Supertagging. In: Proceedings of EMNLP 2006, Sydney, pp. 164–171 (2006)

3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (Almost) from Scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
4. Constant, M., Sigogne, A.: MWU-aware part-of-speech tagging with a CRF model and lexical resources. In: Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World (MWE 2011), pp. 49–56 (2011)
5. Constant, M., Sigogne, A., Watrin, P.: Discriminative strategies to integrate multiword expression recognition and parsing. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), pp. 204–212 (2012)
6. Constant, M., Tellier, I.: Evaluating the impact of external lexical resources into a CRF-based multiword segmenter and part-of-speech tagger. In: Proceedings of LREC 2012, Istanbul, Turkey (2012)
7. Courtois, B., Silberztein, M.: Dictionnaires électroniques du français. Larousse, Paris (1990)
8. Denis, P., Sagot, B.: Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In: Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (PACLIC 2009), Hong Kong (2009)
9. Finkel, J.R., Manning, C.D.: Joint parsing and named entity recognition. In: Proceedings of the conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2009) (2009)
10. Green, S., de Marneffe, M.C., Bauer, J., Manning, C.D.: Multiword expression identification with tree substitution grammars: a parsing tour de force with French. In: Proceedings of the conference on Empirical Method for Natural Language Processing (EMNLP 2011) (2011)
11. Gross, M.: The use of finite automata in the lexical representation of natural language. In: Gross, M., Perrin, D. (eds.) LITP 1987. LNCS, vol. 377, pp. 34–50. Springer, Heidelberg (1989). https://doi.org/10.1007/3-540-51465-1_3
12. Krstev, C., Obradović, I., Utvić, M., Vitas, D.: A system for named entity recognition based on local grammars. J. Log. Comput. **24**(2), 473–489 (2014)
13. Krstev, C., Vitas, D.: Finate state transducers for recognition and generation of compound words. In: Erjavec, T., Žganec Gros, J. (eds.) Proceedings of the 5th Slovenian and 1st International Conference Language Technologies. pp. 192–197. Institut "Jožef Stefan" (2006)
14. Lafferty, J., McCallum, A., Pereira, F.: Conditional random Fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), pp. 282–289 (2001)
15. Legrand, J., Collobert, R.: Phrase representations for multiword expressions. In: Proceedings of the 12th Workshop on Multiword Expressions, pp. 67–71. Association for Computational Linguistics, Berlin, Germany, August 2016
16. Maurel, D., Friburger, N., Antoine, J.Y., Eshkol, I., Nouvel, D., et al.: Cascades de transducteurs autour de la reconnaissance des entités nommées. Traitement Automatique des Langues **52**(1), 69–96 (2011)
17. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003, pp. 188–191 (2003)

18. Nivre, J., Nilsson, J.: Multiword units in syntactic parsing. In: Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA) (2004)
19. Savary, A., Ramisch, C., Cordeiro, S., Sangati, F., Vincze, V., QasemiZadeh, B., Candito, M., Cap, F., Giouli, V., Stoyanova, I., Doucet, A.: The PARSEME shared task on automatic identification of verbal multiword expressions. In: Proceedings of EACL 2017 Workshop on MWEs, Valencia, pp. 31–47, April 2017
20. Sečujski, M., Delić, V.: A software tool for semi-automatic part-of-speech tagging and sentence accentuation in Serbian language. In: Proceedings of IS-LTC (2006)
21. Shigeto, Y., Azuma, A., Hisamoto, S., Kondo, S., Kouse, T., Sakaguchi, K., Yoshimoto, A., Yung, F., Matsumoto, Y.: Construction of English MWE dictionary and its application to POS tagging. In: Proceedings of the NAACL/HLT Workshop on MWEs, Atlanta, GA, pp. 139–144 (2013)
22. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Daelemans, W., Osborne, M. (eds.) Proceedings of CoNLL-2003, Edmonton, Canada, pp. 142–147 (2003)
23. Tufiş, D., Koeva, S., Erjavec, T., Gavrilidou, M., Krstev, C.: Building language resources and translation models for machine translation focused on south slavic and balkan languages. In: Proceedings of the 6th International Conference Formal Approaches to South Slavic and Balkan Languages (FASSBL 2008), Dubrovnik, Croatia, pp. 145–152, September 2008
24. Utvić, M.: Annotating the corpus of contemporary Serbian. INFOtheca **12**(2), 36a–47a (2011)
25. Vincze, V., Nagy, I., Berend, G.: Multiword expressions and named entities in the Wiki50 corpus. In: Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP 2011), pp. 289–295 (2011)
26. Vitas, D., Krstev, C.: Processing of Corpora of Serbian Using Electronic Dictionaries. Prace Filologiczne LXIII, 279–292 (2012)

Improving Chunker Performance Using a Web-Based Semi-automatic Training Data Analysis Tool

István Endrédy(✉)

Faculty of Information Technology and Bionics, MTA-PPKE Hungarian Language Technology Research Group, Pázmány Péter Catholic University,
50/a Práter Street, Budapest 1083, Hungary
endredy.istvan@itk.ppke.hu

Abstract. Fine tuning features for NP chunking is a difficult task. The effects of a modification are sometimes unpredictable. The tuning process with a (un)supervised learning algorithm does not produce necessarily better results. An online toolkit was developed for this scenario highlighting critical areas in training data, which may pose a challenge for the learning algorithm: irregular data, exceptions in trends, unusual property values. This overview of problematic data might inspire the linguist to enhance the data (for example by dividing a class into more detailed classes). The kit was tested on English and Hungarian corpora. Results show that the preparation of datasets for NP chunking is accelerated effectively, which result in better F-scores. The toolkit runs on a simple browser and its usage poses no difficulties for non-technical users. The tool combines the abstraction ability of a linguist and the power of a statistical engine.

Keywords: NP chunking · Training data analysis · Feature tuning
Web based analysis tool · IOB labelling · WordNet

1 Introduction

The task of noun phrase (NP) extraction from a sentence is called NP chunking. This can be considered a sequential tagging process labeling each word by describing its role in the phrase. Various label sets are used in this task [16], out of which the IOB2 set contains only three items: beginning of a NP ("B"), inside ("I") or outside the NP ("O"). One of these labels are assigned to each word in the text (Table 1). The label set can also be more detailed. For instance, an explicit label may sign the end of the chunk ("E"), or a chunk composed of one token (a single) may have its own label ("S" or "1").

The input data of chunking traditionally has the following format: each word is in a new line with a tab-separated feature list. Feature can be any property of the given word: POS category, length of the word, countable/uncountable,

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 290–303, 2018.
https://doi.org/10.1007/978-3-319-93782-3_21

Table 1. More IOB representations: a sentence with five different IOB label sets

word	IOB1	IOB2	IOE1	IOE2	OC
These	I	B	I	E	[]
include	O	O	O	O	O
,	O	O	O	O	O
among	O	O	O	O	O
other	I	B	I	I	[
parts	I	I	I	E	]
,	O	O	O	O	O
each	I	B	I	I	[
jetliner	I	I	E	E	]
’s	B	B	I	I	[
two	I	I	I	I	O
major	I	I	I	I	O
bulkheads	I	I	I	E	]
,	O	O	O	O	O

animate or not, abstract or not, etc. The linguist has to find adequate features, the use of which will result in better F-scores. In a typical scenario, when the selection or modification is done, the NP chunking process will be run and after a while its quality can be seen. If the improvement is only moderate or not satisfactory, the linguist can modify the parameters of the NP chunker and some input features, and has to execute the program again. This iteration, however, takes time due to the requirements of statistical learning methods.

For feature tuning the classical approach is the trial-and-error style: the linguist adds many features to the data (new columns to each word) and (s)he tries out them and their combinations. This means several train/test phases and more weeks iteration time. This paper is connected to the data improving, and it tries to help only to prepare the dataset before tagging.

The present article would like to help at that point where the linguist tunes the features. The toolkit described gives an overview about the structure of the training set, and can give a feedback about the modifications of features on the fly, without running real tests. Standard evaluations will be needed only at the end, since the toolkit is used only to prepare datasets and to speed up feature tuning.

Certainly, the method described (and its results) can be reached with command line tools as well. The added value is this user friendly online tool, which can be used for users without advanced computer skills as well.

2 Related Works

There are two basic types of improving chunker performance: software or data improvements. Several researches create, combine or improve a tagger engine, which performs better on the same data [6,10,14]. Other approach is to find, define or combine features and/or labels which produce better F-score for the given chunking task [12,13]. F-score is typically counted against a gold standard: correctly recognized chunks are true positive (tp), incorrect ones are false positive (fp), and missing ones are false negative (fn). Precision is counted by tp/(tp + fp), recall by tp/(tp + fn), and F-score is the harmonic mean of precision and recall.

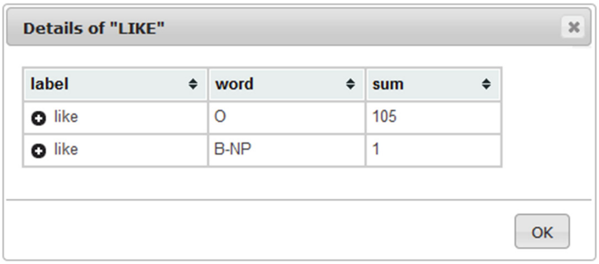
Specialization also works, for instance, when IOB labels are split into more detailed classes [9,12]. In that approach labels (above a threshold) are completed with POS category. For example, the following (word, POS, IOB) tuple *(You, PRP, B-NP)* get detailed IOB label: *(You, PRP, PRP-B-NP)*. Our solution is similar, however, we try to fine tune the POS info instead of the labels.

The state-of-the-art NP chunker solution for English is based on a voting system between more learned models with different IOB representations [12].

3 Use Case

The primary usage area of this tool are the use cases where the supervised or unsupervised learning algorithm faces a problem it is not able to solve in the required quality. If learning algorithm is able to learn and solve a problem on the given training/test data (choosing the best parameters for the highest F-score) the usage of this tool is superfluous. The tool highlights areas in training data, which may pose a challenge for the learning algorithm: irregular data, exceptions in trends, unusual values of properties. This overview of problematic data might inspire the linguist to enhance the training data (for example by dividing a class into more detailed classes or annotate a given property with a new class) or any further data fine tuning ideas.

A typical use case might be the recognition of a word behaving differently than its POS category, so it can be split into more classes by the linguist: *for, with, after, above, like, near* belongs to POS *IN* (in CoNLL2000 dataset [15], it uses Penn Treebank POS tags [7]), although they do not appear in an NP most of the time (as other *IN* words do).



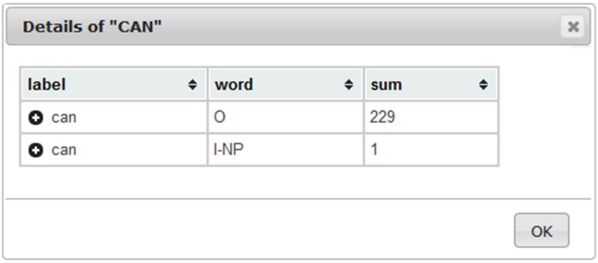
Details of "LIKE"

label	word	sum
⊕ like	O	105
⊕ like	B-NP	1

Fig. 1. If a word behaves differently than others in its POS (*like* has *IN* POS tag): it gets 105 times tag "O" (outside of NP) and once "B-NP" (begin of NP), it might require a new own POS (e.g. *IN* → *LIKE*)

Another example: words *every, what, who, whom* are always inside of an NP though other words in their POS are not. They might require a new category. This type of words can be found on "macros" tab of the web gui (Fig. 1).

Another use case might be an incorrect POS annotation in the dataset: the word *can* is always annotated as *MD* (modal verb), but once (in this training set) it was tagged as a part of an NP. This clearly shows that this word can be noun (as a tin) in this dataset as well, so it signs an annotation error. Similar examples for dataset error can be *saw, thought, won* (noun or verb), which might have been incorrectly annotated. The "categories" tab of the toolkit (Fig. 2) allows browsing and fixing of these symptoms.



Details of "CAN"

label	word	sum
can	O	229
can	I-NP	1

Fig. 2. Tool easily detects annotation errors in dataset (noun or verb): *can* has *MD* POS, but once it is annotated as "I-NP" (inside of NP), this suggests that case was noun in fact (not modal verb)

Recapping the advantages of dataset fine tuning: (1) the modified data is easier to learn, the tool only shows the irregular cases, and (2) the decision is made by a human, (3) the data becomes more readable, modifications are simple (data is not transformed into a machine friendly format as is the case with most learning algorithms).

There are without doubt situations which cannot be solved by a new class or class splitting. This tool provides a helpful overview, the problem resolution still remains the responsibility of the linguist. The tool does not aim to compete with learning algorithms, it provides help from a different angle: by giving an overview of the data it might give clues to the linguist about the next steps in feature tuning to create better data for the used learning algorithm. The main drive behind the development of this tool was exactly this feature and it proved useful for the author in many cases.

4 Problem of Feature Tuning

A key aspect of NP chunking is the set of features which are used in labelling algorithms of any type. At this abstraction level, the feature set is independent from the chunking method. Thus, the task of feature selection can be separated from the actual algorithm, and its direct effect on F-score can be monitored.

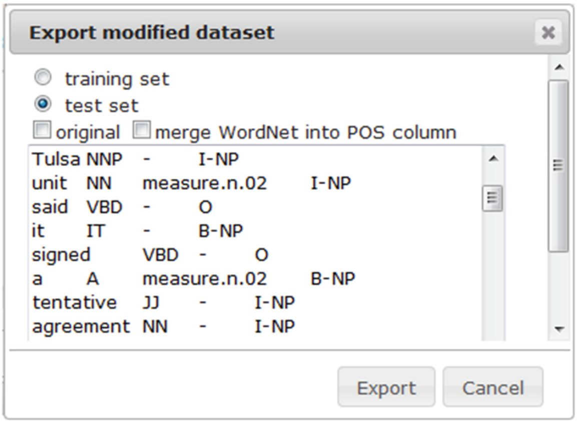
Thus questions like how the features can be selected or which ones help labelling come up. Since in solving such problems, a linguist is usually involved, we might rely on their good intuitions. However, their work is also to be supported.

The available statistical NP chunker methods have a common point from this aspect [6,10]. All of them are used as black boxes, no one can control their feature level processes. However, they can easily be applied to any language or data. On the contrary, rule based systems are fully controlled by linguists, but they are built manually in a slow process, and they are strongly connected to the given language and POS tagset [2,5].

5 Idea

The basic idea of our method is to combine the intuition and control of a linguist and the power of statistical engines. A set of texts contains useful information regarding the connection between features and IOB labels that can be extracted by statistical algorithms using these texts as training data. For instance, a human would hardly find a few features which behave differently with respect to IOB labelling. These features therefore need to be split into more types.

Our aim is to detect and show the best features (from the aspect of IOB labelling), and to find the ones which need manual fine tuning. At the end of the process, training and test sets can be exported (see Fig. 3) and used in any type of NP chunkers. The features tuning in this article focuses only on the POS information. However, the tool can work on other features as well, but POS feature tuning is able to demonstrate the tool and its usage.



Export modified dataset

○ training set
● test set
□ original □ merge WordNet into POS column

```
Tulsa NNP   -        I-NP
unit   NN   measure.n.02      I-NP
said   VBD  -        O
it     IT   -        B-NP
signed      VBD  -        O
a      A    measure.n.02      B-NP
tentative   JJ   -        I-NP
agreement   NN   -        I-NP
```

Fig. 3. Dataset preview at exporting (word, POS, wordnet synsets, iob label)

6 Toolkit

An online tool[1] was developed, which contains a training and test set for NP chunking. At the moment CoNLL2000 [15] dataset is imported. CoNLL2000 contains the following tuples: word, POS, IOB label. Although, IOB labels can also be tuned successfully (as mentioned in Sect. 2), in this article feature tuning is limited to POS fine-tuning only. This strong limitation is only for simplifying and demonstrating the impact of the toolkit.

Moreover, there are plenty of methods for feature selection. These methods can handle many types of feature, and they reduce effectively their number. But automatic feature selection is able to distinguish between useful and unuseful features, and this process is a black box for a human. (In practice: it might show which feature columns can be skipped.) On the one hand, in our case we investigate only one feature (POS). On the other hand, this toolkit would like to keep the feature tuning supervised and controlled by a human. Therefore automatic feature selection was not applied.

The tuples of the dataset are simple, and the tool can demonstrate the basic idea on it: statistical suggestions help the linguist to make useful modification on the dataset. In practice, the tool is able to tune other features as well.

To sum it up, this tool can optimize one feature at a time, in our case POS is tuned. But it can be applied to any features (not only for POS), even to output of another feature selection methods. The tool was tested with English and Hungarian datasets.

Statistical or rule based chunkers perform better with features which occur always with the same IOB label. On this assumption, we would like to have such features. It can be achieved when features have low cardinality, and if more specific sub-features would cause higher kurtosis in the distribution of features and their IOB labels. In other words, if the number of IOB labels assigned to a given feature would decrease with a more specific feature, it should be used. For instance, supposing $NOUN$ have 3 IOB labels (B, I, E) with the same probability, but a special subset $MISTER$ has most of the time only label B, then usage of this sub-feature would help the system performance. This conditions are fulfilled at the case of POS not only in English but in Hungarian, too. In this article English POS is tuned, but similar features exist in agglutinative languages as well. For clarity, these new sub-features (subPOS) are used as new decision classes instead of the original features.

The toolkit investigates the training set, and makes suggestions for features classified to more than one different IOB labels. Of course, this can be normal (a noun can stand in different positions of a NP), but it can also indicate that certain POS categories should be split into more detailed POS tags for better IOB labelling. For instance, a word might behave differently in a given POS category than the others (with respect to labelling), therefore this word should get a new category. These suggestions are important, and the main function of the toolkit is to hunt for them.

[1] https://github.com/endredy/onlineChunkerToolkit.

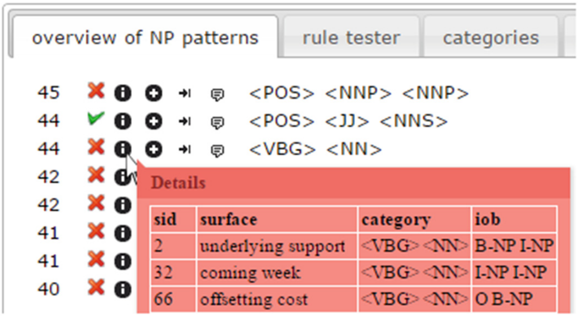
The features of the toolkit:

- the best feature patterns are detected,
- ambiguous patterns are shown,
- features can be browsed with respect to IOB labelling,
- new features can be defined, which are applied to the training/test sets,
- it may speed up feature tuning
- a regular-expression-based grammar is built automatically to verify quality,
- grammar rules can be added manually as well,
- it estimates an approximately F-score of the training set on the fly
- each suggestion comes from the training set only, the test set is kept separately. The test set is used only when final modifications are exported.
- modified training/test datasets can be exported
- the tool is open source

7 How it Works

Firstly, toolkit investigates the training set, secondly, it offers feature suggestions in more ways: browsing POS tags (ordered by frequency/assigned IOB label number/usefulness respect to IOB labelling), listing all valid NP sequences with POS, and listing POS suggestions which might correlate better with IOB labels (generated from WordNet, details in Sect. 9). If user accepts some suggestions (or create a new one), the dataset is automatically investigated again with the modified POS, and it starts over again.

If a feature sequence (in our case POS sequence) always gets the same IOB label in the training set, then this pattern will be signed with a green check icon, and it will be put into a grammar set. Other patterns will be signed with a red x, more than one IOB labels are assigned to them. (Figure 4)



45 ✗ <POS> <NNP> <NNP>
44 ✓ <POS> <JJ> <NNS>
44 ✗ <VBG> <NN>
42 ✗
42 ✗
41 ✗
41 ✗
40 ✗

Details

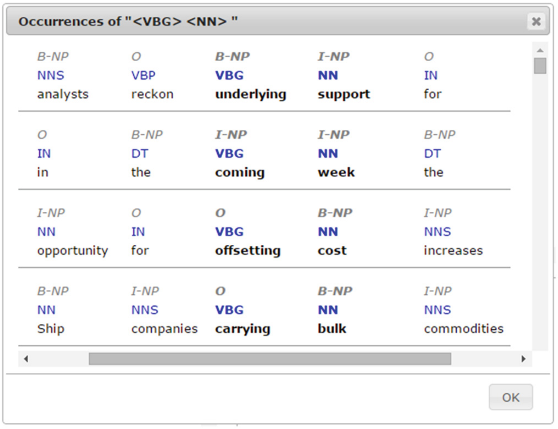
sid	surface	category	iob
2	underlying support	<VBG> <NN>	B-NP I-NP
32	coming week	<VBG> <NN>	I-NP I-NP
66	offsetting cost	<VBG> <NN>	O B-NP

Fig. 4. Red and green NP patterns with respect to IOB labels

However, accepting every green case would result in overfitting and in low recall. It is therefore important that all decisions are done by the linguist, the tool only prepares and suggests. If (s)he accepts general cases (not data specific ones), overfitting can be avoided.

Red patterns can be overviewed, and if a pattern seems to be acceptable, it can be put into the grammar with one click.

In addition, features can be overviewed with respect to IOB labelling: each feature shows its assigned IOB labels (from the training set). If a feature has only one label, it is the best case. (Statistical NP chunkers will learn it easily.) If not, then one can browse all its occurrences in the training set (Fig. 5).



Occurrences of "<VBG> <NN> "

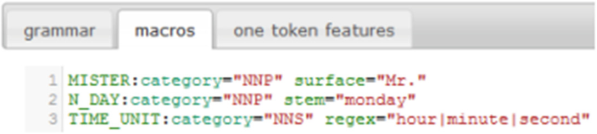
B-NP NNS analysts	O VBP reckon	**B-NP** **VBG** **underlying**	**I-NP** **NN** **support**	O IN for
O IN in	B-NP DT the	**I-NP** **VBG** **coming**	**I-NP** **NN** **week**	B-NP DT the
I-NP NN opportunity	O IN for	**O** **VBG** **offsetting**	**B-NP** **NN** **cost**	I-NP NNS increases
B-NP NN Ship	I-NP NNS companies	**O** **VBG** **carrying**	**B-NP** **NN** **bulk**	I-NP NNS commodities

Fig. 5. Browsing all occurrences of a POS pattern and its context with respect to their IOB labels

At this point, there are usually some words which behave differently than most words in the same category. For instance, in the CoNLL2000 dataset the word "Mr." has the POS tag NPP, but it behaves differently from the other proper names: "Mr." likes to stand at the beginning of a NP. (In other words it gets most of the time labelled as "B".) The toolkit supports to find easily that type of features, which get mostly one label (80%), and presents them to the user who can split the POS category by one click. Our toolkit gives the opportunity to create a new so called macro, and the given word will have the new feature instead of the original one. In this case "Mr." will get the tag MISTER, therefore it can be handled separately. This way, the feature tag of the odd word is replaced with a new one. The macro can be defined based on features, the surface form, the stem of the word or a regular expression pattern as well (latter two options are for future usage).

When features are browsed, one can define a new macro with one click. The program automatically writes a macro and gives an opportunity to modify it.

Macro definition supports the creation of a more powerful grammar: not only features (for example POS tags) but the surface form of words can also be added to the rules by their macro name. (Figure 6)



```
MISTER:category="NNP" surface="Mr."
N_DAY:category="NNP" stem="monday"
TIME_UNIT:category="NNS" regex="hour|minute|second"
```

Fig. 6. Examples for defining new categories (macros): based on surface form of word, stem or regex

This is made automatically by the toolkit: if new macros are applied ("apply" button is pushed), then the training set is converted with macros, and the extraction of the best feature sequences is done with these new features. If the F-score is not increased, then the added macros are useless. They should be dropped and other ones can be added. The evaluation takes only a few seconds, much faster than, for instance, tests in real statistical NP chunkers.

8 F-score Computing in the Toolkit

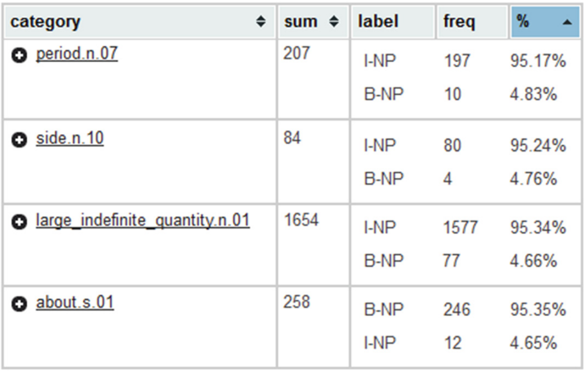
A regular expression rule based NP grammar engine was also developed which gets its rules automatically from the training set. (The author believed in pure rule based NP grammar for Hungarian, but this project had no success.) Every unique NP pattern in the training set becomes a rule in the grammar, and the training set itself is evaluated by these rules (no test set F-score hiking). This can show the quality and conformity of the dataset after the last modification of the linguist. Just to speed up development process: you don't have to export data, evaluate it in real environment (learning algorithm train and test phases) and then go back to feature tuning with the feedback.

At this moment, rules are used only for verifying F-score, they do not play role in the final exported dataset. The grammar contains regular expression rules, just like the grammar found in the Natural Language Toolkit (NLTK) [1]. The difference is that our rules are executed from javascript in a browser, not in a standalone Python program, and the results can be seen on the fly in a window ("grammar" tab on the gui). Their syntax is similar, but the grammar engine is written from scratch in our case, serving two aims.

To sum up the aims of the F-score computing in the toolkit: first, it can give a fast feedback about the last modifications: whether they moved the dataset into a better state or not. Second, grammar rules can be tuned. The F-score is an approximately value based on a simple NP grammar, which is built automatically from features and IOB labels. It is not a real F-score, just a metric how features correlate to IOB labels. It is counted on training set, and it might show how "IOB friendly" the dataset is. Test set is separated, it is used only at exporting modified dataset. The F-score computed by the toolkit can be compared only to the earlier F-scores: it rates the last modifications. (The value is independent from the F-score of real learning train/test results, like results in Sect. 10.)

9 WordNet Helps Discovering New Features

Another source of possible feature suggestions is WordNet synsets [8]. First of all, synsets and hyponyms of each word were generated into a new column of the dataset (separated by slash). Second, these synsets were split by slash, and every single synset was investigated with respect to IOB labelling.



category	sum	label	freq	%
period.n.07	207	I-NP	197	95.17%
		B-NP	10	4.83%
side.n.10	84	I-NP	80	95.24%
		B-NP	4	4.76%
large_indefinite_quantity.n.01	1654	I-NP	1577	95.34%
		B-NP	77	4.66%
about.s.01	258	B-NP	246	95.35%
		I-NP	12	4.65%

Fig. 7. Examples for feature suggestions generated from WordNet: synsets correlations of IOB labelling

Finally, synsets were sorted by the number of IOB labels and frequencies: best synsets have one or rarely two labels. If the user likes any of the suggestions, it can be added by a single click. WordNet synset suggestions will be used instead of the originally POS in the exported dataset, and every related word will have it. For instance, POS of the word *period* could be *period.n.07* instead of *NN*, because the former correlates better to its IOB labels than the latter one. Some suggestions are shown in Fig. 7.

10 Experimental Results

The CoNLL2000 dataset was imported into the toolkit. As we focused on NP chunking, other labels (VP, PP, etc.) were eliminated from the dataset and they were changed to the label “O”.

During the evaluation, first, POS features were evaluated by their co-occurrences with IOB labels. Second, some macros were added to the dataset, concentrating on words having the same POS category, but different behaviour. The best WordNet suggestions were also added, and finally, training and test data were exported and used in several NP chunkers.

All these steps were made with the help of the toolkit, and only POS were fine-tuned. Results were measured on this modified CoNLL2000 dataset.

The baseline test was made with the unigram and bigram NP chunkers of the NLTK toolkit. In addition, a new statistical tagger was also used in the evaluation: HunTag3 [3], which is a general-purpose sequential tagger with linear SVM classifier (Maximum Entropy) of Liblinear [4] and Maximum Entropy Markov Models [11]. This tool was developed in an ongoing parallel project.

Table 2. CoNLL2000 test runs with and without the toolkit, only POS data were modified

method	F-score
NLTK - unigram chunker	
with original tags	83.2%
with modified tags by toolkit	**83.8%**
NLTK - bigram chunker	
with original tags	84.5%
with modified tags by toolkit	**86.1%**
HunTag3	
with original tags	92.68%
with modified tags by toolkit	**92.74%**
voting system between more chunkers[12]	
with original tags	92.74%
with modified tags by toolkit	**94.12%**
voting system between more chunkers + HunTag3	
with original tags	93.13%
with modified tags by toolkit	**94.59%**

Results show the usefulness of the toolkit: it could help every NP chunker to reach higher F-scores (Table 2).

The current state-of-the-art NP tagger is 'SS05' [12], which achieves 95.23% on the CoNLL2000 dataset. Its method is based on voting between more data representations, which means different IOB labelsets (IOB1, IOB2, IOE1, IOE2, O+C) and each IOB label is completed with POS. This solution modifies the IOB labels, specialized them with POS. This idea is similar to our approach, however, we try to fine tune the POS info instead of the labels. SS05 voting system was reimplemented in python: conversions between different IOB labelsets, adding POS to labels, training each representation with TnT tagger of NLTK [1], converting results to a common labelset, and voting between the results. HunTag3 was also added as a 6th system (see Table 3), and it could improve the final F-score of the voting (+1.4%).

Even though our results are lower than SS05, but our aim was to demonstrate the power of the toolkit, when using to boost the results of existing chunkers.

Table 3. Detailed results of the voting system between different data representations: not only IOB labels (based on SS05) but POS were also modified by the toolkit. It could improve each voting format.

voting format	with original POS	modified POS by the toolkit
IOB1	92.01%	93.57%
IOB2	90.71%	92.04%
IOE1	90.64%	92.18%
IOE2	88.67%	89.96%
O+C	90.52%	91.71%
after voting	92.74%	**94.12%**
after voting, HunTag3 added	93.13%	**94.59%**

11 Conclusion

Hundreds of features increase not even the training space and time, but contradictory features may prevent to reach higher F-score in the task of NP chunking. Manual detection of missing or inaccurate features is more than problematic. However, the tool presented in this paper could easily improve the quality of the features quickly (+2% F-score improvement).

No doubt, the results of this toolkit could be reached with command line tools as well. However, the steps of the data tuning were made in a user friendly web application, with mouse clicks in a short time, without any advanced IT user skills. This toolkit is designed for linguists who are not developers but have good intuitions and a web browser.

This toolkit provides an automated way of training set analysis. The user is guided through the problematic cases by the program. A person cannot overview all the specific details of a corpus, while a machine can not make abstraction. This toolkit connects the two approaches: details are shown to the user who can decide on the use of the features.

Any learning algorithm can work better with a more consistent dataset and with more IOB-friendly features.

12 Future Plans

The toolkit can be used to build a regular expression based NP grammar. At this moment it is used only for the verification of feature tuning. It would be interesting to build a simple rule-based NP chunker with this grammar. Rules may contain POS categories, macros, surface forms of words with the power of regular expressions. An algorithm is needed which merges rules in the simplest form and patterns should be imported from a bigger corpus (e.g. web). Then, it may result in a new NP chunker. Of course a language can not be described with finite patterns, but if the corpora is huge enough, this idea may work.

Acknowledgments. I would like to show my gratitude to Dr Gábor Prószéky for motivation, and I thank Dr Nóra Wenszky, Zsuzsanna Balogh, Borbála Siklósi and the anonym reviewers for their comments.

References

1. Bird, S., Klein, E., Loper, E.: Natural language processing with Python. O'Reilly Media Inc., Sebastopol (2009)
2. Déjean, H.: Learning syntactic structures with xml. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th CoNLL, ConLL 2000, vol. 7, pp. 133–135. ACL, Stroudsburg, PA, USA (2000). https://doi.org/10.3115/1117601.1117632
3. Endrédy, I., Indig, B.: HunTag3: a general-purpose, modular sequential tagger - chunking phrases in English and maximal NPs and NER for Hungarian. In: 7th Language & Technology Conference, Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC 2015), pp. 213–218. Poznań: Uniwersytet im. Adama Mickiewicza w Poznaniu, Poznań, Poland, November 2015
4. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
5. Johansson, C.: A context sensitive maximum likelihood approach to chunking. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th CoNLL, CoNLL 2000, vol. 7, pp. 136–138, ACL, Stroudsburg, PA, USA (2000). https://doi.org/10.3115/1117601.1117633
6. Koeling, R.: Chunking with maximum entropy models. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th CoNLL, CoNLL 2000, vol. 7, pp. 139–141. ACL, Stroudsburg, PA, USA (2000). https://doi.org/10.3115/1117601.1117634
7. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: the penn treebank. Comput. Linguist. **19**(2), 313–330 (1993)
8. Miller, G.A.: WordNet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
9. Molina, A., Pla, F.: Shallow parsing using specialized hmms. J. Mach. Learn. Res. **2**, 595–613 (2002)
10. Osborne, M.: Shallow parsing as part-of-speech tagging. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th CoNLL, ConLL 2000, vol. 7, pp. 145–147, ACL, Stroudsburg, PA, USA (2000). https://doi.org/10.3115/1117601.1117636
11. Ratnaparkhi, A., et al.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 133–142. Philadelphia, PA (1996)
12. Shen, H., Sarkar, A.: Voting between multiple data representations for text chunking. In: Kégl, B., Lapalme, G. (eds.) AI 2005. LNCS (LNAI), vol. 3501, pp. 389–400. Springer, Heidelberg (2005). https://doi.org/10.1007/11424918_40
13. Simon, E.: Approaches to Hungarian Named Entity Recognition. Ph.D. thesis, Budapest University of Technology and Economics Budapest (2013)
14. Sun, X., Morency, L.P., Okanohara, D., Tsujii, J.: Modeling latent-dynamic in shallow parsing: a latent conditional model with improved inference. In: Proceedings of the 22nd COLING, vol. 1, pp. 841–848. ACL (2008)

15. Tjong Kim Sang, E.F., Buchholz, S.: Introduction to the CoNLL-2000 shared task: Chunking. In: Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th CoNLL, ConLL 2000, vol. 7, pp. 127–132. ACL, Stroudsburg, PA, USA (2000). https://doi.org/10.3115/1117601.1117631
16. Tjong Kim Sang, E.F., Veenstra, J.: Representing text chunks. In: Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics, pp. 173–179. Association for Computational Linguistics (1999)

A Connectionist Model of Reading with Error Correction Properties

Max Raphael Sobroza Marques[1](✉), Xiaoran Jiang[2], Olivier Dufor[1],
Claude Berrou[1], and Deok-Hee Kim-Dufor[1]

[1] Institut Mines Télécom Atlantique, Département d'Electronique,
Technopôle Brest-Iroise, 29238 Brest, France
max.sobrozamarques@imt-atlantique.fr

[2] Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes, France

Abstract. Recent models of associative long term memory (LTM) have emerged in the field of neuro-inspired computing. These models have interesting properties of error correction, robustness, storage capacity and retrieval performance. In this context, we propose a connectionist model of written word recognition with correction properties, using associative memories based on neural cliques. Similarly to what occurs in human language, the model takes advantage of the combination of phonological and orthographic information to increase the retrieval performance in error cases. Therefore, the proposed architecture and principles of this work could be applied to other neuro-inspired problems that involve multimodal processing, in particular for language applications.

Keywords: Connectionist model · Reading model
Word error correction · Associative memory
Multimodal neural network

1 Typoglycemia: The Error Correction Abilities of the Brain When Reading

Typoglycemia refers to the ability to read and understand words wherein letters are transposed.

> Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are.[1]

The message says that according to a supposed study from Cambridge University, we can easily read sentences containing words with transposed letters. As stated by (Grainger and Whitney 2004), this ability to correct words with scrambled letters indicates the existence of a special way of decoding input information that allows us to access the correct meaning of the word.

[1] http://www.mrc-cbu.cam.ac.uk/people/matt.davis/Cmabrigde/.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 304–317, 2018.
https://doi.org/10.1007/978-3-319-93782-3_22

Based on the literature in psycholinguistics on priming effects in reading, it has been shown that this ability involves among others the open bigram theory. The models such as the SERIOL model (Whitney 2001) manipulating open bigrams provide a good alternative to solve this issue while only considering the orthographic information. However, many controversies about both theories and models subsist and experimental counterexamples have been plentiful.

The aim of this work[2] is to implement a multimodal approach of a connectionist model to correct errors in reading process. For this purpose, we will use techniques based on neural cliques (See Sect. 3).

The dual-route cascaded (DRC) model (Coltheart et al. 2001) is a framework that includes the notion of multimodality in word recognition. Several studies found evidence to support this framework especially from dyslexia impairments. The proposed architecture is based on DRC framework. It is also composed of two routes: a lexical route and a sublexical route. The sublexical route is usually used for the recognition of non-words and it contains a grapheme-phoneme rule system that converts letters in phonemes representations. On the other hand, the lexical route includes orthographic and phonological representations of words.

The input errors are classified into five different categories, illustrated in the examples below (Table 1).

Table 1. Categories of typographical errors in words

Class	Target word: *AIRPLANE*
Transposition	AIR**LP**ANE
Erasure	AIR_LANE
Substitution	AIR**K**LANE
Deletion	AIRPLNE
Insertion	AIR**F**PLANE

Recent research links the behavior of correction reading with priming effect properties. Among them, we lay emphasis on *relative-position priming*, *transposition priming*, *superset priming* and *phonological priming*.

When a stimulus shares sub-sequences of letters with the target word, the reading process is facilitated. For example *JUSTICE* and *JUASTICE*. This effect is called relative-position priming (Grainger et al. 2006). The proportion of shared sub-sequences and the word length count to the presence of priming effect.

Transposition priming is an effect that facilitates the reading process when the stimulus has the same letters as the target word and when there are small variations in the order of the letters (Schoonbaert and Grainger 2004). Moreover, (Perea et al. 2003) observed a stronger priming effect when transpositions are in

[2] This work was supported by the European Research Council under ERC Grant agreement number 290901 NEUCOD.

adjacent positions. (Rayner et al. 2006) reported that sentences with transposed letters decrease the reading rate. However, these sentences are much easier to read than sentences with substitutions. They found evidence that when transpositions concern the ending or beginning of words, the sentences are more difficult to read. Furthermore, (Christianson et al. 2005) showed that the morpheme boundaries play an important role in visual recognition. For example, when the target word is *SUNSHINE*, the stimulus *SUNHSINE* is read more easily than *SUSNHINE*.

Superset priming is a phenomenon observed when unrelated letters are inserted in the stimulus and when all letters of the target word are preserved. It is demonstrated that each inserted letter linearly increases the processing cost of word recognition (Welvaert et al. 2008). Nevertheless, this gain (average of 11 ms per letter insertion) remains small compared to global processing of visual word recognition.

Finally, the experiments in (Van Orden 1987) show that phonological sources of activation are used in word recognition. So, if primes have a phonological that overlap better those of target words, they contribute better to the recognition process. (Example: *TOATL* vs *TTAOL* for *TOTAL*).

All the mentioned effects above are related with word error correction capability in transposition, erasure, deletion and insertion cases. Priming effect is considered both an ascendent (bottom-up) and descendant (top-down) mechanism independently of the level of pre-processing. These two phenomena are implicit and undissociable (Squire 2004; Tulving et al. 1990; Schacter et al. 1998). In the present article, only priming effects as a consequence of learning orthographic and phonological sequences is considered in order to building of binary connections in the network. As such, this learning procedure acts as a form of preconditioning phenomenon for the testing step. However, both the binary nature of the network and the absence of favoritism when one or more neurons are present within the several answers the network returns considerably reduces the effect of top-down priming in our design.

2 The Connectionist Nature of Cortical Operation

The human neocortex is a complex circuit composed of tens of billions of neurons with a surface of 2600 cm^2. They are interconnected by a vast number of synapses (order of 10^{14}) (Azevedo et al. 2009). The activation of one or more synapses can fire other neurons. In other words, a neuron is a processing unit which aggregates one or more inputs and combines them to produce an output signal. The same idea is present in McCulloch-Pitts neuron model (McCulloch and Pitts 1943).

Connectionist or neural network modeling is a specific computational method that simulates the behavior of interactions between neurons. These models have some advantages over other methods (Plaut 2005). First, connectionist models are explicit about mechanisms and constraints in the brain. Second, they are a good tool to validate some hypotheses related to the representation of a cognitive or learning process. Third, neural networks have the ability to generalize input patterns. In our case, we will use this property to correct errors and provide invariability in the decoding process. For example, when reading the pattern *AIRLPANE*, our cortex provides invariability to read this word with the meaning *AIRPLANE*. Finally, some of these networks offer mechanisms to avoid loss of knowledge even if there are damaged connections or neurons. We call this property resilience.

3 Assembly Coding and Neural Cliques, Coding and Decoding Principles

3.1 Clique-Based Neural Networks (CBNN)

(Gripon and Berrou 2011) proposed a new model of a neuro-inspired associative memory. This model combines the Willshaw-type model (Willshaw et al. 1969), a clustered structure and distributed codes to encode and decode mental information. The advantages to use this type of network are the ability to store patterns with a good performance when retrieving partially damaged messages, robustness and biological plausibility as explained in (Berrou et al. 2014).

Binary tournament-based neural networks, as an extension to deal with sequences efficiently, were introduced in (Jiang et al. 2015a). To do so, non-oriented connections of a clique-based model are replaced with oriented connections (chain of tournaments). Therefore, when a clique is activated, sequences related to it may also be triggered.

As in any associative memory, there are two different procedures. The first one is storing. In this procedure, an input pattern is given to the network and then connections are drawn. The second one is message retrieval in which for a given input the network will activate the pattern with maximum correspondence.

Model Representation and Storing Procedure. In CBNN, a message is represented by a fully interconnected sub-graph (namely clique) with binary connections. There are five important concepts to describe a clique-based neural network:

Fanal: A node in the network.

Message: A vector of c fanals.

Connection: A non-oriented edge in a sub-graph.

Cluster: A group of l fanals. In a clique, there is at most one fanal per cluster.

Clique: A group of c fanals that encodes a message in the network. This is a fully interconnected sub-graph, as depicted in Fig. 1 and 3b).

3.2 Coding Neural Cliques with Twin Neurons

In clique-based associative memories, if the stored messages materialize correlated data, the retrieval quality decreases. (Boguslawski et al. 2014) proposed a method to alleviate this problem, the principle of *twin neurons*. After each

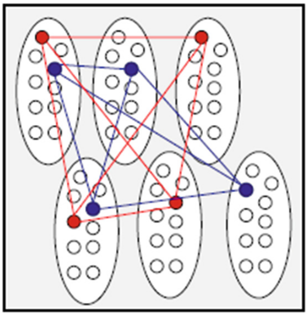


Fig. 1. An example of a clique-based network. Two cliques with order $c = 4$ are stored: in red and in blue. In this network there are $\chi = 6$ clusters and each one contains $l = 10$ fanals.

step of message storing, if the number of outgoing connections of a fanal exceeds a given threshold value, then a new fanal (*twin neuron*) is created. From this moment, new connections will use this cloned fanal, thus limiting the number of connections per node. An example is depicted in Fig. 3a).

3.3 Message Retrieval

The decoding procedure or message retrieval consists in an iterative process with two steps: *message passing* and *selection of winners*. In the first step, input fanals are activated in the network using an aggregation rule. There are two main aggregation rules in CBNN: the Sum-of-Sum (SoS) and Sum-of-Max (SoM) (Aboudib et al. 2014). In the second step, a selection rule is applied. For example, one fanal with the highest activity is elected in each cluster, named Local Winner Take-All rule (LWTA), or fanals with less activity are eliminated using a threshold filter, named Losers Kicked-Out rule (LsKO) (Jiang et al. 2015b).

3.4 Decoding with Boosting

The performance, using a mentioned selection rule (LWTA or LsKO), drops dramatically in scenarios where input pattern distortions and/or severe interference (when cliques have overlapped fanals). Recently, (Jiang et al. 2015b) proposed an approach based on heuristic retrieval in order to increase the performance in these situations. It consists of several iterations composed of three steps. First, a fanal is chosen out of input graphs of activation. Second, this fanal receives a strong impulse activity. This activity will propagate towards its adjacent fanals. In the last step, a decoding rule is applied. The iterations stop after reaching a specific condition. This procedure is detailed in (Jiang et al. 2015b).

3.5 Decoding Blurred Messages

Blurred inputs are stimuli that contains inserted letters (Example: *AIR**R** PLANE*). (Gripon and Jiang 2013) proposed solutions in order to adapt the model of clique-based network to allow decoding these inputs in the case of stored uniform data. The algorithm uses the SoM rule and the Local Winner-Take-All (LWTA) propagation rule. Based on biological plausibility, the principle of divergence of neural connections (Thivierge and Marcus 2007) is strong evidence of the existence of blurred messages in the neocortex.

4 The Proposed Architecture

Reading is a complex cognitive process that involves interactions between several functional modules in the brain: visual, phonological, semantics and contextual modules.

As we mentioned in the introduction, we propose an architecture based on DRC model. For that reason, a Text-To-Phoneme converter software[3] for French language the Grapheme-Phoneme Rule System, depicted in Fig. 2.

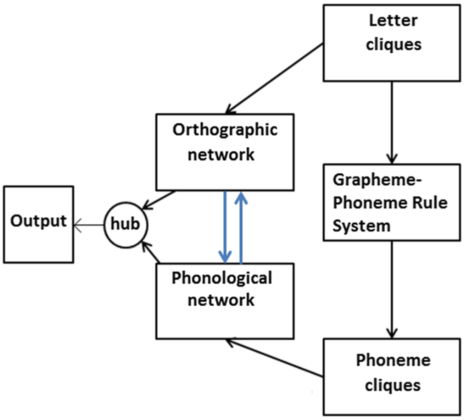


Fig. 2. Illustration of the network architecture based on the DRC framework.

(Leaver et al. 2009) found evidence of an anticipation effect of sound sequences in the brain. This effect is modeled by chain of tournaments in the phonological network.

As such, the proposed architecture is an interface network of phonological and orthographic information combining. The model ends up in a hub and other networks, which are not considered in this work, can be connected to the hub to create any kind of output system (speech or writing system). (Van den Heuvel and Sporns 2013) indicate the existence of nodes, named *hubs*, in the cortical network that have an important role in linking several modules.

[3] LIA_PHON v1.2, under GPL license, available in http://lia.univ-avignon.fr/chercheurs/bechet/download_fred.html.

4.1 Orthographic Network

In this network, letters of words are encoded in positional clusters. This model is an implementation of a clique-based network where fanals are letters, clusters are positions of letters and words are cliques. Nevertheless, according to (Boguslawski et al. 2014), clique-based neural networks are adapted only for data with uniform distribution. When the messages are correlated, the performance of this type of network decreases. In our application, the words are extremely correlated data. For this reason, we chose to use the technique of twin neurons. An example of the proposed orthographic network is illustrated in Fig. 3.

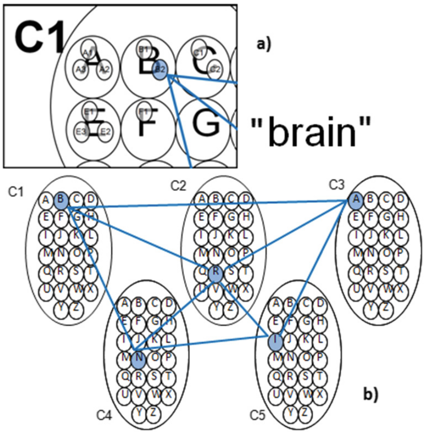


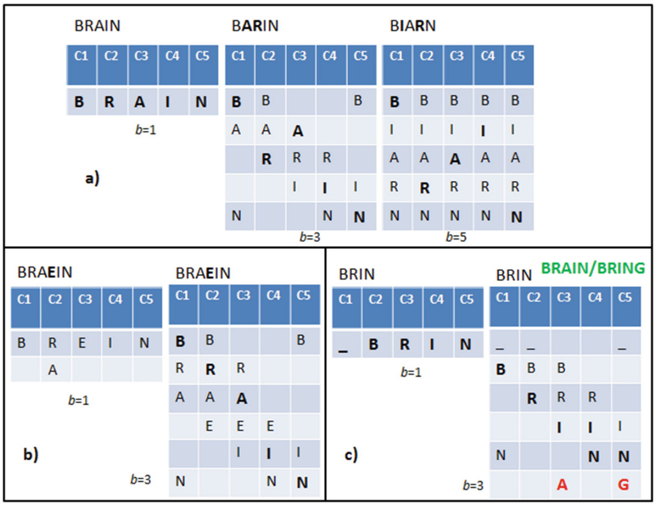
Fig. 3. Illustration of the storing procedure for the word BRAIN in the orthographic network. In b), the network has $\chi = c = 5$ clusters and each one contains $l = 26$ groups of fanals representing letters. The most frequent letters are represented in a) with more fanals according to the twin neurons principle.

We use a decoding strategy based on blurred messages and the boosting approach described in Sects. 3.5 and 3.4 respectively. The purpose of this strategy is to achieve error correction abilities in the orthographic network. So, we considered a system decoding capable of activating fanals of letters in different positions in the word. Then, this decoding system eliminates fanals that do not match those of the target word, thanks to the LWTA rule applied in the third step of the boosting decoding. The parameter b is the length of the activation window[4].

During the boosting decoding procedure, the parameter b is adjusted[5] to obtain an activated clique or several cliques. The number of clusters is adapted

[4] Example: if $b = 3$ the same letter pattern is activated in 3 adjacent clusters. Using the example of Fig. 3 activated fanals are {(B,C5);(B,C1);(B,C2);(R,C1);(R,C2);(R,C3);(A,C2);(A,C3);(A,C4);(I,C3);(I,C4);(I,C5);(N,C4);(N,C5);(N,C6)}.

[5] $b_0 = 1$ and then $b_{t+1} = 2 * b_t + 1$ (for $t = 0, 1, 2, 3, ...$) until the stopping condition is reached or $b > c$.



BRAIN

C1	C2	C3	C4	C5
B	**R**	**A**	**I**	**N**

b=1

BARIN

C1	C2	C3	C4	C5
B	B			B
A	A	**A**		
	R	R	R	
		I	**I**	I
N			N	**N**

b=3

BIARN

C1	C2	C3	C4	C5
B	B	B	B	B
I	I	I	**I**	I
A	A	**A**	A	A
R	**R**	R	R	R
N	N	N	N	**N**

b=5

BRAEIN

C1	C2	C3	C4	C5
B	R	E	I	N
	A			

b=1

BRAEIN

C1	C2	C3	C4	C5
B	B			B
R	**R**	R		
A	A	**A**		
	E	E	E	
		I	**I**	I
N			N	**N**

b=3

BRIN

C1	C2	C3	C4	C5
_	**B**	**R**	**I**	**N**

b=1

BRIN

C1	C2	C3	C4	C5
_	_			_
B	B	B		
	R	R	R	
		I	**I**	I
N			**N**	**N**
		A		**G**

b=3

Fig. 4. Illustration of activation of input fanals using a blurred messages strategy. In the table: (a) transposition case; (b) insertion case; (c) deletion case.

to the word length before the activation process. For the cases of insertion, the inserted letters are merged either in the previous or in the next cluster randomly. For the cases of deletion, we consider that the network does not know which cluster correspond to each letter. To model this phenomenon, empty clusters are randomly marked among all the available clusters.

Figure 4 shows examples of activation of input letters with different values of *b*. However, in certain cases, other words could also be activated, for example, in Fig. 4c) the target word *BRING* instead of *BRAIN* is also a possibility. Thus, we need to look into an additional mechanism to disambiguate these cases. Our strategy is to combine orthographic and phonological information to increase the degree of certainty in the correction mechanism of words. Indeed we concentrated our work on the convergence of two networks helping each other and acting according a bottom-up procedure.

4.2 Phonological Network

The results presented in (Perea and Carreiras 2006) show that the brain encodes the order of letters at the orthographic level rather than the phonological level and transposition priming effect is less present at phonological level. Therefore, we consider that the phonological network is less flexible in transposition error cases. In that way, the chain of tournaments has an important role in fixing the order of sub-sequences of phonemes of a word. (Example: sub-sequences of the word */breIn/* are */b/*, */r/*, */e/*, */I/*, */n/*, */br/*, */re/*, */eI/*, */In/*, */bre/*, */reI/*, */eIn/*, */breI/* and */reIn/*)

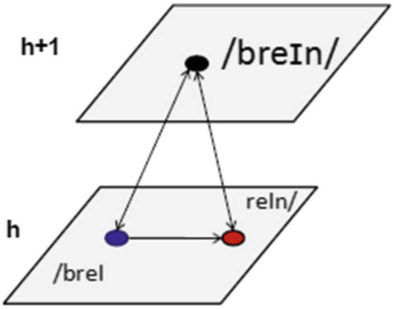


Fig. 5. An example of triangular pattern in the phonological network. The nodes are cliques. Two cliques at the bottom level */breI/* (in blue) and */reIn/* (in red) aggregate to form a clique of phonemes */breIn/* at the top level.

The architecture of the phonological network is a hierarchical multilayer network in which nodes represent cliques. The hierarchical structure is formed by triangular patterns connecting two consecutive layers h and $h + 1$ (See Figs. 5 and 6).

Bidirectional arrows represent connections between two levels in the network. One-way arrows represent a chain of tournaments (sequences). Each clique encodes a sub-sequence of phonemes of a word.

An example of the proposed phonological network is illustrated in Fig. 6. In the learning procedure, fanals are randomly chosen to compose cliques of single phonemes. Those cliques can share fanals together. At the next level ($h + 1$ level), the procedure is repeated to form cliques which represent the aggregated sequence of phonemes. In addition, all fanals of cliques of sub-sequences are fully connected with the main clique representing the word.

Sequences are used in order to anticipate the activation of cliques in the phonological network.

A bottom-up decoding approach is implemented for decoding. For each triangular pattern, as depicted in Fig. 6, the clique at the top layer is obtained using the propagation of *feedforward* activities of two cliques at the bottom level. Then, the activity is propagated to the sequence on the right. Therewith, a decoding procedure is applied with the SoM and WTA rules.

4.3 Combining Orthography and Phonology in a Hub Neural Network

The clique-based hub network is capable of integrating decisions from two different modules (phonological and orthographic). A bipartite clique-based network architecture is used in order to integrate decisions from the orthographic network to the activated cliques at the upper level of the phonological network. All fanals of the activated cliques in the phonological and orthographic networks are connected to a clique in the hub. The decoding procedure happens in

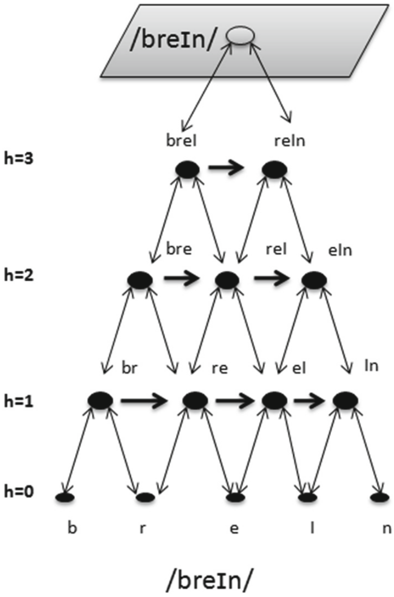


Fig. 6. Storing procedure of the word */breIn/* in the phonological network. h is the level number in the architecture. The nodes in each layer of the figure are cliques. In each triangular pattern, two cliques at the bottom level (h level) aggregate to form a new clique at the top level ($h+1$ level). All cliques of sub-sequences of phonemes (e.g. */br/* or */reI/*) are fully connected with the main clique of word */breIn/*. For the sake of clarity, these connections are not represented in the figure.

parallel within the two networks. The last step is to propagate the activity of these cliques towards the hub.

5 Results

For the tests, we used a lexical database[6] of French language to select the stimuli. All networks were created with fixed parameters. (For each level of the phonological network: $c = 8$, $\chi = 200$, $l = 200$ and for the orthographic network: $ConnectionLimitTwin = 25$, $\chi = c = 9$ and $numberOfFrenchLetters = 54$[7]. For the hub: $c = 8$, $\chi = 300$, $l = 300$).

In the first procedure, due to the random picking of fanals in the network, we built 20 different networks. Each network learns 6,163 French 7-letter lemmas (in the orthographic network) and its corresponding phonemes (in the phonological network). Then, 200 samples of each type of error were tested in these 20 networks. The results of these tests are given in Table 2.

[6] Lexique.org is a French lexical database of lexical information of 135,000 words and 55,000 lemmas.

[7] Accentuated and special characters are included.

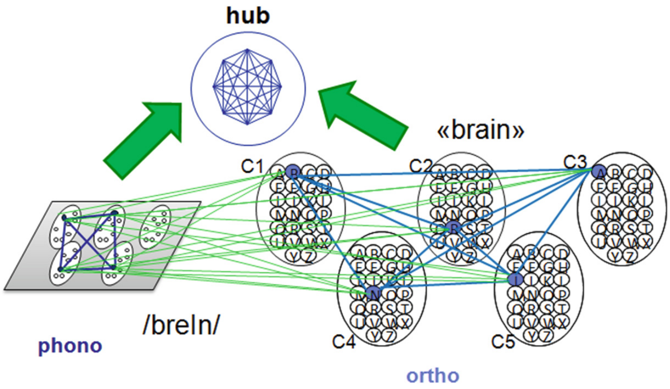


Fig. 7. Illustration of the decoding process in a bipartite clique-based network. The propagation is performed from the right to left side and from the phonological network. All fanals of both cliques are fully connected (*feedforward* connections) with a clique in the hub network.

Two criteria are used to evaluate the performance of word correction in each network: *matching*[8] rate and *error*[9] rate. The matching rate corresponds to the number of expected fanals (correct fanals) among the activated fanals in the last layer of the phonological network, the orthographic network or the hub. The error rate is calculated by computing the number of failures to find the exact clique within 4,000 tests in each situation. Even if the correct clique is found with additional fanals, the error rate equals one because it is a strict criterion. The final recognition rate is calculated based on the ratio of correctly recognized words (or phonemes) to the number of total words ($recognitionRate = 1 - errorRate$). The error rate is not calculated for the phonological network because several cliques are elected each time ($errorRate = 1$).

In the last procedure, the network learns 79 unique French words and phonemes of a translated typoglycemia text test bench proposed in (Starzyk et al. 2009). Then, the translated text was modified according to transposing rules used in the paper. As showed below:

> Je n'ariravis pas à criroe que je psuse effeiceetvmnt cmprnodere ce que j'éaits en trian de lrie: le puovior phnémnoéal de l'episrt haumin. Sleon une éqpuie de recehrche de l'Uvinertisé de Cmabrigde, ce n'est pas l'odrre des ltteers qui cmopte dnas un mot, la suele coshe ipmrotnate est que la pmeirére et la drenèire sioent à la bnnoe pclae. Le rsete puet êrte dnas un dsérorde ttoal et vuos puoevz tujoruos lrie snas polbrème. C'est prace que le creaveu hmauin ne lit pas chuaqe ltetre une par une, mias le mot cmome un tuot. Cttee cnotidion s'alepple la Typoglycémie. Inocrblyae, non ? Ouias, et vuos aevz tojruous psneé que l'oroathgrphe éatit impoartnte.

[8] $matching = numberCorrectFanals/c$.

[9] It provides zero if $numberActivatedFanals = c$ and $matching = 1$ else it provides one.

This network is then able to store words with variable length. For this purpose: $\chi = c = 16$ and each word contains $c - lengthWord$ padding characters at the end (Example: importante, importante######).

Table 2. Recognition rates (percentage) of the network.

Testing set	Rate	Network		
		Phono	Ortho	Hub
Transposition (adjacent)	Match	70.0	100.0	98.2
	Error	-	5.0	2.0
Transposition (1 between)	Match	57.0	99.0	90.7
	Error	-	26.6	9.8
Erasure (1 letter)	Match	100.0	100.0	94.5
	Error	-	11.5	5.5
Deletion (1 letter)	Match	85.0	99.7	88.2
	Error	-	43.2	11.62
Insertion (1 letter)	Match	99.0	100.0	99.0
	Error	-	5.8	1.1
Insertion (2 letters)	Match	93.0	99.8	98.0
	Error	-	12.2	2.2
Benchmark with 79 words	Match	76.9	100.0	100.0
	Error	-	0.0	0.0

Here are some examples of recognition ambiguities for the first procedure: (*COLONLE, colonel, colonne*); (*_EINDRE, ceindre, geindre, feindre, teindre, peindre*); (*NCRTASHER, cascher, crasher, castrer, cracher, catcher*).

The present model has a retrieval accuracy rate of 100% of 111 tested words in the proposed typoglycemia benchmark for French language. We can compare this result with the state-of-the-art for the English benchmark[10]. For instance, we have an accuracy of 94.67% for the hidden Markov models (HMM) and 84.36% for the Levenshtein distance methods. 100% accuracy was obtained for the LTM model based on the spatio-temporal memory proposed in (Starzyk et al. 2009) and the episodic neural memory model based on the fusion adaptive resonance theory proposed in (Wang et al. 2012). None of the two last models consider phonological information of words, the number of learned messages using the test bench is limited to 73 words and there is no insertion, deletion or erasure in the testing set.

The result shows that the error retrieval rate globally decreased in our hub network using a multimodal approach. Transpositions in adjacent letters are more easily recognized than non-adjacent cases. Finally, insertions of one or

[10] The English benchmark has 107 words, among which there are 73 unique words.

two letters have a strong superset priming effect (with 1.1% and 2.2% recognition error rates respectively). The existence of the connection in the phoneme sequence or the word composition is a form of minimalist top-down priming effect.

Those results indicate the model seems to mimic quite adequately some of the behavioral performances described in the literature of priming effects. The phonological information is not completely independent from orthographical information considering that an external system perform the conversion. Diversity of information could be more advantageous from a information theory perspective if the both modules were completely independent. All developed code in Java is available[11].

6 Future Work

Our future work will be to adapt the orthographic model to allow for words with variable length and consider the word frequencies by reinforcement[12] of connections. Then, we will study an extension of this model for sentences. To overcome this issue, we will consider the context and semantic information.

References

Aboudib, A., Gripon, V., Jiang, X.: A study of retrieval algorithms of sparse messages in networks of neural cliques. In: COGNITIVE 2014: The 6th International Conference on Advanced Cognitive Technologies and Applications (2014)

Azevedo, F.A.C., Carvalho, L.R.B., Grinberg, L.T., Farfel, J.M., Ferretti, R.E.L., Leite, R.E.P., Lent, R., Herculano-Houzel, S.: Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. J. Comp. Neurol. **5**, 532–541 (2009)

Berrou, C., Dufor, O., Gripon, V., Xiaoran, J.: Information, noise, coding, modulation: what about the brain? In: 8th International Symposium on Turbo Codes and Iterative Information Processing (ISTC) (2014)

Boguslawski, B., Gripon, V., Seguin, F., Heitzmann, F.: Huffman coding for storing non-uniformly distributed messages in networks of neural cliques. In: 28th Conference on Artificial Intelligence, vol. 1. AAAI (2014)

Christianson, K., Johnson, R., Rayner, K.: Letter transpositions within and across morphemes. J. Exp. Psychol. Learn. Mem. Cogn. **31**(6), 1327 (2005)

Coltheart, M., Rastle, K., Perry, C., Langdon, R., Ziegler, J.: DRC: a dual route cascaded model of visual word recognition and reading aloud. Psychol. Rev. **108**(1), 204 (2001)

Grainger, J., Granier, J., Farioli, F., Van Assche, E., Van Heuven, W.: Letter position information and printed word perception: the relative-position priming constraint. J. Exp. Psychol. Hum. Percep. Perform. **32**(4), 865 (2006)

[11] https://gitlab.com/msobroza/context-typo-network.git.

[12] The reinforcement of connections in a multilayer clique-based neural network is an unpublished problem.

Grainger, J., Whitney, C.: Does the huamn mnid raed wrods as a wlohe? Trends Cogn. Sci. **8**(2), 58–59 (2004)

Gripon, V., Berrou, C.: Sparse neural networks with large learning diversity. IEEE Trans. Neural Netw. **22**(7), 1087–1096 (2011)

Gripon, V., Jiang, X.: Mémoires associatives pour observations floues. In: Proceedings of XXIV-th Gretsi Seminar (2013)

Jiang, X., Gripon, V., Berrou, C., Rabbat, M.: Storing sequences in binary tournament-based neural networks. IEEE Trans. Neural Netw. Learn. Syst. **27**(5), 913–925 (2015a)

Jiang, X., Sobroza Marques, M.R., Kirsch, P.-J., Berrou, C.: Improved retrieval for challenging scenarios in clique-based neural networks. In: Rojas, I., Joya, G., Catala, A. (eds.) IWANN 2015. LNCS, vol. 9094, pp. 400–414. Springer, Cham (2015b). https://doi.org/10.1007/978-3-319-19258-1_34

Leaver, A., Van Lare, J., Zielinski, B., Halpern, A., Rauschecker, J.: Brain activation during anticipation of sound sequences. J. Neurosci. **29**(8), 2477–2485 (2009)

McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)

Mountcastle, V.: The columnar organization of the neocortex. Brain **120**(4), 701–722 (1997)

Perea, M., Carreiras, M.: Do transposed-letter similarity effects occur at a prelexical phonological level? Q. J. Exp. Psychol. **59**(9), 1600–1613 (2006)

Perea, M., Lupker, S., Kinoshita, S.: Transposed-letter confusability effects in masked form priming. Masked Priming: State of the Art, 97–120 (2003)

Plaut, D.: Connectionist approaches to reading. The Science of Reading: A Handbook, pp. 24–38 (2005)

Rayner, K., White, S., Johnson, R., Liversedge, S.: Raeding wrods with jubmled lettres there is a cost. Psychol. Sci. **17**(3), 192–193 (2006)

Schacter, D.L., Buckner, R.L.: Priming and the brain. J. Neuron **20**(2), 185–195 (1998)

Schoonbaert, S., Grainger, J.: Letter position coding in printed word perception: effects of repeated and transposed letters. Lang. Cogn. Process. **19**(3), 333–367 (2004)

Starzyk, J., He, H., et al.: Spatio-temporal memories for machine learning: a long-term memory organization. IEEE Trans. Neural Netw. **20**(5), 768–780 (2009)

Squire, L.R.: Memory systems of the brain: a brief history and current perspective. Neurobiol. Learn. Mem. **82**(3), 171–177 (2004)

Tulving, E., Schacter, D.L.: Priming and human memory systems. JSTOR (1990)

Thivierge, J., Marcus, G.: The topographic brain: from neural connectivity to cognition. Trends Neurosci. **30**(6), 251–259 (2007)

Van den Heuvel, M., Sporns, O.: Network hubs in the human brain. Trends Cogn. Sci. **17**(12), 683–696 (2013)

Van Orden, G.: A rows is a rose: spelling, sound, and reading. Mem. Cogn. **15**(3), 181–198 (1987)

Wang, W., Subagdja, B., Tan, A., Starzyk, J., et al.: Neural modeling of episodic memory: encoding, retrieval, and forgetting. IEEE Trans. Neural Netw. Learn. Syst. **23**(10), 1574–1586 (2012)

Welvaert, M., Farioli, F., Grainger, J.: Graded effects of number of inserted letters in superset priming. Exp. Psychol. **55**(1), 54–63 (2008)

Whitney, C.: How the brain encodes the order of letters in a printed word: the seriol model and selective literature review. Psychon. Bull. Rev. **8**(2), 221–243 (2001)

Willshaw, D., Buneman, O., Longuet-Higgins, H.: Non-holographic associative memory. Nature (1969)

Applications in Language Learning

The Automatic Generation of Nonwords for Lexical Recognition Tests

Osama Hamed(✉) and Torsten Zesch

Language Technology Lab,
Department of Computer Science and Applied Cognitive Science,
University of Duisburg-Essen, Forsthausweg 2, 47057 Duisburg, Germany
osama.hamed@uni-due.de
http://www.ltl.uni-due.de/

Abstract. Lexical recognition tests are frequently used to assess vocabulary knowledge. In such tests, learners need to differentiate between words and artificial nonwords that look much like real words. Our ultimate goal is to create high quality lexical recognition tests automatically which enables repetitive automated testing for different languages. This task involves both simple (words selection) and complex (nonwords generation) subtasks. Our main goal here is to automatically generate word-like nonwords. We compare different ranking strategy and find that our best strategy (a specialized higher-order character-based language model) creates word-like nonwords. We evaluate our nonwords in a user study and find that our automatically generated test yields scores that are highly correlated with a well-established lexical recognition test which was manually created.

Keywords: Lexical recognition tests · Nonwords generation
Words selection · Language models

1 Introduction

Lexical recognition tests [13] are frequently used for measuring language proficiency. In such a test, students are typically shown either real words (*denial*) or nonwords (*platery*), and need to decide if they are valid or not. The nonwords used in lexical recognition tests should look like words without being actually in the lexicon. Thus, *platery*, *interfate*, or *purrage* have been shown to work well while *abcde* or *autobahn* are less suitable. The main advantage of lexical recognition testing is its simplicity. It only takes five minutes, only "Yes/No" questions are asked as shown in Fig. 1, and scoring can easily be automated.

The task of having students recognize words for vocabulary proficiency testing goes back quite a long time, cf. [16]. The Eurocentres Vocabulary Size Test [13] is an early example of using nonwords for testing. They used 150 items – two thirds real words and one third nonwords. Lemhöfer and Broersma [12] create an adapted version called LexTALE that can be finished faster, as it only

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 321–331, 2018.
https://doi.org/10.1007/978-3-319-93782-3_23

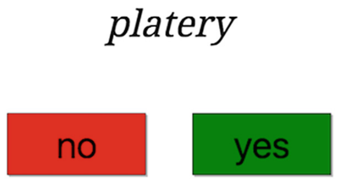
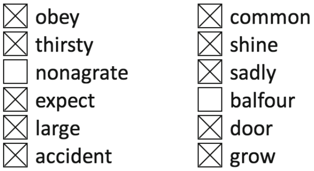


Fig. 1. Example of a lexical recognition test as Yes/No question.

uses 60 items. They validate the resulting scores by correlating them with other proficiency scores based on a word translation task and the commercial ‘Quick Placement Test’.

LexTALE has been adapted to other languages beyond English, e.g. Dutch and German [12], French [3], or Spanish [9]. Using nonwords constitutes an improvement over other forms of vocabulary proficiency testing, as it simplified the setup. For example, the Vocabulary Levels Test [14] is based on matching words with definitions, which is much harder to administer and automate.

Lexical recognition tests achieve a quite good approximation of a learner’s vocabulary with a relatively small number of test items [8]. Thus, lexical recognition tests can be quickly finished and usually fit on a single sheet of paper. This is the so called *checklist* format as shown in Fig. 2. When used in a computerized form, individual items are usually presented in isolation (e.g. LexTALE like tests) in order to minimize context effects.



☒ obey
☒ thirsty
☐ nonagrate
☒ expect
☒ large
☒ accident

☒ common
☒ shine
☒ sadly
☐ balfour
☒ door
☒ grow

Fig. 2. Example of a lexical recognition test in checklist format.

In the past, nonwords have been manually created, but for repeated testing as used in formative assessment [18] we need to be able to generate them automatically. Thus, in this paper we explore methods for automatically generating good nonwords.

2 Generating Nonwords

We model the selection of word-like nonwords as a two-step process where we first generate candidate strings and then rank them according to their ‘wordness’.

2.1 Candidate Selection

We generate random strings of different length and check against a list of known English words in order to ensure that we only have nonword candidates. This strategy will obviously create a lot of bad nonwords, which have little resemblance with known words. However, more informed strategies might already use the same information as will be later used for ranking and thus bias the results.

2.2 Candidate Ranking

In this section, we describe the different ranking strategies used to find good (i.e. word-like) nonwords.

Random Baseline. This is a simple baseline that randomly orders the nonwords. It is mainly used to set the other results into perspective.

Neighbourhood Size (nh-size). We compute the edit distance between a generated nonword and all words from a dictionary with known English words. We then rank the candidates according to the number of English words with low edit distance ($k = 1$ in our case). This means that nonwords having more orthographic neighbors are being ranked higher, which is a simple approximation for the probability that a learner confuses a nonword with a known word from the lexicon [5].

Character Language Model. This set of ranking methods is motivated by the observation that words in a language contain certain characteristic character combinations that make them look like a valid word of that language. For example, the word *großzügig* might look vaguely German to you even if you don't speak German.[1] This fact is also used in language identification where character language models are frequently used in order to distinguish languages [4,17]. We are going to use character language models with the goal to find nonwords like *platery* that look English, but actually are not part of the lexicon. We experiment with unigram, bigram, and trigram models, but expect higher-order language models to work better. For ranking, we assign to each word its probability returned by the language model.

Position Specific. A drawback of the simple character language model is that it assigns equal probability to a character n-gram no matter where it appears in a word. However, it is clear that the trigram *ing* is more likely at the end of a word than at the beginning. We thus augment the simple model to include position specific information following [5].

As the importance of the first and last letters of each word for reading is well known [10], we break each string into three parts: *start*, *middle*, and *end*. Figure 3 shows an example of our split. For each part, we separately train and apply a position-specific character language model.

[1] It means *generous* in English.

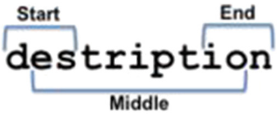


Fig. 3. Example for position specific splitting. Each part is scored with its own character language model.

3 Experimental Setup

In our experiments, we want to find the best ranking strategy, where we expect higher-order n-gram models to work better, and position specific language models to outperform corresponding simple models. We train all language models using the Brown Corpus [6]. We deliberately used a rather small corpus to show that character language models do not need much training data.

3.1 Evaluation Metric

In order to measure the quality of a ranking, we need to know whether word-like nonwords are ranked on the top positions. For that purpose, we are taking the 21 nonwords from LexTALE lexical recognition test [12] as a gold standard. They are known to be easily confused with real words, which means that a good ranking function should rank them at the top.

As evaluation metric, we are utilizing average precision (AP) from information retrieval. Table 1 gives an example showing two example rankings. Each time we find one of our gold standard nonwords from the LexTALE (LT) list, we

Table 1. Example for computing average precision (AP) for two different rankings. Whenever an LexTALE (LT) word is observed, precision P is computed for this subset.

Pos	Ranking #1	P	Ranking #2	P
1	LT	1.00	Nonword	-
2	Nonword	-	LT	0.50
3	LT	0.67	Nonword	-
4	LT	0.75	Nonword	-
5	Nonword	-	LT	0.40
6	Nonword	-	Nonword	-
7	Nonword	-	Nonword	-
8	Nonword	-	Nonword	-
9	Nonword	-	Nonword	-
10	Nonword	-	LT	0.30
	AP	**0.81**		**0.40**

compute the precision at that point taking only into account the items retrieved so far. For example in ranking #1, we find an LT nonword at the first position. As all items retrieved so far are LT nonwords, the precision is 1. The next LT nonword is on position 3. At this point, we have retrieved 2 LT items and 1 candidate nonword item which results in a precision of $^2/_3$. The third LT nonword in ranking #1 is found on position 4, for a precision of $^3/_4$. Average precision is now computed as the average over the three precision values. Computing the average precision in the same way for ranking #2 confirms that #1 is much better than #2.

3.2 Evaluation Dataset

In order to create the evaluation dataset, we generate 10,000 random nonwords with length between 4 and 11 letters (the same length limits as in LexTALE). We then add the 21 gold standard nonwords from LexTALE that are going to be used for evaluation. In order to smooth the results, we repeat the experiment 100 times (generating new random nonwords every time) and report mean average precision values.

Table 2. Average precision of ranking strategies

Strategy	Nonword length (characters)								
	All	4	5	6	7	8	9	10	11
Random	.00	.00	.00	.00	.00	.00	.00	.00	.00
nh-size	.02	.00	.07	.29	.53	.57	.57	.57	.57
1-gram	.02	.01	.01	.02	.03	.05	.09	.11	.14
1-gram-PS	.06	.04	.04	.04	.06	.07	.09	.11	.14
2-gram	.13	.04	.07	.14	.26	.41	.54	.64	.72
2-gram-PS	.43	.19	.26	.40	.53	.65	.73	.79	.83
3-gram	.30	.09	.19	.38	.54	.69	.78	.85	.90
3-gram-PS	.67	.41	.55	.67	.75	.81	.84	.87	.89

4 Results

Table 2 shows the average precision values for all nonwords in the dataset as well as per nonword length. From the table, we can see that the random baseline is close to zero showing that our dataset size of 10,000 candidates is large enough to avoid random strategies to have any effect. Neighborhood size does not work well in general, which is especially due to the bad performance on the shorter nonwords, while it works reasonably well on the longer ones. For the language model based approaches, we observe two trends which are in line with our hypotheses: (i) higher n-gram models work better, and (ii) position specific

Table 3. The top-10 LexTALE nonwords (LTs); top-10 and bottom-10 nonwords as per the ranking of 10 K randomly generated nonwords using 3-gram-PS approach.

LexTALE	Ranked nonwords	
Nonwords	Top-10	Bottom-10
Platery	Ahers	zlkcltmirk
Destription	Dand	ydbwehwve
Alberation	Whil	oumacivcgi
Mensible	Lign	dkucrxuvhvi
Interfate	Folli	lzurtqsrv
Proom	Golay	athfiprzbjq
Fellick	Poteru	qocbuabvh
Exprate	Alopirdrel	vnesfrqqjt
Rebondicate	Hindscomy	bgicpzycl
Purrage	Sherotspia	kcnkqpgt

models always work better than the simple model. Our best strategy is thus the 3-gram position specific ranking with an average precision of 0.67, which means that almost all gold standard nonwords are ranked very high among the 10,000 candidates. The breakdown of results per nonword length shows that longer nonwords are generally easier to rank which can be explained by the fact that the score of longer nonwords is more difficult to influence by a single very frequent n-gram.

In Table 3, we show some examples of the LexTALE nonwords that we use as a gold standard. We also show the top-10 as well as the bottom-10 candidates as ranked by our best strategy. The top-10 looks much more work-like compared to the bottom-10 showing that our ranking is effective, but compared with the gold standard LexTALE words, our generated nonwords seem to be of lower quality. However, this is only an informal evaluation and it is unclear whether the perceived difference will have any effect in an actual lexical recognition test. Thus, in the next section we formally compare our test with LexTALE in a user study.

5 User Study

The goal of the user study is to test how well our generated nonwords work in a lexical recognition test compared to an established test like LexTALE.

5.1 Selecting Words

For our test, we use the nonwords generated by our best strategy (3-gram-PS) as described above. However, besides nonwords, we also need a suitable set of known English words. Ideally, they should span the whole difficulty range from simple to sophisticated. We follow [12] who select words from different ranges of relative frequency in a large corpus, namely the CELEX corpus [1]. This makes use of the well established fact that there is a high correlation between the frequency of a word and its difficulty [7]. This also ensures a better comparability when the test is conducted for different languages [5].

We use the Brown corpus [6] in order to determine the relative frequency of words. We follow the LexTALE procedure and randomly select words with 4 to 12 letters[2] and a corpus frequency between between 1 and 26 occurrences per million words. We also make sure to select the same number of words from different word classes as in LexTALE. However, many English words have multiple word classes, so an exact mapping from out-of-context words into word classes is not possible anyway. The resulting list of words is shown in Table 4.

Table 4. Set of words used in our test categorized by word classes.

Class	Set
Nouns (15)	canto, hilt, quantum, leeway, barbell, vintage, allegory, fable, pallor, shovel, tavern, huddle, primacy, gadfly, syndicate
Adjectives (12)	intermittent, turbulent, appreciative, parasitic, snobbish, arrogant, lusty, exquisite, endurable, reverent, orchestral, septic
Adverbs (2)	lengthwise, precariously
Verbs (11)	mold, forfeit, veer, enrich, rape, intervene, expel, strut, buckle, blend, forestall

5.2 Setup

We asked participants to complete a three-part study: (i) a self-assessment of English language proficiency, (ii) the manually created LexTALE test, and (iii) our automatically generated test. We utilize Moodle[3] (a well-known learning management system) to conduct the study.

First, we provide participants with a set of instructions including some sample items. Then the participants were asked to provide information about gender, age, L1, the number of years they had taken English courses in school, and the self-rated language proficiency using Common European Framework of Reference (CEF)[4] levels. Finally, participants had to finish the LexTALE test and our test. In order to avoid sequence effects, participants randomly either get LexTALE first and then our test, or vice versa. However, we do not randomize the order of items within a test following the LexTALE guidelines.

[2] This is a different size compared to nonwords in LexTALE that are 4 to 11 letters long. In order to ensure comparability with LexTALE, we follow those length constraints, but newly generated tests should use the same constraints for words and nonwords.

[3] https://moodle.org.

[4] http://www.englishprofile.org/index.php/the-cef.

Scoring. There are several possible methods to score LRTs. We only want one combined score for word and nonword performance - in order to avoid test-wiseness effects, e.g. students answering that they know all the words. For each participant, we compute the test score using the scoring scheme introduced for LexTALE, as it turned out to yield the best results [12].

The score consists of the ratio of correct responses for words and nonwords – i.e. the recall for each class. This way, a yes bias (creating high error rates in the nonwords) would be *penalized* in the same way as a no bias (causing high error rates for words), independently of the different numbers of words versus nonwords. In order to yield a single score, the two recall values are averaged:

$$score(R) = \frac{(R_w + R_{nw}) \cdot 100}{2} \tag{1}$$

where R_w is the recall on words and R_{nw} on nonwords.

5.3 Study Results

We recruited 80 participants from two German universities, but only 45 finished all three parts of the study. 23 are female, 28 are German native speakers, and the average age is 22.4 years.

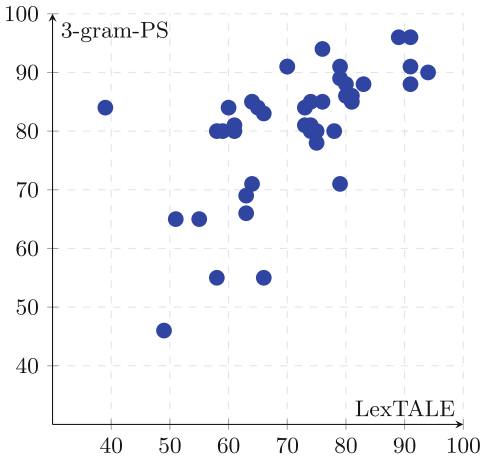


Fig. 4. Participants' scores on original LexTALE test vs. the test generated by our approach. Original scoring function.

In order to compare the quality of our test with the original LexTALE test, we compute for each student the test score according to formula (1) and then compute Spearman correlation ρ between the resulting score vectors for both tests. We obtain a correlation of 0.68 and Fig. 4 shows the corresponding scatterplot. We see that our test assigns vocabulary proficiency scores close to the ones assigned by LexTALE, but that there are some outliers.

In order to further analyze the differences between the two tests, we show a breakdown of recall for correctly detecting words vs. correctly rejecting nonwords in Table 5. We see that the recall for words is almost the same for both tests (.70 vs. .73), while our nonwords are much easier to recognize (.90 recall) compared to LexTALE (.75). This indicates that our nonwords do have lower quality compared to the LexTALE nonwords, as we suspected in Sect. 4. Interestingly this has little effect on the words, i.e. they do not get easier even if the nonwords are easier. This is probably due to the fact that nonwords do only need to be of reasonable quality in order to force students to make mistakes on the words.

We can conclude that in the light of the high correlation between the two tests, our automatically generated test is as effective as the manually created LexTALE in measuring the vocabulary proficiency level of learners.

Table 5. Recall of student responses for words a_w and nonwords a_{nw}.

LexTALE		3-gram-PS	
R_w	R_{nw}	R_w	R_{nw}
.70	.75	.73	.90

Self Ratings. Lemhöfer and Broersma [12] provide a partial mapping from the test score to CEFR levels, where scores equal to 59 points and below are mapped to B1 (or lower), scores between 60 and 79 points are mapped to B2, and scores above 80 points are mapped to C1 (or higher). The mapping is only partial because the test is not able to distinguish well for very early and very advanced stages of learning. We map the LexTALE scores and our scores to CEF levels. In 73% of all cases, both test agree on the same level (with a 33% chance of random agreement).

CEFR levels also allow us to compare with the self ratings, but this needs to be taken with a grain of salt, as we have no way of knowing how accurate the self ratings actually are. LexTALE assigns the self rated level in 40% of all cases, compared to 49% for our test showing again that both tests behave quite similar.

6 Related Work

The two main paradigms for creating nonwords (either manually or automatically) are (i) to start from a known word and change it to get a nonword, or (ii) to use smaller units (letters, syllables) to construct a larger nonword string.

The first paradigm was followed by the English Lexicon Project[5] [2], where they constructed a nonword database by manually changing one or more letters starting with known English words.

[5] http://elexicon.wustl.edu.

An example for the second paradigm is the ARC nonword database [15] that contains monosyllabic nonwords which follow the phonotactic and orthographic rules of (Australian) English. The database only provides the nonwords, but does not rank them according to their quality. Another approach is WordGen [5], which is an interactive tool for generating nonwords. It supports both paradigms and lets the user manipulate nonword properties that are similar to the ones we use for ranking, e.g. neighborhood size, position specific bigram frequency etc. In the end, the user is supposed to pick suitable nonwords, while our approach is fully automatic. Wuggy [11] builds on WordGen but introduces syllable template to build nonwords that more closely resemble a certain word.

All those approaches are more geared towards psycholinguistic research letting researchers select suitable nonwords or generate nonwords that are similar to a given word. In contrast our approach is supposed to work fully automatic and to create a new list of high quality nonwords whenever a lexical recognition test needs to be conducted.

7 Conclusion

We have tackled the task of automatically generating nonwords for lexical recognition tests. We show that character language models can be used to distinguish low and high quality nonwords, and that higher-order models incorporating position specific information work best. We evaluate the generated nonwords in a user study showing that our approach yields test scores that are highly correlated with the scores obtained from an established lexical recognition test. The study also shows that the difficulty of the nonwords has little effect on how well words are recognized. Nonwords only act as distractors forcing students to make mistakes on the words. With our experiments, we have shown that lexical recognition tests for English can be fully automatically created.

References

1. Baayen, R.H., Piepenbrock, R., Gulikers, L.: The Celex Lexical Database (Release 2). Linguistic Data Consortium, Philadelphia (1995)
2. Balota, D.A., Yap, M.J., Hutchison, K.A., Cortese, M.J., Kessler, B., Loftis, B., Neely, J.H., Nelson, D.L., Simpson, G.B., Treiman, R.: The English lexicon project. Behav. Res. Methods **39**(3), 445–459 (2007)
3. Brysbaert, M.: LexTALE_FR a fast, free, and efficient test to measure language proficiency in French. Psychol. Belg. **53**(1), 23–37 (2013)
4. Cavnar, W.B., Trenkle, J.M., et al.: N-gram-based text categorization. Ann. Arbor. MI **48113**(2), 161–175 (1994)
5. Duyck, W., Desmet, T., Verbeke, L.P., Brysbaert, M.: Wordgen: a tool for word selection and nonword generation in dutch, english, german, and french. Behav. Res. Methods Instrum. Comput. **36**(3), 488–499 (2004)
6. Francis, W.N., Kuçera, H.: Manual of Information to Accompany a Standard Corpus of Present-day Edited American English, for use with Digital Computers. Brown University, Providence (1964)

7. Greenberg, J.H.: Some generalizations concerning initial and final consonant sequences. Linguistics **3**(18), 5–34 (1965)
8. Huibregtse, I., Admiraal, W., Meara, P.: Scores on a yes-no vocabulary test: correction for guessing and response style. Lang. Test. **19**(3), 227–245 (2002)
9. Izura, C., Cuetos, F., Brysbaert, M.: Lextale-esp: a test to rapidly and efficiently assess the spanish vocabulary size. Psicol. Int. J. Methodol. Exp. Psychol. **35**(1), 49–66 (2014)
10. Johnson, R.L., Eisler, M.E.: The importance of the first and last letter in words during sentence reading. Acta Psychol. **141**(3), 336–351 (2012)
11. Keuleers, E., Brysbaert, M.: Wuggy: a multilingual pseudoword generator. Behav. Res. Methods **42**(3), 627–633 (2010)
12. Lemhöfer, K., Broersma, M.: Introducing lextale: a quick and valid lexical test for advanced learners of english. Behav. Res. Methods **44**(2), 325–343 (2012)
13. Meara, P., Jones, G.: Tests of vocabulary size in english as a foreign language. Polyglot **8**(1), 1–40 (1987)
14. Nation, P.: Teaching and Learning Vocabulary. Newbury House, Rowley (1990)
15. Rastle, K., Harrington, J., Coltheart, M.: 358,534 nonwords: the arc nonword database. Q. J. Exp. Psychol. Sect. A **55**(4), 1339–1362 (2002)
16. Schmitt, N.: Vocabulary in Language Teaching. Ernst Klett Sprachen, Stuttgart (2000)
17. Vatanen, T., Väyrynen, J.J., Virpioja, S.: Language identification of short text segments with n-gram models. In: LREC. Citeseer (2010)
18. Wang, T.H.: What strategies are effective for formative assessment in an e-learning environment? J. Comput. Assist. Learn. **23**(3), 171–186 (2007)

Teaching Words in Context: Code-Switching Method for English and Japanese Vocabulary Acquisition Systems

Michal Mazur[1](✉), Rafal Rzepka[2], and Kenji Araki[2]

[1] Institute for the Advancement of Higher Education,
Hokkaido University, Sapporo, Japan
mazzi@high.hokudai.ac.jp
[2] Graduate School of Information Science and Technology, Hokkaido
University, Sapporo, Japan

Abstract. One of the essential parts of second language curriculum is teaching vocabulary. Until now many existing techniques tried to facilitate word acquisition, but one method which has been paid less attention to is code-switching. In this paper, we present an experimental system for computer assisted vocabulary learning in context using a code-switching based method, focusing on teaching Japanese vocabulary to foreign language learners. First, we briefly introduce our Co-MIX method for vocabulary teaching systems using code-switching phenomenon to support vocabulary acquisition. Next, we show how we utilize incidental learning technique with graded readers to facilitate vocabulary learning. We present the systems architecture, underlying technologies and the initial evaluation of the system's performance by using semantic differential scale. Finally, we discuss the evaluation results and compare them with our English vocabulary teaching system.

Keywords: Computer assisted e-learning · Code-switching
Teaching vocabulary · Natural language processing

1 Introduction

With the increasing demand from a global society, learning foreign languages has become a necessary skill in the modern world for cross-cultural communication. This increasing trend brings the problem of finding qualified teachers that can give students a chance to practice their language skills, and while coming across the right person is relatively easy in a big city, people living in rural or remote areas generally have less chance of finding qualified educators. The most essential part of second language (L2) learning is vocabulary (Ling 2010). The problem of vocabulary acquisition is one of the most significant issues that must be faced to master foreign languages.

In a spoken language, about 1,800 words constitute about 80% of the spoken corpus (McCarthy 2004). Because some frequent words are often repeated, it is thought that learners can understand a large proportion of foreign language conversation with a relatively small vocabulary (McCarten 2007). The importance of vocabulary is

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 332–345, 2018.
https://doi.org/10.1007/978-3-319-93782-3_24

explained by Wilkins, who states: “Without grammar very little can be conveyed, without vocabulary nothing can be conveyed.” (Wilkins 1972).

In our research, the main scientific question we address is whether the code-switching phenomenon can expand students’ vocabulary and enable them to understand words without given definitions. This research presents work on implementation of code-switching method, an innovative learning approach that can give students a chance to expand their lexical knowledge and utilize the advantages of the code-switching phenomenon, in an e-learning application. To the author’s best knowledge there exists no robust methodology or teaching application for code-switching based vocabulary acquisition in e-learning.

2 Code-Switching

In recent years, many approaches to vocabulary learning have been presented. New possibilities come with the code-switching phenomenon, in which a word from one language is used in a sentence in which the grammatical structure belongs to another one. Code-switching presents a chance for students to think about second language words in a deeper manner and offers the potential for expansion of its vocabulary. Recently this technique has become a target of interest for researchers in different domains, like linguists (Coady and Huckin 1997) and computer scientists (Labutov and Lipson 2014). It can be defined as “the alternation of two languages within a single discourse, sentence or constituent” (Jamshidi and Navehebrahim 2013). Generally, code-switching can be explained as a brief insertion of a word from one language into another. For example: “The TENKI[1] forecast is great for the coming days.” (‘The WEATHER forecast is great for the coming days.’) - Japanese word within an English sentence. There are also two possible locations where the switch can occur, i.e. intersentential (after the sentence) and intrasentential (within the sentence).

Most people using code-switching is bilingual and able to speak two languages. They can easily code-switch and use it to find better ways to convey different meanings. However, there is an increasing amount of evidence that code-switching can also be used by less proficient people to fill linguistic gaps whenever the learner of a second language encounters the problem of insufficient vocabulary (Nishimura 1995). These two aspects create a common distinction between code-switching as an asset for bilingual people with a high competence in both languages, and as a “reparation tool for insufficiency in the second language” (Hamers and Blanc 2000).

In the last two decades of research, there seems to be an agreement on the positive effects of code-switching in learning and that it could be applied in second language learning to strengthen the learning outcome (Celik 2003; Jamshidi 2013; Lin 2013). There is a common agreement that when code-switching is planned beforehand, it can greatly contribute to more efficient understanding of a given topic. These effects of code-switching have been studied by Cook (2001) who also found that vocabulary

[1] From here onward romanized transliterations of Japanese are italicised. English translations in parentheses accompany the transliterations when needed.

learning can be facilitated by mixing two languages. Another interesting finding is demonstrated in the work presented by Lin (2013). According to this study, code-switching may increase the amount of cognitive processing by students. A larger cognitive effort is necessary to process new vocabulary, and students will learn new words more thoroughly. Although Lin's research did not show conclusive proof for higher effectiveness of code-switching for learning vocabulary, it indicated that code-switching does not affect the vocabulary acquisition in a negative way.

3 Related Works

To the authors' best knowledge there exists no robust methodology or teaching application for code-switching based vocabulary acquisition in e-learning. The problem of automatic creation of study materials for English classes has been approached by Ginsburg (2012), who proposed a learning program to generate editable English lesson materials to speakers of Japanese. The author underlines that higher vocabulary skills are crucial to understand the words while using this application. He also provides reading exercises with highlighted lexical units and their dictionary definitions. We follow this general idea of an exercise generator by developing a new method for assessing the vocabulary proficiency level of the user.

We also utilize an idea from a previous study by Mazur et al. (2012) regarding the method of increasing user engagement by introducing various language quizzes with the purpose of increasing users' motivation to study. The novelty of our method is found in combining the phenomenon of code-switching with several other existing methods of teaching vocabulary that have been proved to be effective, and implementing them in e-learning software to facilitate second language vocabulary acquisition.

4 Research Background

The theoretical background of our research comes from many sources. We benefit from a study by Nation (2001) that brings an insight on how considerate amount of L2 vocabulary acquisition happens by incidental learning and long-term vocabulary growth is greater by incidental learning method. Much of vocabulary is learned gradually through multiple exposures to a word over time. Another idea that contributed to our research originated in a study by Nagy and Herman (1985). This study claims that extensive reading should be promoted, because it can lead to greater vocabulary growth.

Further research by Nagy et al. 1987) also showed that even one or two exposures to a target word can contribute to a students' initial vocabulary knowledge. Krashen believes that language learners can learn new vocabulary and spelling more efficiently by receiving comprehensible input while they read, and postulated this concept in the Input Hypothesis (Krashen 1985). Labutov and Lipson (2014) claim that humans tend to be good at inferring meaning and that single words can be understood from the context of the surrounding text in the paragraph.

Further proof for the potential effectiveness of code-switching for language acquisition comes from the recent findings of Borovsky et al. (2012). This research contributes evidence that even a single exposure to a novel word in a constrained context results in the integration of the word to the reader's personal semantic base. It is contrary to a prior belief that a given word must be repeated with a sufficiently high frequency, and shows that words can be remembered incidentally while reading. Coady and Huckin (1997) have stated that many adult L2 learners are acquiring new words through reading to improve language proficiency.

5 Teaching Vocabulary

5.1 Methodology

There are many existing methods of teaching second language vocabulary. Teaching vocabulary mostly involves making students remember new language units by presenting words many times before students can successfully memorize them. Repetition is one of the popular techniques, along with actively recalling a word to be learned. Coady and Hucking (1997) claim that acquiring a substantial vocabulary in the second language is necessary to attain competencies in other language skills, such as listening, speaking or writing. This research follows Krashen's belief that "language learners acquire vocabulary and spelling more efficiently by receiving comprehensive input while reading" (Krashen 1985). Therefore, it can be asserted that new vocabulary units should not be presented only with their first language equivalents and without the presence of second language context.

We follow this general idea with our graded readers approach (see Subsect. 5.2). Gibbons's research (Gibbons 1987) indicates that nouns are the most often code-switched parts of speech. As many single lexical item switches are nouns, in our study we decided to utilize this finding to make the best use of the code-switching phenomenon. Further evidence of the importance of nouns is presented in the studies by Vihman (1998), who also proposed that nouns are more commonly used for the embedded language, because translations to the second language are often provided for them.

5.2 High Frequency Words

Our systems measure students' second language ability and adjust the difficulty of a given task to their level. They offer the opportunity for incidental learning using graded readers - texts graded according to high frequency word count, designed to give learners of a language practice in reading and used to support the extensive reading approach. Graded readers are comprehensible learning materials with the high frequency word count. These words are very important because they account for a large proportion of the content in written language. High frequency words are linguistically adjusted to learners' L2 competence and therefore can be used by students on different levels of proficiency.

Knowledge of about 4,000 words usually covers 86.0% of a given text (Nation 2001). Therefore, both teachers and learners should spend considerable time on their acquisition. We facilitate vocabulary acquisition through repetition of target high frequency words through a series of tests (quizzes) and presenting comprehensible examples of their usage. If there is a delay between the presentation of a word form and its meaning, students have an opportunity to try to guess or recall its meaning. Nation (2001) states that guessing can only be successful if we provide a good clue to the meaning (incidental learning). Our method offers the chance to experience both incidental vocabulary acquisition and direct teaching - two popular ways of learning high frequency words.

6 Co-MIX Method

Co-MIX is a vocabulary teaching method for e-learning systems that harnesses the benefits of code-switching, incidental learning and advantages of learning vocabulary in context. Until now, we implemented it in two systems: Co-MIX English, for facilitating English vocabulary acquisition (Mazur et al. 2016), and currently presented Co-MIX Japanese (from now on also referred to as proposed system) aimed at users who want to study Japanese. The modular architecture allows easy extensions of the system with additional functionalities.

Our systems consist of two modules. The first one is responsible for the automatic generation of study materials based on given graded readers in L1 (Fig. 1). The other module, the Language Quiz Generator, checks the users' understanding of L2 vocabulary through a series of language tests. The system also responds to users' mistakes by providing them with example sentences that show the correct usage of a given word. Since each user possesses a different level of vocabulary knowledge, in the first step the Co-MIX systems evaluate user language competency (see Subsect. 6.2) and present a study article appropriate to the user's vocabulary knowledge level. The text is used to introduce new vocabulary units.

Both systems are provided with a selection of graded articles in Japanese for the Co-MIX English system, and their translations into English used by Co-MIX Japanese and baseline systems. Graded articles are written and sorted by different levels of difficulty on a scale from 1 to 5, where 5 is the easiest and 1 is the most difficult. These readers help users to learn systematically by introducing them to easier language before moving up to more difficult tasks.

The baseline system is a variation of our proposed system without the code-switching module for experimental purposes. It is equipped with some commonly used vocabulary learning techniques (see Subsect. 5.1), however, it lacks the code-switching method presented in the proposed system. Another difference comes from the fact that instead of using readers in a mixed language with code-switching, the baseline system presents reading exercises entirely in L2 using the set of articles manually translated and proofread by a native speaker.

6.1 Study Material Generator

The main function of this application is to automatically create editable learning materials from a given text file. The study material generator is fed with text files with

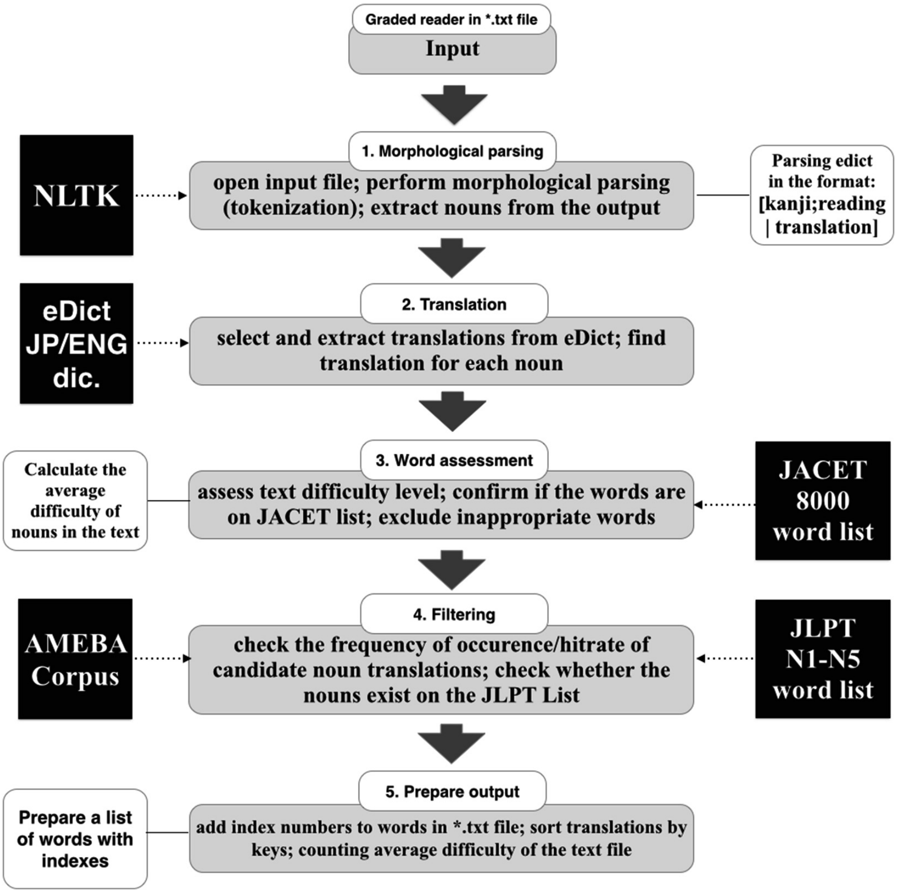

Graded reader in *.txt file
Input
NLTK
1. Morphological parsing
open input file; perform morphological parsing (tokenization); extract nouns from the output
Parsing edict in the format: [kanji;reading | translation]
eDict JP/ENG dic.
2. Translation
select and extract translations from eDict; find translation for each noun
Calculate the average difficulty of nouns in the text
3. Word assessment
assess text difficulty level; confirm if the words are on JACET list; exclude inappropriate words
JACET 8000 word list
AMEBA Corpus
4. Filtering
check the frequency of occurence/hitrate of candidate noun translations; check whether the nouns exist on the JLPT List
JLPT N1-N5 word list
Prepare a list of words with indexes
5. Prepare output
add index numbers to words in *.txt file; sort translations by keys; counting average difficulty of the text file

Fig. 1. Process of study material generator

graded readers in a selected language and automatically generates an editable output file with tagged vocabulary items (nouns), their translations in second language for the proposed system, and their positions on the JACET 8000 vocabulary list (Uemura and Ishikawa 2004). In case of the proposed system we use the JLPT vocabulary list[2]. Vocabulary lists are used to assess users' language ability and their prior knowledge. Our systems are using standardized lists of over 8,000 JLPT words[3] required for the

[2] The Japanese Language Proficiency Test (JLPT) is a standardized test to evaluate and certify the Japanese language proficiency of non-native speakers. "The JLPT is offered in five levels (N1, N2, N3, N4, N5). To measure Japanese-language proficiency as thoroughly as possible, test items are designed for each level". JLPT Homepage, https://www.jlpt.jp/e/about/points.html, last accessed 2014/10/1.

[3] This selection is based on a database of about 18,000 vocabulary words that have appeared on the test with high frequency over the past 20 years.

exam across all 5 levels. Both lists provide the average difficulty of a text, which allows the system to categorize the text to an appropriate proficiency level.

Usually, preparing suitable reading materials for students presents a challenge for teachers. The study material generator can be used to manually create additional learning materials from given input texts. The process of the study material generator is as follows. First program searches the input file for a selected part of speech (e.g. noun) and checks if their English translations are available in the dictionary. Then the program analyzes the word rankings in each reading and an average difficulty level is provided for the whole article.

In the next step, the system generates an output text with selected words preceded by an index number and supplemented with a list of all chosen words, their proposed translations and their difficulty ranking. In this step, we allow the human teacher to review the result and eliminate possible mistakes that may be caused by limitations of the available list word coverage. This is a necessary step before handing over the materials to the language quiz generator and presenting them to users. A few search rules had to be established. In this study we focused nouns, which have an available translation in the dictionary, have a translation that consists of no more than three words, are on our frequency list and are not on the list of the first 500 most common words that students are most likely to know.

6.2 Language Quiz Generator

The language quiz generator utilizes reading materials prepared by the study material generator (Fig. 2). In the first step, it checks users' vocabulary level and provides them with reading exercises corresponding to their language level. It also generates quizzes to check the understanding of given vocabulary units after completing the selected reading exercise.

The program works in the following manner: the user is presented with a set of 20 polar (yes/no) questions to determine the initial level of language proficiency. Next, the user is assigned to one of the five levels of language proficiency according to his/her final test score. We used the JLPT scale, which corresponds with the JACET 8000 vocabulary list (Japan Association of College English Teachers List of 8000 Basic Words) described by Uemura and Ishikawa (2004). Based on this selection, an appropriate text is selected and presented to the user in English with randomly selected nouns replaced with Japanese words.

After finishing the reading exercise, users are presented with a set of a maximum of 20 test questions. The number of test questions depends on the user's language level, the difficulty of the selected reading, the number of available example sentences, and number of nouns extracted from the input file.

The main purpose of these quizzes is to bring new vocabulary into productive use and enhance its memorization. The objective of each question is to type the word in second language. In the first phase of questions, random letters (not more than three and depending on word length) are presented to the user. If the user provides the system with correct answers, the test ends and automatic grading is provided.

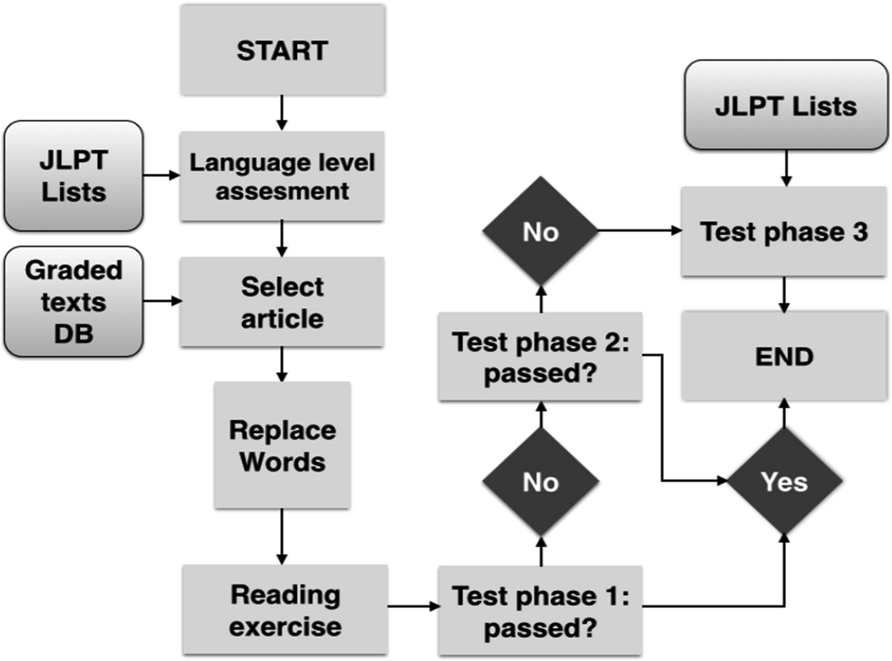


Fig. 2. Process of language quiz generator

In the case of error, the system searches for an example sentence with the proper use of a given word and presents it to the user. We use sentences from the Tanaka corpus[4], a large public-domain collection of parallel Japanese and English sentence pairs.

Next, mistaken words are repeated in the second phase of questions, where the difficulty level is slightly higher due to lack of hints and the necessity to write the target vocabulary in the sentence. Analogically to the second phase, the system repeats user mistakes in the third and final phase, where four words are presented and user must select the correct answer.

7 Evaluation

7.1 Participants

Sixteen test subjects (university students and adults) participated in a preliminary evaluation experiment and interacted with the proposed system (Co-MIX Japanese). Among the respondents 12 were male and four were female. Ten participants were in their twenties and six participants were over 30 years old.

[4] Tanaka Corpus of parallel Japanese-English sentences, http://www.edrdg.org/wiki/index.php/Tanaka_Corpus, last accessed 2014/10/1.

In this experiment, the target group were people who had some previous Japanese education, because presented method requires basic Japanese skills, such as reading ability (hiragana, katakana and some basic Japanese kanji characters). Therefore, we decided to perform experiments with university students and adults with comparable language ability (the number of years both groups had studied foreign language was similar).

We also selected 16 samples from baseline system experiment without code-switching to compare user attitudes towards baseline and proposed systems. Sixteen samples were selected to correspond with age, gender and education level of the proposed system users'. All participants were informed about the purpose of the study and the confidentiality of the results.

Ten participants were adults (eight company employees, one self-employed and one housewife) and six were university students (undergraduate and graduate). Ten respondents (62.5%) have been learning Japanese from 2 to 5 years, three have been learning from 1 to 3 years, one have been studying less than 6 months, one have been studying less than 3 months and one have been studying more than 5 years.

Seven respondents previously took the JLPT Exam (4 of them took Level 4, 2 of them level 5 and 1 of them Level 3). Thirteen participants also expressed their interest in taking JLPT exam in the future, two participants did not plan to take an exam, and one was not sure. Seven people were interested in taking Level 3, two participants wanted to start with Level 5, two participants were aiming at Level 2, one participant wanted to try Level 4 and one aimed at the highest Level 1.

7.2 Semantic Differential Scale

The participants of our experiment were asked to fill out a paper survey after they finished the learning tasks with a given system. To investigate users' attitudes towards the system, we used the semantic differential (SD) scale created by Osgood (1957).

The SD scale utilizes polar words pairs such as "good - bad", "positive - negative" and "valuable - worthless" to measure the meaning of a given concept. The position marked 0 is labeled "neutral," the 1 position is labeled "slightly," the 2 position "quite," and the third position "extremely."

In our experiment, we decided to use 14 bipolar words and a seven-point scale to derive subjects' attitudes towards the given concepts and measure their opinions on a metrically controlled scale. The bipolar words were determined based on the most typical pairs of opposite expressions used to evaluate the peoples' attitude towards a computer system. We selected SD scale as a conceptual model of assessing the value of e-learning system because it fits in the framework to measure overall users perceived satisfaction with an e-learning system and learners' behavior/usage of the system proposed by Levi (2006).

7.3 Experiment Results

The language level of each participant was determined by their number of correct answers in the initial vocabulary test. The average vocabulary level was 2.09 which corresponds to the Level 4, according to JLPT scale. In the test results only one person reached 100.0% correct answers in the first round, followed by 5 out of 15 in the second round (33.3%) and 7 out of 10 in the third round (70.0%).

The average answering time for the proposed system was 11.8 min. In case of the baseline system, 2 people out of 16 reached 100.0% correct answers in the first round (12.5%), followed by 4 out of 14 (28.5% of respondents) in the second round and 10 out of 10 (100.0%) reaching the maximum score in the third round. The average answering time for the baseline system was 8.8 min.

We collected the calculated average of system users' attitudes towards the Co-MIX Japanese system using the SD Scale. In Fig. 3., numbers closer to one represent the part of the spectrum assigned to the words on the left, whereas numbers close to seven are better characterized by the words on the right.

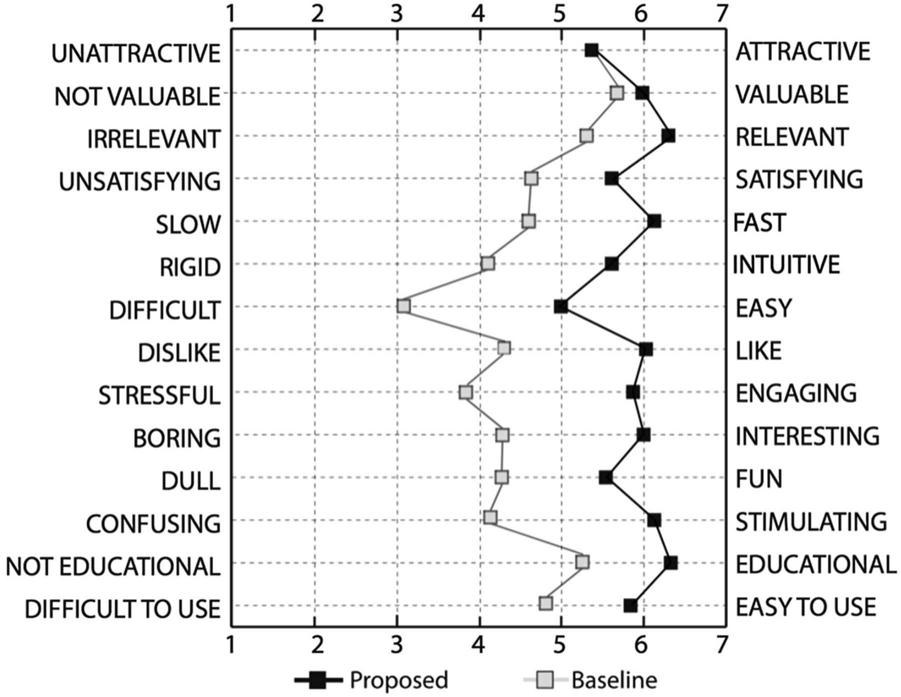


Fig. 3. Semantic differential scale for proposed and baseline systems

After analyzing the results, we reached the conclusion that the average results of the proposed system were distinctly higher than the baseline. The overall average of the proposed system was 5.8 points, which was superior to the baseline system (4.5 points). The general trend line for the proposed system stayed within 5/6 at the higher end of the scale, whereas items in the baseline system such as "like/dislike" and "interesting/boring" diverted from the given trend line. On the other hand, three items ("stimulating/confusing", "educational/not educational", "easy to use/difficult to use") had similar results in both systems, which indicates general user satisfaction with both systems.

7.4 Additional Experiment

Additionally, we compared results of the proposed system with the previous one designed for Japanese users to learn English vocabulary and described in a previous work by Mazur et al. (2016). The reason was to see how two variations of our method were perceived by users and to look for some possible similarities and differences.

Therefore, we asked a group of 16 participants with comparable age, gender and language ability to participate in an experiment with Co-MIX Japanese system. The results are presented on Fig. 4. and show some derivations of attitudes towards the given concepts for users of both systems.

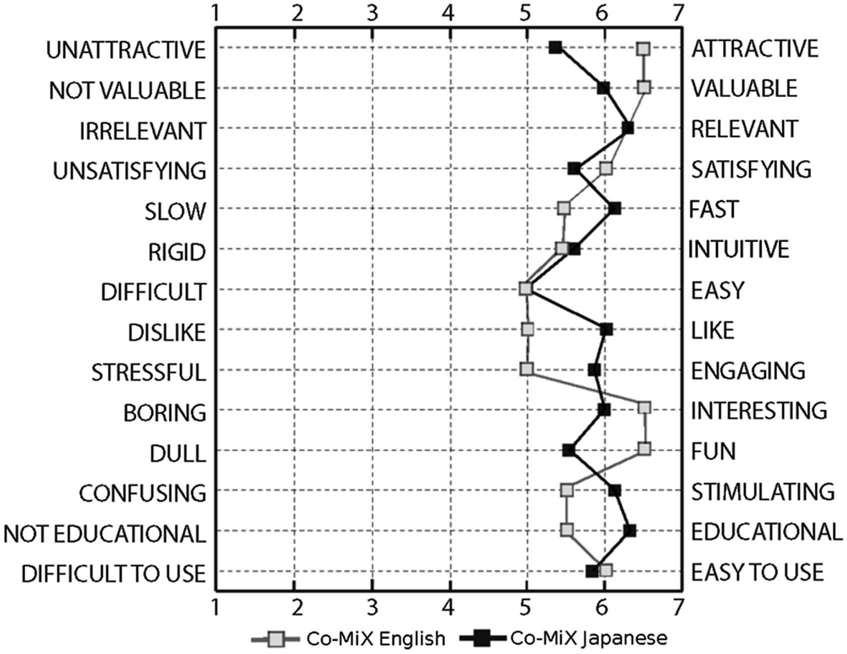


Fig. 4. Semantic differential scale for Co-MIX English and Co-MIX Japanese systems

The overall average of the proposed system was 5.8 points, which was slightly better than the Co-MIX Japanese system (5.6 points). Similar scores indicate that the participants considered both systems as relevant, easy to use, intuitive and generally satisfying.

However, proposed system was scored higher (6.5 points) in categories such as attractiveness, value and considered to be more interesting and fun, whereas Co-MIX English was indicated as more stimulating and educational. The difference may lie in a fact that English speaking users found the system more difficult and less approachable than the Japanese users.

The results of our survey indicate that one of the reasons for this difference may be that the Co-MIX Japanese uses kanji (Chinese characters) and some example sentences may be too difficult for the beginners. Higher attractiveness of the Co-MIX English system may be attributed to the fact that the Japanese users found our method engaging and in comments they pointed it out as an interesting and fun way to learn new vocabulary.

8 Conclusions and Future Work

In this paper we presented Co-MIX Japanese, a system based on the experimental method for second language vocabulary acquisition in context using a code-switching based approach for studying Japanese vocabulary. We propose an innovative learning method that gives students a chance to expand their lexical knowledge.

Further experiments confirmed the results of a previous study on a system for English vocabulary acquisition based on code-switching (Mazur et al. 2016). They demonstrated evidence that the proposed method is useful and effective way of expanding students' second language vocabulary.

The high percentage of correct answers, making successful connections between meanings of words in two languages and positive comments from the participants suggest that a system for learning English vocabulary in context using a code-switching based approach has the potential to provide learners of Japanese with engaging learning activities. The participants improved their vocabulary scores after using the application, which demonstrates the pedagogical benefits of the Co-MIX method. The extension to support another language was successful which supports our belief that this method can be easily implemented in other languages with a proper morphological analyzer, dedicated study materials (graded readers) and high frequency word lists.

Judging from the results, it can be asserted that user attitudes to our proposed system were positive and in most categories, it significantly outperformed the baseline system. Additional rankings, based on the systems' performances and users' preferences, also indicated the proposed system as more favorable. High user satisfaction with our proposed system also provides motivation to improve and refine it in many ways in the future.

In the next step, we aim to incorporate additional functionalities into the Co-MIX systems and advance the project towards our initial idea to provide a chatterbot able to process user queries in mixed languages. Such Co-MIX systems enabled with a chatterbot could teach users second language vocabulary using the presented code-switching method in a human-like conversation.

We are also planning to test our method on sentences with emotions expressed by emoticons. This would allow us to deal with shifting attitudes of users towards the system and respond to the problem of keeping users engaged in learning tasks and motivated to continue using the system in the future.

It is also planned to take into consideration suggestions provided by the experiment participants in their comments and extend the system to support other languages. The first step in this direction was already taken by introducing Co-MIX English. Previous experiments provide other evidence that presented code-switching-based method can be easily implemented in other languages with a proper morphological analyzer, dedicated study materials (graded readers) and high frequency word lists.

The next step will be to further compare the previous research on the Co-MIX Japanese system with the Co-MIX English system. The first step towards this goal has been done in this work and we discussed some differences between two variations of the Co-MIX systems for English and Japanese vocabulary acquisition.

Since our latest work on the system dedicated to aid English-speaking users in studying Japanese vocabulary and its initial results gave more implications on possible adaptation of this method into other languages, performing more experiments with a larger group of participants to compare both systems performance is planned. With the constant improvement of proposed method, we believe this approach to second language vocabulary acquisition in E-learning systems may bring even better results in the future.

References

Borovsky, A., Elman, J., Kutas, M.: Once is enough: N400 indexes semantic integration of novel word meaning. Lang. Learn. Dev. **8**(3), 278–302 (2012)

Celik, M.: Teaching vocabulary through code-mixing. ELT J. **57**(4), 361–369 (2003)

Coady, J., Huckin, T. (eds.).: Second Language Vocabulary Acquisition: A Rationale for Pedagogy. Cambridge University Press, Cambridge (1997)

Cook, V.: Using the first language in the classroom. Can. Mod. Lang. Rev./La Revue canadienne des langues vivantes **57**(3), 402–423 (2001)

Gibbons, J.: Code-mixing and Code Choice: A Hong Kong Case Study. Multilingual Matters, Clevedon (1987)

Ginsburg, J.: Automatic generation of English lesson materials for native speakers of Japanese. In: Proceedings of the 2012 Joint International Conference on Human-Centered Computer Environments, pp. 14–18 (2012)

Hamers, J.F., Blanc, M.H.: Bilinguality and bilingualism. Cambridge University Press, Cambridge (2000)

Jamshidi, A., Navehebrahim, M.: Learners use of code switching in the English as a foreign language classroom. Aust. J. Basic Appl. Sci. **7**(1), 186–190 (2013)

Krashen, S.D.: The Input Hypothesis: Issues and Implications. Longman, New York (1985)

Labutov, I., Lipson, H.: Generating code-switched text for lexical learning. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 562–571 (2014)

Levy, Y.: Assessing the Value of E-Learning Systems. pp. 93–94. IGI Global (2006)

Lin, A.: Classroom code-switching: three decades of research. Appl. Linguist. Rev. **4**(1), 195–218 (2013)

Ling, Z.: Effectiveness of L2 vocabulary learning and LT-WM. J. Xinjiang Univ. (Philos. Humanit. Soc. Sci.) **6**, 029 (2010)

Mazur, M., Rzepka, R., Araki, K.: Chatterbots with occupation-between non-task and task oriented conversational agents. Linguist. Cogn. Approaches Conversat. Agents **61**, 61–66 (2012)

Mazur, M., Karolczak, K., Rzepka, R., Araki, K.: A system for English vocabulary acquisition based on code-switching. Int. J. Distance Educ. Technol. (IJDET) **14**(3), 52–75 (2016)

McCarthy, M.J.: Touchstone: From Corpus to Course Book. Cambridge University Press, Cambridge (2004)

McCarten, J.: Teaching Vocabulary. Lessons from the Corpus. Lessons for the Classroom. Cambridge University Press, Cambridge (2007)

Nagy, W., Herman, P.: Incidental vs instructional approaches to increasing reading vocabulary. Educ. Perspect. **23**, 16–21 (1985)

Nagy, W.E., Anderson, R.C., Herman, P.A.: Learning word meanings from context during normal reading. Am. Educ. Res. J. **24**(2), 237–270 (1987)

Nation, I.S.: Learning Vocabulary in Another Language. Ernst Klett Sprachen, Stuttgart (2001)

Nishimura, M.: A functional analysis of Japanese/English code-switching. J. Pragmat. **23**(2), 157–181 (1995)

Osgood, C.E.: The Measurement of Meaning, no. 47. University of Illinois Press, Urbana (1957)

Uemura, T., Ishikawa, S.: JACET 8000 and Asia TEFL vocabulary initiative. J. Asia TEFL **1**(1), 333–347 (2004)

Wilkins, D.A.: Linguistics in Language Teaching, vol. 243. Edward Arnold, London (1972)

Vihman, M.: A developmental perspective on codeswitching: conversations between a pair of bilingual siblings. Int. J. Biling. **2**, 45–84 (1998)

Emotions, Decisions and Opinions

Automatic Extraction of Harmful Sentence Patterns with Application in Cyberbullying Detection

Michal Ptaszynski[1](✉), Fumito Masui[1], Yasutomo Kimura[2], Rafal Rzepka[3],
and Kenji Araki[3]

[1] Department of Computer Science, Kitami Institute of Technology, Kitami, Japan
{ptaszynski,f-masui}@cs.kitami-it.ac.jp
[2] Department of Information and Management Science,
Otaru University of Commerce, Otaru, Japan
kimura@res.otaru-uc.ac.jp
[3] Graduate School of Information Science and Technology, Hokkaido University,
Sapporo, Japan
{rzepka,araki}@ist.hokudai.ac.jp

Abstract. The problem of humiliating and slandering people through Internet, generally defined as *cyberbullying* (later: CB), has been recently noticed as a serious social problem disturbing mental health of Internet users. In Japan, to deal with the problem, members of Parent-Teacher Association (PTA) perform Internet Patrol – a voluntary work by reading through the whole Web contents to spot cyberbullying entries. To help PTA members we propose a novel method for automatic detection of malicious contents on the Internet. The method is based on a brute force search algorithm-inspired combinatorial approach to language modeling. The method automatically extracts sophisticated sentence patterns and uses them in classification. We tested the method on actual data containing cyberbullying provided by Human Rights Center. The results show our method outperformed previous methods. It is also more efficient as it requires minimal human effort.

Keywords: Cyberbullying detection · Natural language processing
Pattern extraction

1 Introduction

Contributions of information technology to preservation, support and development of public health are numerous. Some of the recent ones include analysis and prediction of the spread of epidemics, analysis of health data or construction

Previous version of this paper appeared in: *Proceedings of 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC'15)*, pp. 370–375.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 349–362, 2018.
https://doi.org/10.1007/978-3-319-93782-3_25

of biomedical ontologies. However, most of these contributions address physical sphere of public health. The mental or psychological part, although equally important, has been mostly disregarded. Recent years have brought to light a problem greatly impairing public mental health, often in young Internet users. It is the problem of cyberbullying, defined as exploitation of online means of communication, such as Internet forum boards, or social networks to convey harmful and disturbing information about private individuals, often children and students. Messages classifiable as cyberbullying, include ridiculing someone's personality, body type, or appearance, slandering or spreading rumors and insinuations. Some cases of cyberbullying lead the victims to self mutilation, suicides, or attacking their offenders. In USA, a great focus on this issue began in 2006 after a 13 year old girl committed suicide after receiving bullying messages on MySpace. Similar cases have been noticed in other countries, including Japan, on which this research is focused.

In Japan the problem has become serious enough to be noticed by the Ministry of Education [12]. In 2007 Japanese school personnel and members of Parent-Teacher Association (PTA)[1] have started monitoring activities under the general name Internet Patrol (later: net-patrol) to spot Web sites containing such inappropriate contents. However, the net-patrol is performed manually as a volunteer work. Countless amounts of data on the Internet make this an uphill task.

This situation motivated us to take up a long term project, in which we aim to contribute to solving the problem of cyberbullying. In the present research we aim at developing a solution which would help and ease the burden of the net-patrol members and create a net-patrol crawler automatically spotting cyberbullying entries on the Web and reporting them to appropriate organs. In this paper we specifically focus on developing a systematic approach to automatically detecting and classifying cyberbullying entries.

The outline of this paper is as follows. Firstly, we define the problem of cyberbullying and present some of the previous research related to ours. Next, we describe our method and the dataset used in this research. Finally, we explain the evaluation settings, thoroughly analyze and discuss the results.

2 Background

2.1 Cyberbullying – A Social Problem

The problem of harmful and offending messages on the Internet has existed for many years. One of the reasons such activities evolved was the anonymity of communication on the Internet, giving users the feeling that anything can go unpunished. Recently the problem has been officially defined and labeled as cyberbullying (CB). The National Crime Prevention Council states that CB happens "when the Internet, cell phones or other devices are used to send or post text or images intended to hurt or embarrass another person."[2].

[1] An organization composed of parents and school personnel.

[2] http://www.ncpc.org/cyberbullying.

Some of the first robust research on CB was done by Hinduja and Patchin, who performed numerous surveys about the subject in the USA [14]. They found out that the harmful information may include threats, sexual remarks, pejorative labels, or false statements aimed to humiliate others. When posted on a social network, such as Facebook or Twitter, it may disclose humiliating personal data of the victim defaming and ridiculing them personally.

In Japan, after a several cases of suicides of CB victims, Japanese Ministry of Education, Culture, Sports, Science and Technology (later: MEXT) considered the problem as serious and began a movement against it. In a manual for handling the CB cases [12], the Ministry puts a great importance on early detection of suspicious entries, especially on Social Networking Services (SNS) and informal school Websites, and distinguishes several types of cyberbullying noticed in Japan. These are:

1. Cyberbullying appearing on BBS forums, blogs and on private profile websites;
   (a) Entries containing libelous, slanderous or abusive contents;
   (b) Disclosing personal data of natural persons without their authorization;
   (c) Entries and humiliating online activities performed in the name of another person;
2. Cyber-bullying appearing in electronic mail;
   (a) E-mails directed to a certain person/child, containing libelous, slanderous or abusive contents;
   (b) E-mails in the form of chain letters containing libelous, slanderous or abusive contents;
   (c) E-mails send in the name of another person, containing humiliating contents;

In this research we focused mostly on the cases of cyber-bullying that appear on informal web sites of Japanese secondary schools. The latter are Web sites where school pupils exchange information about subjects or contents of tests, etc. However, it was noticed that such pages witness a rapid increase of CB toward pupils and even teachers [19]. Cases like that make other users uncomfortable using the Web sites and cause undesirable misunderstandings.

A movement of Internet Patrol (later: net-patrol) was founded to deal with the problem. Its participants are usually teachers and PTA members. Based on the MEXT definition of CB, they read through all Internet contents, and when they find a harmful entry they send a deletion request to the Web page administrator and report about the event to the Police.

Unfortunately, at present net-patrol is performed manually as a voluntary work. This includes reading the countless entries, deciding about their harmfulness, printing out or taking photos of the pages, sending deletion requests and reports to appropriate organs. The surveillance of the whole Web is an uphill task for the small number of net-patrol members. Moreover, the task comes with great psychological burden on mental health to the net-patrol members. With this research we aim to create a tool allowing automatic detection of CB on the Internet to ease the burden of net-patrol volunteers.

2.2 Previous Research

There has been a small number of research on extracting harmful information from the Internet. For example, [7] developed a dictionary of abusive expressions based on a large Japanese electronic bulletin board (BBS) *2channel*. In their research they labeled words and paragraphs in which the speaker explicitly insults other people with words and phrases like *baka* ("stupid"), or *masugomi no kuzu* ("trash of mass-mudia"). Based on which words appeared most often with abusive vocabulary, they extracted abusive expressions from the surrounding context.

[16] performed affect analysis of small dataset of cyberbullying entries to find out that distinctive features for cyberbullying were vulgar words. They applied a lexicon of such words to train an SVM classifier. With a number of optimizations the system was able to detect cyberbullying with 88.2% of F-score. However, increasing the data caused a decrease in results, which made them conclude SVMs are not ideal in dealing with frequent language ambiguities typical for cyberbullying.

Ikeda and Yanagihara manually collected a set of harmful and non-harmful separate sentences [6]. Based on word occurrence within the corpus they created a list of keywords for classification of harmful contents. However, they struggled with variations of the same expressions differing with only one or two characters, such as *bakuha* "blow up" and *baku–ha* "blooow up". All variations of the same expression needed to be collected manually, which was a weakness of this method.

Fujii et al. proposed a system for detecting documents containing excessive sexual descriptions using a distance between two words in a sentence [2]. They defined as harmful "black words" those in close distance to words appearing only in harmful context, rather than in both harmful and non-harmful context ("grey words").

Hashimoto et al. proposed a method for detecting harmful meaning in jargon [4]. In their method they assumed that the non-standard meaning is determined by the words surrounding the word in question. They detected the harmful meaning based on calculating co-occurrence of a word with its surrounding words.

Next, [11] proposed a method to automatically detect harmful entries, in which they extended the SO-PMI-IR score [18] to calculate relevance of a document with harmful contents. With the use of a small number of seed words they were able to detect large numbers of candidates for harmful documents with an accuracy of 83% on test data.

Later, [13] proposed an improvement to Matsuba et al.'s method. They used seed words from three categories (abusive, violent, obscene) to calculate SO-PMI-IR score and maximized the relevance of categories. Their method achieved 90% of Precision for 10% Recall. We used both of the above methods as a baselines for comparison due to similarities in used datasets and experiment settings. Unfortunately, method by [13], based on *Yahoo!* search engine API, faced a problem of a sudden drop in Precision (over 30 percentage-points) across two years, since being originally proposed. This was caused by change in information available on the Internet. In Sect. 4.5 we discuss the possible reasons for this change. Recently [3] tried to improve the method by automatically acquiring and filtering harmful seed words, with a considerable success.

In our research we aimed at minimization of human effort. Most of the previous research assumed that using vulgar words as seeds will help detecting cyberbullying. However, all of them notice that vulgar words are only one kind of distinctive vocabulary and do not cover all cases. We assumed that this kind of vocabulary could be extracted automatically. Moreover, we did not restrict the scope to words, (unigrams, tokens), or even phrases (ngrams). We extended the search to sophisticated patterns with disjoint elements. To achieve this we developed a pattern extraction method based on the idea of brute force search algorithm.

3 Method Description

We assumed that applying sophisticated patterns with disjoint elements should provide better results than the usual bag-of-words or n-gram approach. Such patterns can be defined as ordered combinations of sentence elements.

To extract such sophisticated patterns we applied a language modeling method based on the idea of language combinatorics [17]. This idea assumes that linguistic entities, such as sentences can be perceived as bundles of ordered non-repeated combinations of elements (words, punctuation marks, etc.). Furthermore, the most frequent combinations appearing in many different sentences can be defined as sentence patterns.

In this method, firstly, ordered non-repeated combinations are generated from all elements of a sentence. In every n-element sentence there is k-number of combination clusters, such as that $1 \leq k \leq n$, where k represents all k-element combinations being a subset of n. The number of combinations generated for one k-element cluster of combinations is equal to binomial coefficient. In this procedure the system creates all combinations for all values of k from the range of $\{1, ..., n\}$. Therefore the number of all combinations is equal to the sum of all combinations from all k-element clusters of combinations, like in Eq. 1.

$$\sum_{k=1}^{n} \binom{n}{k} = \frac{n!}{1!(n-1)!} + \frac{n!}{2!(n-2)!} + ... + \frac{n!}{n!(n-n)!} = 2^n - 1 \tag{1}$$

Next, all non-subsequent elements are separated with an asterisk ("*"). All patterns generated this way are used to extract frequent patterns appearing in a given corpus. Their occurrences O is used to calculate their normalized weight w_j according to Eq. 2. The score of a sentence is calculated as a sum of weights of patterns found in the sentence, like in Eq. 3.

$$w_j = \left(\frac{O_{pos}}{O_{pos} + O_{neg}} - 0.5 \right) * 2 \tag{2}$$

$$score = \sum w_j, (1 \geq w_j \geq -1) \tag{3}$$

The weight can be later calculated in several ways. Two features are important in weight calculation. A pattern is the more representative for a corpus when, firstly, the longer the pattern is (length k), and the more often it appears in the corpus (occurrence O). Thus the weight can be modified by

- awarding length (later: LA),
- awarding length and occurrence (later: LOA).

The list of generated frequent patterns can be also further modified. When two collections of sentences of opposite features (such as "positive" vs. "negative") are compared, a generated list of patterns will contain patterns that appear uniquely in only one of the sides (e.g. uniquely positive or negative patterns) or in both (ambiguous patterns). Therefore the pattern list can be further modified by

- erasing all ambiguous patterns (later: AMB),
- erasing only ambiguous patterns which appear in the same number in both sides (later zero patterns, or 0P).

Moreover, a list of patterns will contain both the sophisticated patterns (with disjoint elements) as well as more common n-grams. Therefore the experiments were performed either with patterns (PAT), or n-grams (NGR) only. If the initial collection of sentences was biased toward one of the sides (e.g., more sentences of one kind, or the sentences were longer, etc.), there will be more patterns of a certain sort. Thus to avoid bias in the results, instead of applying a rule of thumb, threshold is automatically optimized. The above settings are automatically verified in the process of evaluation (10-fold cross validation) to choose the best model. The metrics used in evaluation are standard Precision (P), Recall (R) and balanced F-score (F). Finally, to deal with the combinatorial explosion mentioned on the beginning of this section we applied two heuristic rules. In the preliminary experiments we found out that the most valuable patterns in language are up to six element long, therefore we limited the scope to $k \leq 6$. Thus the procedure of pattern generation will (1) generate up to six elements patterns, or (2) terminate at the point where no frequent patterns were found.

4 Evaluation Experiment

4.1 Dataset

At first we needed to prepare a dataset. We used the dataset created originally by [10] and developed further by [11]. The dataset was also used by [16] and recently by [13]. It contains 1,490 harmful and 1,508 non-harmful entries. The original data was provided by the Human Rights Research Institute Against All Forms for Discrimination and Racism in Mie Prefecture, Japan[3] and contains data from unofficial school Web sites and fora. The harmful and non-harmful sentences were manually labeled by Internet Patrol members according to instructions

[3] http://www.pref.mie.lg.jp/jinkenc/hp/.

included in the MEXT manual for dealing with cyberbullying [12]. Some of those instructions are explained shortly below.

The MEXT definition assumes that cyberbullying happens when a person is personally offended on the Web. This includes disclosing the person's name, personal information and other areas of privacy. Therefore, as the first feature distinguishable for cyberbullying MEXT defines private names. This includes such information as:

- Private names and surnames,
- Initials and nicknames,
- Names of institutions and affiliations,

As the second feature distinguishable for cyberbullying MEXT defines any other type of personal information. This includes:

- Address, phone numbers,
- Questions about private persons (e.g. "Who is that tall guy straying on Computer Science Dept. corridors?"),
- Entries revealing other personal information (e.g. "I hate that guy responsible for the new project against cyberbullying.").

Also, according to MEXT, vulgar language is distinguishable for cyberbullying, due to its ability to convey offenses against particular persons. This is also confirmed in other literature [14,16]. Examples of such words are, in English: *sh*t*, *f*ck*, or *b*tch*, in Japanese: *uzai* (freaking annoying), or *kimoi* (freaking ugly).

In the prepared dataset all entries containing any of the above information was classified as harmful. Some examples from the dataset are represented in Table 1.

4.2 Dataset Preprocessing

The language combinatorics method takes as an input sentences separated into elements (words, tokens, etc.). Therefore we needed to preprocess the dataset and make the sentences separable into elements. We did this in several ways to check how the preprocessing would influence the results. We used MeCab[4], a standard morphological analyzer for Japanese to preprocess the sentences from the dataset in the following ways:

- **Tokenization:** All words, punctuation marks, etc. are separated by spaces (later: `TOK`).
- **Parts of speech (POS):** Words are replaced with their representative parts of speech (later: `POS`).
- **Tokens with POS:** Both words and POS information is included in one element (later: `TOK+POS`).

[4] http://taku910.github.io/mecab/.

Table 1. Four examples of cyberbullying entries gathered during Internet Patrol. The upper three represent strong sarcasm despite of the use of positive expressions in the sentence. English translation below Japanese content. Harmful patterns recognized automatically – underlined (underlining in English was made to correspond as closely to Japanese as possible).

>>104 Senzuri koi te shinu nante? sonna hageshii senzuri sugee naa. "Senzuri masutaa" toshite isshou agamete yaru yo. >>104 Dying by 'flicking the bean'? Can't imagine how one could do it so fiercely. I'm gonna worship her as a 'master-bator', that's for sure.
2-nen no tsutsuji no onna meccha busu suki na hito barashimashoka? 1-nen no anoko desuyo ne? kimogatterunde yamete agete kudasai Wanna know who likes that awfuly ugly 2nd-grade Azalea girl? Its that 1st-grader isn't it? He's disgusting, so let's leave him mercifully in peace.
Aitsu wa busakute sega takai dake no onna, busakute se takai dake ya noni yatara otoko-zuki meccha tarashide panko anna onna owatteru She's just tall and apart of that she's so freakin' ugly, and despite of that she's such a cock-loving slut, she's finished already.
Shinde kureeee, daibu kiraware-mono de yuumei, subete ga itaitashii... Please, dieeee, you're so famous for being disliked by everyone, everything in you is so pathetic

The examples of preprocessing are represented in Table 2. Theoretically, the more generalized a sentence is, the less unique patterns it will produce, but the produced patterns will be more frequent. This can be explained by comparing tokenized sentence with its POS representation. For example, in the sentence from Table 2, we can see that a simple the phrase *kimochi ii* ("pleasant") can be represented by a POS pattern `N ADJ`. We can easily assume that there will be more `N ADJ` patterns than *kimochi ii*, because many word combinations can be represented by this pattern. On the other hand, there are more words in the dictionary than POS labels. Therefore POS patterns will come in less variety but with higher occurrence frequency. By comparing the result of the classification using different preprocessing methods we can find out whether it is better to represent sentences as more generalized or as more specific.

4.3 Experiment Setup

The preprocessed original dataset provides three separate training and test sets for the experiment (tokenized, POS-tagged and tokens with POS together). The experiment was performed three times, one time for each kind of preprocessing to choose the best option. For each version of the dataset a 10-fold cross validation was performed and the results were calculated using standard Precision, Recall and balanced F-score for the whole threshold span. In one experiment 14 different versions of the classifier are compared with 10-fold cross validation condition. Since the experiment is performed for three different versions of preprocessing, we obtained overall number of 420 experiment runs. There were several evaluation

Table 2. Three examples of preprocessing of a sentence in Japanese; N = noun, TOP = topic marker, ADV = adverbial particle, ADJ = adjective, COP = copula, INT = interjection, EXCL = exclamative mark.

Sentence: 今日はなんて気持ちいい日なんだ！
Transliteration: *Kyōwanantekimochiiihinanda!*
Meaning: Today TOP what pleasant day COP EXCL
Translation: What a pleasant day it is today!

Preprocessing examples

1. Tokenization: *Kyō wa nante kimochi ii hi nanda !*
2. POS: N TOP ADV N ADJ N COP EXCL
3.Tokens+POS: *Kyō*[N] *wa*[TOP] *nante*[ADV] *kimochi*[N] *ii*[ADJ] *hi*[N] *nanda*[COP] *!*[EXCL]

criteria. Firstly, we looked at which version of the algorithm achieved the top score within the threshold span. This is referred to as threshold optimization. However, theoretically, an algorithm could achieve its best score for one certain threshold, while for others it could perform poorly. Therefore we also looked at break-even points (BEP) of Precision and Recall. We calculated this as a sum of scores for all thresholds. This shows which version of the algorithm is more balanced thorough the whole threshold span. Finally, we checked the statistical significance of the results. We used paired *t*-test because the classification results could represent only one of two classes (harmful or non-harmful). To chose the best version of the algorithm we compared separately the results achieved by each group of modifications, eg., "different pattern weight calculations", "pattern list modifications" and "patterns vs n-grams". We also compared the performance to the baseline [13].

4.4 Results and Discussion

When it comes to Precision, the highest score of all was achieved by the feature sets: POS/NGR/LA (P = .93), POS/NGR, POS/NGR/OP (P = .92) and POS/NGR/LA/OP (P = .91). Unfortunately, all with low Recall (R = .02–.03). Despite these occasional top scores for Precision, the POS-tagged dataset achieved in general the lowest balanced F-score (up to F = .78).

Also high Precision with much higher Recall was achieved by feature sets: TOK+POS/PAT|NGR and TOK+POS/PAT|NGR/OP (P = .89, R = .34). The dataset preprocessing containing both tokens and POS tags also achieved the highest general results in balanced F-score (F = .8 for TOK+POS/PAT|NGR/OP and F = .79 for TOK+POS/PAT|NGR). Dataset which was only tokenized achieved moderate scores in general. From the fact that the general results ideally corresponded with the sophistication of preprocessing, we infer that the method could be further improved by more sophisticated preprocessing.

Tokenization with POS tagging also provided the highest scores when it comes to break-even point (BEP) of Precision and Recall. The highest scores were achieved by `TOK+POS/PAT|NGR` and `TOK+POS/PAT|NGR/OP` (P = .79, R = .79, F = .79). Since this corresponds to the best results in F-score, we consider the two feature sets as optimal. There were small differences in detailed results between these datasets, however, as they occurred statistically insignificant, we consider both of them as optimal. It could be further noticed that, since `TOK+POS/PAT|NGR/OP` uses less patterns (no zero-patterns), this feature set could be considered as more time-efficient.

When it comes to other modifications, in most cases deleting ambiguous patterns yielded worse results, which suggests that such patterns, despite being ambiguous to some extend (appearing in both cyberbullying and non-cyberbullying entries), are in fact useful in practice. Also, awarding pattern length or occurrence in weight calculation, although causing statistically significant differences in results, did not come with performance improvement.

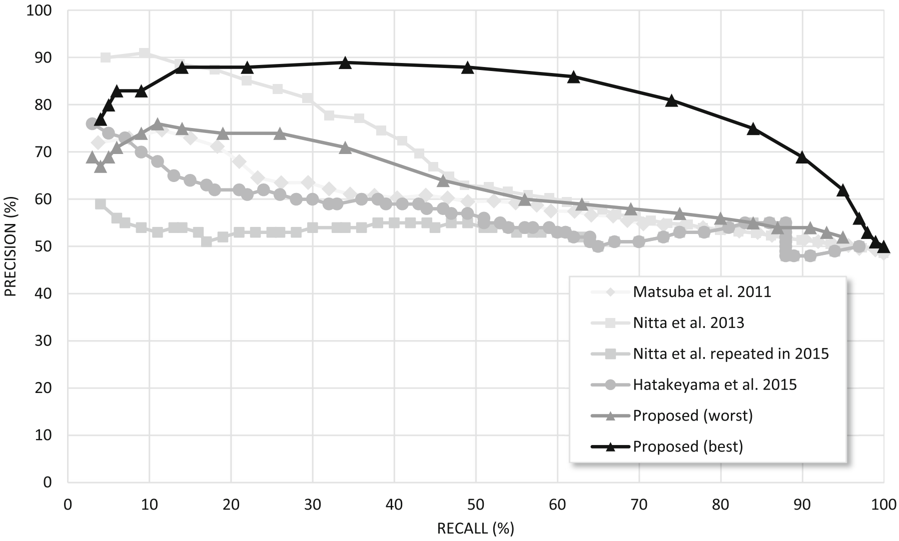


Fig. 1. Comparison between the proposed method (best and worst performance) and previous methods.

4.5 Comparison with Previous Methods

After specifying optimal settings for the proposed method, we compared it to previous methods. In the comparison we used the method by [11,13], and its most recent improvement by [3]. Moreover, since the latter extracts cyberbullying relevance values from the Web (in particular *Yahoo! API*), apart from comparison to the reported results we also repeated their experiment to find out how the performance of the Web-based method changed during the three years. Finally, to make the comparison more fair, we compared both our best and worst results. As the evaluation metrics we used area under the curve (AUC) on the graph showing Precision and Recall. The results are represented in Fig. 1.

The highest overall results when it comes to AUC were obtained by the best settings of the proposed method (tokens with POS, all patterns, no weight modification), which starts from a high 77% and retains the Precision between 80% and 90% for most of the threshold. Although the highest score was still by [13], performance of their method quickly decreases due to quick drop in Precision for higher thresholds. Moreover when we repeated their experiment recently in January 2015, the results greatly dropped. After thorough analysis of the experiment data we noticed that most of the information extracted in 2013 was not available in 2015. This could be due to the following reasons. Firstly, fluctuation in page rankings could push the information lower making it not extractable by Nitta et al.'s method. Secondly, frequent deletion requests of harmful contents by net-patrol members could make their efforts pay off. However, the most probable is the third cause, which is the recent tightening of usage policies by most Web service providers, such as Google[5], Twitter[6] and *Yahoo!* used by [13]. This includes recently introduced DMARC[7] policies related to e-mail spoofing and general improvements in policies aimed at decreasing Internet harassment. Such changes aimed at limiting the growing problem of Internet harassment, implemented on a corporate level, are in general a positive phenomenon, despite reducing the performance of cyberbullying detection software. Moreover, as was recently shown by [3], the performance can be to some extent improved by automatically optimizing the list of seed words applied in such methods.

However, The fact that the performance of Nitta et al.'s method decreased from over 90% to less than 60% during 3 years is an important warning for other research based on Web search engines. Probability of such problems have been indicated some time ago [8], and could become a major problem in the future. This also advocates more focus on corpus-based methods such as the one proposed in this paper.

Finally, while the numerical results were in favor of the proposed approach, we also wanted to know to what extent the patterns automatically recognized by the proposed method cover the manually selected seed words in the previous research [10,11,13]. In the result, all seed words appeared in the list of automatically extracted patterns. This can be interpreted as follows. Firstly, CB definition by [12] and hunch of the researchers, on which previous approaches were mostly based, were generally correct. Secondly, using our automatically extracted patterns it could be possible to improve previous approaches in the future.

[5] https://www.google.com/events/policy/anti-harassmentpolicy.html.

[6] https://blog.twitter.com/2014/building-a-safer-twitter.

[7] http://www.dmarc.org/.

Table 3. All results of experiments on traditional classifiers on all datasets; best classifier in **bold type font**.

			POS	TOK+POS	TOK
SVM	**linear**	**Precision**	.563	.768	.777
		Recall	.563	.766	.776
		F-score	.563	.766	**.775**
		Accuracy	.563	.766	.776
	plynomial	**Precision**	.553	.499	.263
		Recall	.545	.499	.513
		F-score	.528	.450	.348
		Accuracy	.545	.499	.513
	radial	**Precision**	.565	.753	.793
		Recall	.565	.747	.756
		F-score	.565	.746	.746
		Accuracy	.565	.747	.756
	sigmoid	**Precision**	.562	.746	.752
		Recall	.562	.736	.538
		F-score	.561	.733	.403
		Accuracy	.562	.736	.538
Naïve Bayes		**Precision**	.570	.671	.682
		Recall	.569	.669	.678
		F-score	.568	.668	.677
		Accuracy	.569	.669	.678
JRip		**Precision**	.553	.614	.603
		Recall	.553	.613	.603
		F-score	.553	.613	.603
		Accuracy	.553	.613	.603
J48		**Precision**	.566	.671	.675
		Recall	.566	.666	.672
		F-score	.566	.663	.669
		Accuracy	.566	.666	.672
kNN (k=1)		**Precision**	.544	.630	.630
		Recall	.543	.627	.628
		F-score	.542	.625	.626
		Accuracy	.543	.627	.628

* All results were averaged for harmful and non-harmful classification performed separately.

Lastly, we also performed additional experiments using traditional classifiers applied in previous research on cyberbullying detection, namely, SVM, Naive Bayes, JRip, J48 and kNN. The experiments with traditional classifiers were performed on Bag-of-Words language model, with *tf*idf* weighting scheme and under the 10-fold cross-validation condition. All results of those classifiers were always worse than the proposed method, optimized for each dataset. Therefore for each dataset the winning classifier was always the proposed method.

As an additional remark, for traditional classifiers the tendencies in results generally confirmed those achieved by the proposed method. The results were better for tokenized datasets, with or without parts-of-speech, and much worse for POS-only dataset. All results of experiments on traditional classifiers were represented in Table 3.

5 Conclusions and Future Work

In this paper we proposed a method for automatic detection of Internet forum entries that contain cyberbullying (CB) – contents humiliating and slandering other people. CB is a recently noticed social problem which influences mental health of Internet users, and might lead to self-mutilation and even suicide of CB victims.

In the proposed method we applied a combinatorial algorithm, resembling brute force search algorithms, in automatic extraction of sentence patterns, and used those patterns in text classification of CB entries. We tested the method on actual CB data obtained from Human Rights Center. The results show our method outperformed previous methods. It is also more efficient as it requires minimal human effort.

In the near future we plan to apply different methods of dataset preprocessing to find out whether the performance can be further improved and to what extent. We also plan to obtain new data to evaluate the method more thoroughly, and apply different classifiers. Finally, we plan to verify the actual amount of CB information on the Internet and reevaluate the method in more realistic conditions.

References

1. Belsey, B.: Cyberbullying: An Emerging Threat for the "Always On" Generation (2007). http://www.cyberbullying.ca/pdf/Cyberbullying_Presentation_Description.pdf
2. Fujii, Y., Ando, S., Ito, T.: Yūgai jōhō firutaringu no tame no 2-tango-kan no kyori oyobi kyōki jōhō ni yoru bunshō bunrui shuhō no teian (Developing a method based on 2-word co-occurence information for filtering harmful information). In: Proceedings of the 24th Annual Conference of The Japanese Society for Artificial Intelligence (JSAI2010), paper ID: 3D2-4, pp. 1–4 (2010). (in Japanese)
3. Hatakeyama, S., Masui, F., Ptaszynski, M., Yamamoto, K.: Improving performance of cyberbullying detection method with double filtered point-wise mutual information. In: Demo Session of the 2015 ACM Symposium on Cloud Computing 2015 (ACM-SoCC 2015), Kohala Coast, Hawaii, 27–29 August 2015
4. Hashimoto, H., Kinoshita, T., Harada, M.: Firutaringu no tame no ingo no yūgai goi kenshutsu kinō no imi kaiseki shisutemu SAGE e no kumikomi (Implementing a function for filtering harmful slang words into the semantic analysis system SAGE), IPSJ SIG Notes 2010-SLP-81(14), pp. 1–6 (2010). (in Japanese)

5. Hinduja, S., Patchin, J.W.: Bullying Beyond the Schoolyard: Preventing and Responding to Cyberbullying. Corwin Press, Thousand Oaks (2009)
6. Ikeda, K., Yanagihara, T.: Kakuyōso no chūshōka ni motozuku ihō-, yūgai-bunsho kenshutsu shuhō no teian to hyōka (Proposal and evaluation of a method for illegal and harmful document detection based on the abstraction of case elements). In: Proceedings of 72nd National Convention of Information Processing Society of Japan (IPSJ72), pp. 71–72 (2010). (in Japanese)
7. Ishisaka, T., Yamamoto, K.: 2chaeru wo taishō to shita waruguchi hyōgen no chūshutsu (Extraction of abusive expressions from 2channel). In: Proceedings of the Sixteenth Annual Meeting of The Association for Natural Language Processing (NLP2010), pp. 178–181 (2010). (in Japanese)
8. Kilgarriff, A.: Googleology is bad science. Comput. Linguist. **33**(1), 147–151 (2007)
9. Krippendorff, K.: Combinatorial explosion. In: Web Dictionary of Cybernetics and Systems. Principia Cybernetica Web (1986)
10. Matsuba, T., Masui, F., Kawai, A., Isu, N.: Gakkou hikoushiki saito ni okeru yuugai jouhou kenshutsu (Detection of harmful information on informal school websites). In: Proceedings of the 16th Annual Meeting of the Association for Natural Language Processing (NLP2010) (2010). (in Japanese)
11. Matsuba, T., Masui, F., Kawai, A., Isu, N.: Gakkō hi-kōshiki saito ni okeru yūgai jōhō kenshutsu wo mokuteki to shita kyokusei hantei moderu ni kansuru kenkyū (A study on the polarity classification model for the purpose of detecting harmful information on informal school sites). In: Proceedings of the Seventeenth Annual Meeting of the Association for Natural Language Processing (NLP2011), pp. 388–391 (2001). (in Japanese)
12. Ministry of Education, Culture, Sports, Science and Technology (MEXT): 'Netto-jō no ijime' ni kansuru taiō manyuaru jirei shū (gakkō, kyōin muke) ("Bullying on the Net" Manual for handling and collection of cases (for schools and teachers)). Published by MEXT (2008). (in Japanese)
13. Nitta, T., Masui, F., Ptaszynski, M., Kimura, Y., Rzepka, R., Araki, K.: Detecting cyberbullying entries on informal school websites based on category relevance maximization. In: Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP 2013), pp. 579–586 (2013)
14. Patchin, J.W., Hinduja, S.: Bullies move beyond the schoolyard: a preliminary look at cyberbullying. Youth Violence Juv. Justice **4**(2), 148–169 (2006)
15. Ptaszynski, M., Dybala, P., Rzepka, R., Araki, K.: Affecting corpora: experiments with automatic affect annotation system - a case study of the 2 channel forum -. In: Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING-09), pp. 223–228 (2009)
16. Ptaszynski, M., Dybala, P., Matsuba, T., Masui, F., Rzepka, R., Araki, K., Momouchi, Y.: In the service of online order: tackling cyber-bullying with machine learning and affect analysis. Int. J. Comput. Linguist. Res. **1**(3), 135–154 (2010)
17. Ptaszynski, M., Rzepka, R., Araki, K., Momouchi, Y.: Language combinatorics: a sentence pattern extraction architecture based on combinatorial explosion. Int. J. Comput. Linguist. (IJCL) **2**(1), 24–36 (2011)
18. Turney, P.D.: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, pp. 417–424 (2002)
19. Watanabe, H., Sunayama, W.: Denshi keijiban ni okeru yūza no seishitsu no hyōka (User nature evalution on BBS). IEICE Technical report, 105(652), 2006-KBSE, pp. 25–30 (2006). (in Japanese)

Sentiment Analysis in Polish Web-Political Discussions

Antoni Sobkowicz[2](✉) and Marek Kozłowski[1]

[1] National Information Processing Institute, Warsaw, Poland
[2] Warsaw University of Technology, Warsaw, Poland
antoni.sobkowicz@opi.org.pl

Abstract. The article presents analysis of Polish Internet political discussion forums, characterized by significant polarization and high levels of emotion. The study compares samples of discussions gathered from the Internet comments concerning the last Polish election candidates. The authors compare three dictionary based sentiment analysis methods (built using different sentiment lexicons) with two machine learning ones, and explore methods using word embeddings to enhance sentiment analysis using dictionary based algorithms. The best performing algorithm is giving results closely corresponding to human evaluations.

Keywords: Text classification · Sentiment analysis · Machine learning

1 Introduction

The crucial part of information acquisition has always been to find out what other people think. As the availability and popularity of opinion-rich resources such as web reviews and comments on web fora keeps growing, as more and more people start using information technologies in order to seek out and understand the opinions existing within society, new areas of study arise. Internet discussion fora are a very promising field for conducting research on human communication patterns, which encompass the content, timing, and emotional tone of the communication. Such discussions are rarely moderated, allowing for various kinds of expressions, ranging from elaborate texts to simple phrases full of emoticons.

The growing relevance of political communication in social media, particularly microblogging, implies a fundamental change in traditional political communication, which has usually been exclusively initiated and managed by political actors as well as journalists. However, as this field is relatively young, more research is needed to better understand the principles of communication on social media platforms.

We hasten to point out that consumption of goods and services is not the only motivation behind people's seeking out or expressing opinions online. A need for political information is another important factor. For example, Rainie and Horrigan [15] studied a sample of over 2500 American adults from the 31% of Americans—over 60 million people—that were 2006 campaign internet users,

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 363–377, 2018.
https://doi.org/10.1007/978-3-319-93782-3_26

defined as those who gathered information about the 2006 elections online and exchanged views via email. 28% said that a major reason for these online activities was to get perspectives from within their community, and 34% said that a major reason was to get perspectives from outside their community.

First, we verify how comments on Polish Twitter and web fora can be used for building the sentiment lexicon concerning political discussions in the election period. Second, we investigate how sentiment lexicons built in such a way can be used in order to create the relevant training data set, which is needed for utilizing the machine learning approaches. Third, we train the Naive Bayes and Maximum Entropy classifiers and perform 10-fold cross validation and we evaluate several methods using lexicon built using forum comments on two data sets. Finally, we evaluate methods augmented by word2vec to see how word embeddings can help and increase performance of dictionary based methods.

2 Related Work

The main set of sentiment analysis problems shares the following general task: analyze an opinionated piece of text and classify the opinion as falling under one of two opposing sentiment polarities or define its position on the continuum between these two polarities. A large portion of work in sentiment-related classification falls within this category. Much research on sentiment polarity classification has been conducted in the context of reviews (e.g., “thumbs up” or “thumbs down” for movie reviews). While in this context “positive” and “negative” opinions are often evaluative (e.g. “like” vs. “dislike”), there are problems where the interpretation of “positive” and “negative” is subtly different. One example is determining whether a political speech is in support of or in opposition to the issue under debate [12]. A related task is classifying predictive opinions on election fora into “likely to win” and “unlikely to win” [12]. Since all these problems are concerned with two opposing subjective classes, they are often amenable to similar techniques as machine learning tasks. In our work, the focus is on the three label-classification problem (positive, negative, neutral) in the context of the Internet comments analysis.

Internet discussions involve high numbers of people, in contrast to traditional media with their relatively passive audiences. In some Internet environments, such as social media hubs (e.g., Twitter, Facebook, blogs) and discussion forums, the users immediately express their views. With the rise of weblogs and the increasing tendency of online publications to turn into message-board-style reader feedback venues, informal political discourse is becoming an important feature of the intellectual landscape of the Internet, creating a challenging area for experimentation in techniques for sentiment analysis. Mutz and Martin [10] defined the hypothesis that media would surpass face-to-face communications across political divides. The Internet-based discussions provide not only access to facts and opinions “packaged” by professionals and presented in a concise form, typical for press or TV, but also exposure to raw, diverse views of “ordinary people,” and they complement the traditional media in this eye-opening role.

Wojcieszak [21] studied self-organized, assortative grouping of people sharing the same interests and political views into online communities that may separate from each other and become internally homogeneous. The descriptions of the emotional attitude of such communities become impossible to synthesize into one coherent personal worldview, especially when accompanied by the polarized traditional media and biased selection of information sources by the users. The communication between supporters of the conflicted camps would be difficult especially in the face-to-face mode due to the strong emotions that divide the society.

MacKuen et al. [7] introduce an interesting concept of two idealized types of participants in political debates: a deliberative citizen, who considers all arguments, including these opposite to his views, and a partisan combatant, passionate supporter of a single viewpoint. In real-life situations, people's behaviour falls somewhere between these two extremes. The authors argue that it is emotions that distinguish the deliberative from the combative stance.

Mullen and Malouf [9] describe preliminary statistical tests on a new dataset of political discussion group postings, which indicate that posts made in direct response to other posts in a thread have a strong tendency to represent a political viewpoint in opposition to the original post.

Mining the Sentiment from political Web posts is presented in the paper [3]. Sentiment classification of weblog posts, political weblog posts in particular, appears to be a more difficult problem than classification of traditional text because of the interplay of the images, hyperlinks, the style of writing and language used within weblogs. Using a dedicated dataset gathered from a two-years' worth of political weblogging, the authors investigated how correctly Naive Bayes and SVM classifiers predict the political category of a given post.

In the paper by [2], the recent Irish General Election was used as a case study for investigating the potential to model political sentiment through mining social media. The proposed approach combines sentiment analysis using supervised learning and volume-based measures. Evaluation was done against the conventional election polls and the final election result.

Stieglitz and Dang-Xuan [19] conducted research on the sentiment analysis of Twitter messages and their "retweetability". The paper examines whether the sentiment occurring in the politically relevant tweets has an effect on how often these tweets will be retweeted. In the data set of 64,431 political tweets, a positive relationship was found between the quantity of words indicating affective dimensions (including positive and negative emotions associated with certain political parties or politicians) in a tweet, and its retweet rate.

Paltoglou et al. [11] employed Maximum Entropy, Naive Bayes and Lexicon based methods in order to analyze the sentiment of data originating from BBC and Digg. The results show that the Lexicon based methods outperform machine learning methods. However, Naive Bayes scores higher results than Maximum Entropy Classifier.

3 Political Discussion Data Resources

3.1 Data Sources

Political comments were gathered from three major Polish news websites, onet.pl, gazeta.pl and wp.pl.

The comments from onet.pl and wp.pl were selected to cover 5 topics that were important for the public opinion during the three months after the first round of presidential elections in Poland. Three of the topics covered main candidates (Andrzej Duda, Bronisław Komorowski, Paweł Kukiz), last two covered the current prime minister (Ewa Kopacz), and the shadow prime minister and one of campaign leads (Beata Szydło). We gathered 1,533,035 comments from 2057 articles published between 21 May 2015 and 28 August 2015. We used semi-automatic crawling software written in Python and C# that took list of articles (manually gathered) and crawled subsequent pages for news and comments. The dataset containing these comments will hereinafter be called POL2015.

We also gathered two datasets that were used in the creation of sentiment lexicon and in Machine Learning algorithm training. The first dataset contains around 31,095 tweets with automatically tagged sentiment, ranging from −1 (negative) to 1 (positive), with no neutral sentiment. Automatic sentiment tagging was based on an algorithm, provided by Twitter, that interprets tweets with ":)" emoticons as positive and ":(" as negative. This dataset will be hereinafter called TW2015. The second dataset (POL2012) contains 6,500 political texts from 2011 and 2010, gathered by Sobkowicz and Sobkowicz [18], with manually tagged sentiment, ranging from −3 (very negative) to 1 (positive). Comments in POL2012 datasets came from three sources - gazeta.pl website (two batches, one form 2010 and one from 2011) and one from wyborcza.pl (from year 2011).

The datasets were annotated by a single researcher (due to time and financial constrains), which may have skewed the objectivity of sentiment.

Political texts and Twitter texts are vastly different, as shown in the example below. Typical Twitter text is shorter than 140 letters (a Twitter's constraint by design) and often contains tokens that are not words (like links or emoticons), as in examples:

- "RT @przepisy_dzieci: Kokosowa #kasza manna - pyszne #przepisy dla dzieci :) http://t.co/uC1M5GS5yY" ("RT @przepisy_dzieci: Coconut #farina - tasty #recipe for kids :) http://t.co/uC1M5GS5yY")
- "@jerry72p tu chodzi o grubsza sprawe.Niebawem powinna wyciec :)" ("@jerry72p it's about something bigger.It should leak soon :)")

Political comments in the other hand are often longer, have better grammar and rarely have user handles, links or included:

- "Pan Duda to chyba czytać nie umie, bo wszystko mówi z pamięci nie to co pan Komorowski duka,stęka, ale jakoś przeczyta co mu napiszą". ("Mr. Duda probably does not know how to read, because he speaks from memory only unlike Mr. Komorowski stamms,groans, but reads what they write for him.")

- "Właśnie wymieniłeś same zalety. Niestety poprzedni miał tylko dziadka - Osip Szczynukowicz. Niektórym to wystarczyło." ("You mentioned only advantages. Unfortunately previous had only grandfather - Osip Szczynukowicz. For some it was enough.")

It is also important to note, that in case of Onet.pl dataset the user can post under any nickname. This seems to encourage more heated discussions, with lots of insults – token based analysis of the dataset using only known, heavily emotive negative tokens shown that around 5% of all messages can be considered as directly insulting (insulting other users or other parties connected to the topic of the discussion).

3.2 Extended Corpus Description

Apart the corpus used in this research (defined above), we have also gathered more general data set (from two additional periods - September - December 2015 and January to March 2016), concerning political comments on the Onet.pl webpage, without a focus on presidential election, to check basic linguistic properties of discussions.

Extended corpus contains 4,829,076 texts, with average length of 179 characters. Distribution of tokens to a number of texts in the corpus is shown in Figs. 1 and 2 – non-unique tokens and unique tokens only respectively. Corpus itself contains over 160 million tokens, with 1,826,906 unique tokens (as we do no extract lemmas from the words, this number in inflated by different conjugations).

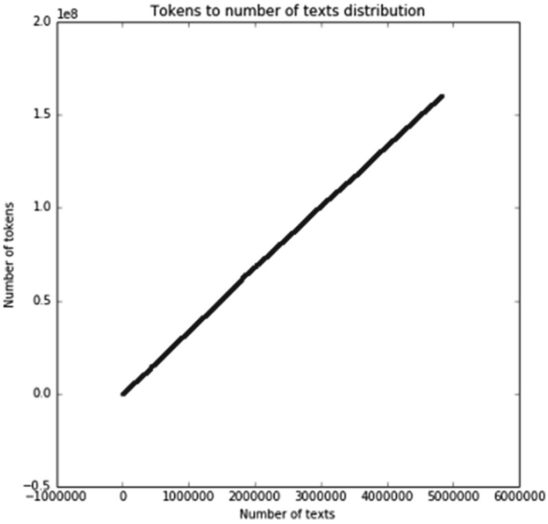


Fig. 1. Distribution of number of non-unique tokens in compare to number of texts in corpus.

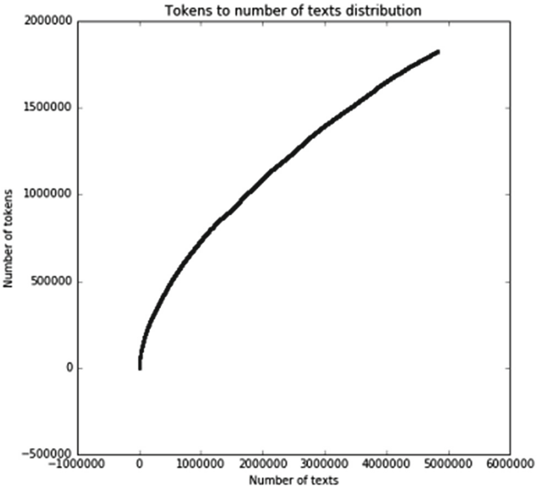


Fig. 2. Distribution of number of unique tokens in compare to number of texts in corpus

3.3 Manual Sentiment Tagging

Manual sentiment tagging was done by the authors, which means that all assessments are highly subjective and could vary for a different group of people, or even for the same person re-reading the analyzed text later.

The neutral emotion tag can additionally increase error rate, since texts with clearly positive and negative emotions can be deemed neutral as more and more diverse extreme emotions appear in texts. One reason for this is that, for humans, negative emotions have higher impact [13], which transfers to higher confidence in the accuracy of one's formed impression when it was formed more on the basis of negative traits than positive traits [1]. This personal impression may serve as an incentive to change the sentiment in order to overcome the negative bias.

In data analyzed by Sobkowicz and Sobkowicz (2012), neutral texts were 56% of all texts, however, authors used different neutrality and emotiveness measure.

3.4 Keywords as Emotion Carriers

Political discussion on web fora are centered on keywords, which are used in a similar way to hashtags on social media - they provide focal points in the discussions. However, in contrast to the social media, keywords are not marked (their recognizability is based on human perception). Extracting keywords manually can is time consuming work, and may not be possible given size of datasets. The presented algorithm allows to detect new keywords based on a starting sample in a way that closely reproduces the intended meaning.

Using Word2Vec [8] allows us to easily extract keywords using only small initial set. This is possible as our keywords, which are nicknames used for political persons, are used in different context than their true names. Using this method we can easily create end expand specialized dictionaries and keyword sets.

Tables below shows tags extracted using Word2Vec algorithm from collected data for several basic keywords. Table 1 contains comparison of keywords extracted from comments collected in two timeframes. This analysis was performed on data coming from Onet.pl website, from June - August 2015 and September - December 2015.

Table 1. Comparison of keywords retrieved using algorithm from two time frames, June - August 2015 and September - December 2015

Komoruski		Ryży		Beatka	
Jun - Aug	Sept - Dec	Jun - Aug	Sept - Dec	Jun - Aug	Sept - Dec
Szogun	Bredzisław	Rudy	Donek	Szydełko	Becia
Gajowy	Bul	Tusek	Rudy	Straszydło	Szydłowa
Bul	Bronek	Donek	Tusk	Szydło	Szydełko
Kompromitowski	Szogun	Nieróbb	Tusek	Beata	Straszydło
Bronek	Gajowy	Szczur	Szczur	Szydłowa	Sołtysowa

These results encouraged us to see if enhancing dictionary based methods with Word2Vec embedding is a viable option for sparsity problem.

4 Non-semantically Enhanced Methods

4.1 Sentiment Lexicon Creation

Both sentiment lexicons were created automatically from the sentiment-tagged data sets using the token sentiment value generation procedure described by Kiritchenko in [6]. The method generates sentiment value s for each word (token) t based on the pointwise mutual information (PMI):

$$s(t) = PMI(t, positive) - PMI(t, negative) \tag{1}$$

$$PMI(t, positive) = log_2(\frac{freq(t, E_1) * count(W)}{freq(t, W) * count(E_1)}) \tag{2}$$

where $freq(t, E_1)$ is the number of times the token t occurs in the collection E_1 (positive tokens) and $count(W)$ is the number of different tokens t in the collection W (all tokens). $freq(t, W), freq(t, E_{-1})$ are described in similar way, as is $PMI(t, negative)$.

It allows for building lexicons without human effort and ensures that the dictionaries were created in the same conditions.

4.2 Methods

Comments were analyzed using five methods: three dictionary based methods, using two different sentiment lexicons, and two machine learning based ones (Naive Bayes and Maximum Entropy Classifiers) using two different training sets.

The dictionary based methods process an input text as a bag-of-words. Simple Dictionary Based (SDB) method takes the sentiment value of each word and returns the sum of sentiment values (s). $\log^2$ Dictionary Based (LDB) algorithm (inspired by [18]) works in a similar way, but the final sentiment S value is derived from the sentiment sum s using the following equation: $S = 0.8 * s * \log^2(\frac{h}{w})$, where h is the number of words that have a sentiment value, and w is the number of all words in an input text. NL Dictionary Based (NLDB) method employs a more complicated procedure described in [17]. In short, NLDB takes the text, lemmatizes each word and checks its part of speech. Using this information, it applies different weights to the sentiment values of each word, depending on its location and predecessors. Finally, it saves the sum of all sentiment values obtained that way.

The machine learning methods evaluated are Maximum Entropy Classifier (MEBoW) and Naive Bayes Classifier(NB). Both methods were trained using the same training data set. Maximum Entropy is logistic regression based classifier, used when more than two outcome classes are needed, as described in [4]. Naive Bayes classifier is a simple probabilistic classifier based on the Bayes' theorem, that was first introduced in 1950s (after [16]).

Output from all methods was normalized to integer values between -1 (negative) and 1 (positive), where 0 means neutral.

4.3 Evaluation Procedure

Machine learning algorithms were evaluated using 10-fold cross-validation. All folds have comments containing all three emotion values (-1, 0, 1). Each method was tested on two data sets[1]: 950 manually sentiment annotated comments from POL2015 (hereinafter POL2015T) and 650 comments from POL2012 (hereinafter POL2012T). The data sets used to train machine learning algorithms and create sentiment lexicons consist of 5850 comments from POL2012 (TRAIN-POL) and all text from the Twitter set TW2015 (TRAIN-TWIT) (Table 2).

The evaluation procedure work as follows:

1. Preparation
   (a) (In case of machine learning based algorithms) Training the classifier
   (b) (In case of dictionary based algorithms) Loading the sentiment lexicon
2. Loading the test data set
3. Sentiment Classification
4. Comparing algorithm's sentiment values with the gold standard ones (manually tagged)

[1] Datasets are available on http://opi-lil.github.io/datasets/website.

Table 2. Comment distribution in POL2012T and POL2015T. While datasets are unbalanced (heavily biased towards neutral and negative comments), argument can be made that artificially balancing the data would not represent the real world situation and method should perform well using unbalanced data as an learning dataset

Text emotion				
	−1	0	+1	Total
POL2012T	327	310	13	650
POL2015T	644	146	160	950
TRAIN-POL	2387	3304	159	5850
TRAIN-TWIT	14053	0	17042	31095

The reported sentiment values were measured using the raw accuracy (number of texts with correctly detected sentiment). The evaluation was performed for two cases: binary sentiment detection (texts with manually tagged neutral sentiment were ignored) and full sentiment detection (including neutral sentiment detection).

4.4 Results

We performed two kinds of experiments: one narrowed to the machine learning algorithms, the second one for all the methods.

Table 3 contains results for 10-fold cross-validation of Maximal Entropy and Naive Bayes classifiers, using TRAIN-POL as a training data set. Cross-validation was done using full, 3 level sentiment categorization (−1: negative, 0: neutral, 1: positive).

Table 3. Results for Accuracy on TRAIN-POL data set.

10-fold cross-validation results	
	TRAIN-POL
MEBoW	0.79
NB	0.45

MEBoW provides better 10-fold cross-validation results than Naive Bayes. This shows that Maximum Entropy performs better than simple Naive Bayes for analysing 3 category sentiment.

Tables 4 and 5 contain the results for dictionary based and machine learning based algorithms using training data sets TRAIN-POL and TRAIN-TWIT respectively. All methods were tested on POL2012T and POL2015T data sets. Columns labeled FULL contain accuracy results for 3-category sentiment detection and columns labeled as BIN contain accuracy results for binary sentiment classification (positive vs negative). Results show that for all algorithms except MEBoW the full sentiment detection accuracy is lower than the binary sentiment detection. MEBoW reports a slightly better 3-category sentiment accuracy when the test data set shares the origin with training data set, but performs noticeably worse on test data sets of different origin. MEBoW scores for TRAIN-TWIT data set were so low in comparison to NB results that this method was omitted in further discussion.

Table 6 and Fig. 3 contain accuracy comparison for algorithms trained using TRAIN-POL and TRAIN-TWIT data sets and tested on POL2015 data set. Column labels are the same as in previous tables. Results show that training on TRAIN-POL data set, based on tagged political texts from web portals, provides better results than using a TRAIN-TWIT, which was based on general Twitter messages.

Table 4. Results for Accuracy on POL2012T and POL2015T test data sets, algorithms were trained with TRAIN-POL data set. It is very notable to see that despite performing very well on similiar data (both TRAIN-POL and POL2012T came from same time period), Maximum Entropy failed on POL2015T dataset.

Accuracy results for test data				
	POL2012T		POL2015T	
	FULL	BIN	FULL	BIN
MEBoW	0.73	0.78	0.39	0.35
NB	0.52	0.95	0.65	0.76
SDB	0.52	0.97	0.68	0.78
NLDB	0.53	0.94	0.63	0.74
LDB	0.53	0.94	0.69	0.73

Table 5. Results for Accuracy on POL2012T and POL2015T data sets, algorithms were trained with TRAIN-TWIT data set. All tested algorithm have very similar binary and full, 3 category accuracy. Maximum Entropy method was omitted due to having very low scores.

Accuracy results for test data				
	POL2012T		POL2015T	
	FULL	BIN	FULL	BIN
NB	0.27	0.44	0.25	0.3
SDB	0.26	0.46	0.31	0.37
NLDB	0.23	0.44	0.28	0.34
LDB	0.25	0.45	0.3	0.35

Table 6. Results for Accuracy on POL2015T using TRAIN-POL and TRAIN-TWIT training data sets for algorithm training and sentiment lexicon generation. When trained using politically-aligned data algorithm are performing a lot better than when using generic data coming from sources such as Twitter.

Accuracy comparison for POL2015				
	TRAIN-POL		TRAIN-TWIT	
	FULL	BIN	FULL	BIN
NB	0.65	0.76	0.25	0.3
SDB	0.68	0.78	0.31	0.37
LDB	0.69	0.73	0.3	0.35
NLDB	0.63	0.74	0.28	0.34

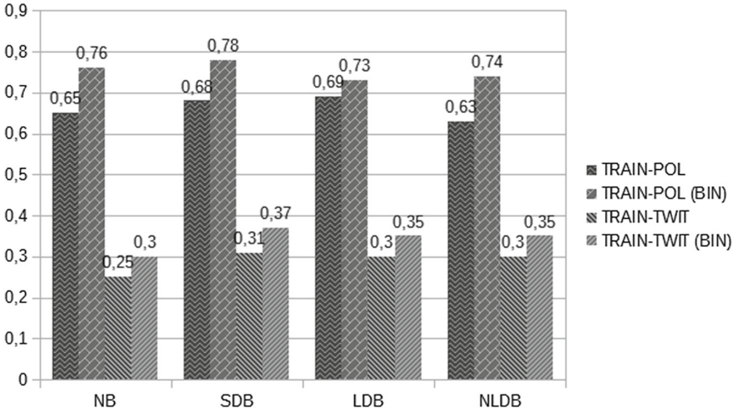


Fig. 3. Results for Accuracy on POL2015 using TRAIN-POL and TRAIN-TWIT training data sets for dictionary based and machine learning methods.

5 Word2Vec Oriented Dictionary Enrichment

One of the strongest trends in Natural Language Processing (NLP) at the moment is the use of word embeddings [5], which are vectors whose relative co-occurrence similarities correlate with semantic similarity. Such vectors are used both as an end in itself (for computing similarities between terms), and as a representational basis for downstream NLP tasks like text classification, document clustering, part of speech tagging, named entity recognition, sentiment analysis, and so on. Also, neural network based distributional semantics received a substantially growing attention. The main reason for this is a very promising approach of employing neural network language models (NNLMs) trained on large corpora to learn distributional vectors for words. Recently, Mikolov et al. [8] introduced the Skip-gram and Continuous Bag-of-Words models, being efficient methods for learning high-quality vector representations of words from large amounts of unstructured text data. The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns.

5.1 Sentiment Lexicon Creation

Similarly as with non semantically enhanced methods, we used manually sentiment annotated texts to automatically calculate sentiment of words (and bigrams) using Pointwise Mutual Information (using the same algorithm as described above). These individually generated dictionaries where next used to assess sentiment of gathered data in the dictionary based methods. This time however, in order to resolve the problem of word's sparsity in the manually tagged comments we have introduced the Word2Vec models to detect semantically similar words to those highly polarized appearing in the manually tagged comments.

5.2 Evaluation Procedure

In order to test our assumption that using word embeddings can indeed help with sentiment analysis using dictionary based methods, we split our datasets - POL2012 and POL2015 into four parts. Three parts came from POL2012 dataset and corresponded to original sources - Gazeta (2010), Gazeta (2011) and Wyborcza (2011). The fourth dataset came from POL2015 and consisted of all manually tagged comments, later called POL2015T (this data set was the same as in previous step).

Test procedure consisted of building word embedding's model on all gathered data, building dictionary using one of the sets as a training set, and then testing each text from all other sets (as test sets) using raw accuracy as a measure. Sentiment was calculated using same methods as when using non-augmented dictionary - with an exception that if the word was not found in the dictionary, algorithm searched if any of the top 10 most similar words - according to the word embedding - was in a dictionary. If such a word was found, we used sentiment value associated with that word.

5.3 Results

Performed analysis has shown that in the case of binary classification we managed to get better accuracy when evaluating POL2015T set - we managed to get accuracy over 0.8 using dictionary based on any of the other sets. Full results are presented in Table 7 and on Fig. 4.

Lower accuracy in case of dictionary created using data from 2015 can be partly due to lower number of manually annotated texts in 2015 dataset in comparison to 2012 ones.

Table 7. Results for cross evaluation of dictionary based methods using different dictionaries to annotate different sets. Rows represent dataset that was being tested, and columns represent dataset from which dictionary was created. It is worth to note that despite 3 years difference, augmented dictionaries created from data from 2012 can be used to assess sentiment of data 2015 with greater accuracy than when using non-augmented ones.

	Gazeta (2010)	Gazeta (2011)	Wyborcza (2011)	POL2015T
Gazeta (2010)	-	0.92	0.89	0.75
Gazeta (2011)	0.93	-	0.90	0.76
Wyborcza (2011)	0.93	0.94	-	0.75
POL2015T	**0.86**	**0.84**	**0.85**	-

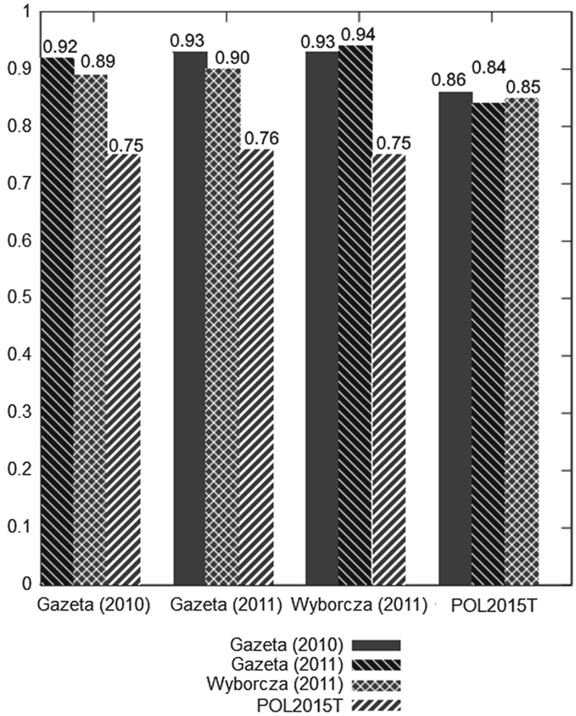


Fig. 4. Raw accuracy results for cross evaluation of dictionary based methods using different dictionaries to annotate different sets. Groups represent dataset that was being tested, and bars represent dataset from which dictionary was created.

6 Conclusions

The paper compares three dictionary based sentiment analysis methods, built using different sentiment lexicons, with two machine learning based sentiment classifiers, using Internet comments concerning the last Polish election candidates. We use Twitter and Internet forum's comments both as training/test data sets and for building the sentiment lexicon concerning political discussions in the election period.

Results show that Naive Bayes algorithm does not work well with 3 category input. Although it has a better binary accuracy than Maximum Entropy, its performance on texts with neutral sentiment falls short. Detecting neutral sentiment is a difficult task (although there are works that try to combat this problems, for example by detecting if text is emotive at all, see [14]); all evaluated methods except Maximum Entropy report worse accuracy on both test datasets.

Comparison of dictionary based methods with Naive Bayes shows that their accuracy is nearly the same on all test data sets. Results above 0.75 for binary sentiment classification are high enough to be usable in real world scenarios, even taking into consideration the bias resulting from the manual sentiment tagging. Notably, SBD, the simplest of dictionary based methods, provides the best overall results. This may be due to the fact that dictionary based methods are less sensitive to missing dataIn addition, a The Maximum Entropy algorithm, while performing well on a data set similar to the training set, works much worse for data coming from other sources.

Experiments using Twitter data set for machine learning training and sentiment lexicons creation show that general, non-filtered Twitter data cannot be easily used to build relevant models/lexicons for analysing political texts from the web portals. This is because political texts very often contain domain-specific slang not represented in other social media.

Results using semantically enhanced dictionaries show very promising results - with accuracy being higher than when not using word embeddings to aid dictionaries - but it must be noted that manually tagged datasets were limited in size.

Interestingly, despite political changes during last 3 years, the training data from 2012 seems to work well on texts from 2015. This indicates that Polish political slang is fairly stable.

Overall, the results show that while high binary accuracy can be easily achieved, detecting all three sentiment categories with high accuracy (>0.75) is hardly possible using only text processing. One way to improve the algorithms is to use some additional features, e.g. ones from social network analysis.

References

1. Baumeister, R.F., Bratslavsky, E., Finkenauer, C., Vohs, K.D.: Bad is stronger than good. Rev. Gen. Psychol. **5**(4), 323 (2001)
2. Bermingham, A., Smeaton, A.F.: On using twitter to monitor political sentiment and predict election results (2011)
3. Durant, K.T., Smith, M.D.: Mining sentiment classification from political web logs. In: Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA (2006)

4. Greene, W.H.: The econometric approach to efficiency analysis. In: The Measurement of Productive Efficiency and Productivity Growth, pp. 92–250 (2008)
5. Huang, E.H., Socher, R., Manning, C.D., Ng, A.Y.: Improving word representations via global context and multiple word prototypes. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, vol. 1, pp. 873–882. Association for Computational Linguistics (2012)
6. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. J. Artif. Intell. Res. **50**(1), 723–762 (2014)
7. MacKuen, M., Wolak, J., Keele, L., Marcus, G.E.: Civic engagements: resolute partisanship or reflective deliberation. Am. J. Polit. Sci. **54**(2), 440–458 (2010)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
9. Mullen, T., Malouf, R.: A preliminary investigation into sentiment analysis of informal political discourse. In: AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, pp. 159–162 (2006)
10. Mutz, D.C.: Facilitating communication across lines of political difference: the role of mass media. In: American Political Science Association, vol. 95, pp. 97–114. Cambridge Univ Press (2001)
11. Paltoglou, G., Gobron, S., Skowron, M., Thelwall, M., Thalmann, D.: Sentiment analysis of informal textual communication in cyberspace. In: Proceedings of the Engage 2010, Springer LNCS State-of-the-Art Survey, pp. 13–25 (2010)
12. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval **2**(1–2), 1–135 (2008)
13. Peeters, G., Czapinski, J.: Positive-negative asymmetry in evaluations: the distinction between affective and informational negativity effects. Eur. Rev. Soc. Psychol. **1**(1), 33–60 (1990)
14. Ptaszynski, M., Masui, F., Rzepka, R., Araki, K.: Emotive or non-emotive: that is the question. In: ACL 2014, p. 59 (2014)
15. Rainie, L., Horrigan, J.: Election 2006 online (2007)
16. Russell, S., Norvig, P.: Artificial intelligence: a modern approach (1995)
17. Sobkowicz, A.: Automatic sentiment analysis in polish language. In: Ryżko, D., Gawrysiak, P., Kryszkiewicz, M., Rybiński, H. (eds.) Machine Intelligence and Big Data in Industry. SBD, vol. 19, pp. 3–10. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30315-4_1
18. Sobkowicz, P., Sobkowicz, A.: Two-year study of emotion and communication patterns in a highly polarized political discussion forum. Soc. Sci. Comput. Rev. **30**(4), 448–469 (2012)
19. Stieglitz, S., Dang-Xuan, L.: Political communication and influence through microblogging-an empirical analysis of sentiment in twitter messages and retweet behavior. In: 2012 45th Hawaii International Conference on System Science (HICSS), pp. 3500–3509. IEEE (2012)
20. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. J. Am. Soc. Inf. Sci. Tech. **61**(12), 2544–2558 (2010)
21. Wojcieszak, M.: "Don't talk to me": effects of ideologically homogeneous online groups and politically dissimilar offline ties on extremism. New Media Soc. **12**, 637–655 (2010)

Saturation Tests in Application to Validation of Opinion Corpora: A Tool for Corpora Processing

Zygmunt Vetulani[1](✉), Marta Witkowska[1], Suleyman Menken[2], and Umut Canbolat[2]

[1] Adam Mickiewicz University in Poznań, Poznań, Poland
vetulani@amu.edu.pl, martusiazielinska@gmail.com
[2] University of Kocaeli, İzmit, Kocaeli, Turkey
suleyman.menken@gmail.com, u.canbolat@yahoo.com

Abstract. Opinion processing has recently gained much interest among computational linguists, public relation experts, marketing companies, and politicians. Studies of the natural language expression of opinions, desires, emotions, and related phenomena require appropriate tools and methodologies. We propose tools for collection of empirical data in the form of a corpus, limiting our research field to customers' written opinions about widely used on-line booking services in the area of hotel reservations (via Booking.com). In this paper, we present the corpus acquisition procedure and our data acquisition tool, as well as discuss our decisions about the selection of the source data. We also present some limitations of our proposal and propose a validation methodology for the resulting corpora.

Keywords: Text corpora · Language resources · Opinion processing
Corpora validation · Saturation tests

1 Introduction

Why is opinion processing so important? One possible answer to this question results from the analysis of the social role of opinions, and in particular of their contribution to the evolutionary success of the humankind. This success is based on the aptitude of taking right decisions, especially when access to facts is limited. According to a popular definition, unlike a verifiable fact, "an opinion is a judgment, viewpoint, or statement that is not conclusive" and is usually subjective (Wikipedia "Opinion" 2018-02-07). Three points worth noticing are as follows:

- decisions are being taken based on premises that may be opinions, in particular when facts are unavailable,
- the quality of decisions is related to the quality of opinions used as decision premises, and
- opinions are often emotionally biased, and this bias affects their quality.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 378–390, 2018.
https://doi.org/10.1007/978-3-319-93782-3_27

The sense we give in this paper to the term *opinion* is close to the one defined in the popular Collins English Dictionary [1] as "judgment or belief not founded on certainty or proof". We propose to make the meaning more precise (after Charaudeau and Maingueneau [2]): *an opinion expresses a subject's evaluative opinion in favor or against facts*.

2 Project Objectives

The medium-term aim of the project is to create a repository of comparable (size, domain, acquisition mode, and nature of texts) opinion text corpora for different languages as a tool for opinion studies. More precisely, the project aims to serve fundamental research—both descriptive and comparative—on opinion expression in natural language for various languages.[1] To give insight into the distribution of these phenomena, the domain (set of domains) should be defined precisely, and the corpora should be representative for the phenomena of interest and large enough to allow qualitative observations. The resources should be free of legal flaws related to the acquisition procedures and should not have usage restrictions for linguistic research purposes.[2] Instead of proposing a closed set of texts for an *a priori* defined list of languages, we decided to design and implement a software tool to compile a corpus of customers' opinions in the area of hotel services. As a source of data, we chose the popular service Booking.com.

The objectives presented above are crucial for further, application-oriented works. In particular, the analysis of emotional layer of opinions used as decision premises may help evaluate the appropriateness of decisions to be taken (risk analysis).

Some other application fields of opinion processing are

- political decisions, election campaigns, and business PR campaigns
- questionnaire-based customer profiling, and
- forensic profiling

3 Data

3.1 Why Did We Choose Hotels?

We were motivated by several factors when choosing hotel opinions as a corpus acquisition domain. The most important was the common (at the world scale) custom of booking portals to acquire client opinions (both positive and critical) and display them publicly in the Internet. We thus expect a large volume of accessible data (over

[1] Only a few opinion corpora exist. One of the best known is the MPQA Opinion Corpus of English texts (University of Pittsburg, PA, USA), http://mpqa.cs.pitt.edu/corpora/mpqa_corpus/ [3]. See also the five-billion-word Corpus of Japanese blogs annotated for affective features [4].

[2] Booking.com guests' comments are not copyright protected elements of the content but just publicly presented opinion recordings.

50 M opinions accessible through Booking.com). Also, opinions are accessible for almost all languages used by customers.

Although the basic function and structure of hotel services is practically the same in almost all hotels, realization modalities and customer expectations may be strongly culturally biased, so one may *a priori* expect a large variety of the content. The choice of analyzing hotel opinions has one great advantage: the field of hotel opinion expression is worldview neutral. Another advantage is that we do not need to approach many taboo areas,[3] which could disturb the acquisition of well-balanced comparative data for various languages.

3.2 Why Did We Choose Booking.Com?

Nowadays, providing customers' opinions is a common practice in the internet commerce, in particular in the sector of services. Such opinions are relatively easy to obtain for research purposes. What is more questionable is data authenticity and representativeness. Representativeness means equal access to all opinions regardless of their content. As long as the opinions are presented by a concerned subject, the risk of manipulation (lack of honesty) should not be ignored. Such a risk is minimal, if the opinions are being gathered by a big, possibly global, service provider (broker). Examples of such providers in the hotel sector are as follows:

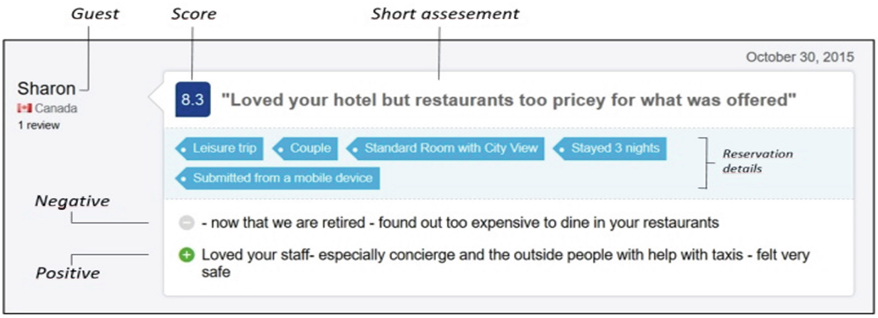
- www.booking.com
- www.hotels.com
- www.hrs.com
- www.tripadvisor.com
- www.trivago.com
- www.worldhotel.com

We chose the popular hotel reservation operator Booking.com[4]. It does not mean, however, that it is the only possible source of high-quality multilingual opinion corpora. To check the completeness (representativeness) of the corpus, it is useful to apply the chosen acquisition method to other thematic domains (e.g. catering or other kinds of services). A good candidate for this research could be, for example, the Tripadvisor, known for offering a wider spectrum of services addressed to tourists and travelers (including flight reservations and restaurants). We chose Booking.com because

- it provides a huge amount of opinion texts (over 59 million of verified opinions)
- it is a global operator covering practically all countries
- it allows opinions in all languages
- opinion texts are easily accessible
- opinions are pre-classified into positive and negative by the opinion providers (i.e., hotel guests) (see Fig. 1)
- basic information about hotels is available (address, category, price, and facilities)

[3] Such as political, religious, or custom-related opinions.

[4] Information about Booking.com presented in this paper was collected in November, 2015.



October 30, 2015

Sharon
Canada
1 review

8.3 "Loved your hotel but restaurants too pricey for what was offered"

Leisure trip | Couple | Standard Room with City View | Stayed 3 nights | Submitted from a mobile device

- now that we are retired - found out too expensive to dine in your restaurants

Loved your staff- especially concierge and the outside people with help with taxis - felt very safe

Fig. 1. An example of the Booking.com opinion record completed by a guest (2015).

– comparable data are available for practically all languages spoken by the users (making it possible to create comparable corpora for a large variety of language pairs)
– opinions are anonymous

Pre-classification of opinion texts is highly useful when introducing pragmatic information to formalized lexicons as WordNet-like lexical databases (e.g., SentiWordNet [5]; see also the discussion of wordnet granulation issues in PolNet in [6]). Notice that Pak and Paroubek [7] used emoticons (following similar procedures as in [8]) as pre-classifiers used in their Twitter-extracted corpus for sentiment analysis and opinion mining.

3.3 General Characteristics of the Available Data

We will study the opinion texts from Booking.com. According to its web page, Booking.com provides over 59,020,000 verified opinions. A verified opinion is one that passed an authenticity test—to prevent abusive opinions, Booking.com assumes that and checks whether the opinions it publishes were written by real hotel guests. For our purposes, a negative aspect of this verification is the censorship aimed to eliminate naughty words, discriminatory remarks, and offensive language, because the validity of the resulting corpora is limited to the neutral and high language registers.[5] Such censorship, however, is standard among all parties publicly presenting customer opinions, hence it is hard to avoid. The users must thus take this limitation into account when studying language expression of the opinions included in the corpora.

The way Booking.com presents opinions has interesting properties. It presupposes opinion pre-classification at the acquisition time, as the guests are requested to provide separately positive and negative observations. This feature reduces the necessity of intention analysis, otherwise necessary to identify and classify the opinions provided in

[5] To get a more precise idea on the nature of these limitations, the reader can consult Booking.com Guest Review Guidelines. To find it, open Booking.com and select any hotel. Find and click Our quests' experiences on the bar at the top of the page and then click read more (last checked on July 30, 2017).

an unstructured text form. Another positive feature is the possibility of partial contextual reconstruction of the situation inspiring a particular opinion. Although opinion authors are practically anonymous (typically they are presented by the first name or nickname and the declared provenance country), some relevant information to help interpret their opinions are supplied and easy to get, such as the hotel location, Booking.com ranking (a score calculated on the basis of opinions and stars given by the customers), and information about the offer and extras.

4 Opinion Corpus Acquisition Software (OCAS)

The first version of the specialized software (OCAS) for building corpora of opinion texts was created in 2015 at Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań, and has been successively up-dated and maintained ever since.[6] The system OCAS permits the user to collect opinion texts from Booking.com in a few steps presented in Fig. 2. First, the user chooses the country and city of interest and then selects hotels. He or she decides on the file format in which the opinion data are to be presented (XML, MS Word .doc, or raw .txt format).

Figure 2 presents the OCAS screenshots (selection) illustrating data acquisition: (a) the start screen, (b) the selection of a city (Istanbul), (c) all hotels in Istanbul, and (d) the three selected hotels in Istanbul. Using the ‘Get Content’ button in panel (b), we can download the opinions in any of the accepted formats. Figure 3 presents example data in the XML format.

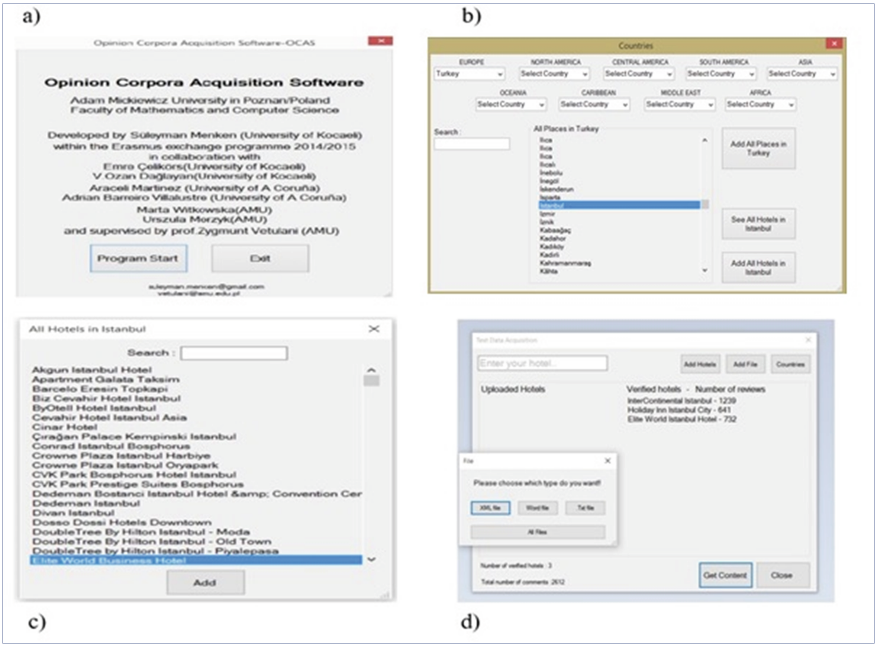
5 Opinion Corpora Acquisition Software for Subcorpora Creation (OCASSC)

The initial version of the system OCAS, presented at LTC 2015, has been up-dated several times during 2016 and 2017. Also, in 2017 the system was complemented with an independent (stand-alone) module OCASSC (Opinion Corpora Acquisition Software for Subcorpus Creation) for the extraction of subcorpora with desired properties, like size. Initially, it was conceived as a tool to process corpora in one of the OCAS[7] formats (text or XML). In particular, it was used to randomly generate sub-corpora of desired sizes. To make it possible to test corpus representativeness[8] for given phenomena, the OCASSC system enables one to make incremental saturation tests (see Sect. 7).

[6] OCAS was designed and implemented by a team composed of visiting Erasmus students of computer science (Süleyman Menken, Emre Çelikörs, and Veysi Ozan Dağlayan from Turkey and Arcaeli Martinez and Adrian Barreiro Vilalustre from Spain) and Polish students of linguistics (Marta Witkowska and Urszula Morzyk), under the supervision of Zygmunt Vetulani (AMU).

[7] In fact, OCASSC may be easily generalized to a system allowing generation of subcorpora of desired size for various XML formats.

[8] We say that the corpus is representative for a given language phenomenon, or a class of phenomena, if it contains examples for all relevant aspects of this phenomenon.



Opinion Corpora Acquisition Software-OCAS

Opinion Corpora Acquisition Software

Adam Mickiewicz University in Poznan/Poland
Faculty of Mathematics and Computer Science

Developed by Süleyman Menken (University of Kocaeli)
within the Erasmus exchange programme 2014/2015
in collaboration with
Emre Çelikors(University of Kocaeli)
V.Ozan Dağlayan(University of Kocaeli)
Araceli Martinez (University of A Coruña)
Adnan Barreiro Villalustre (University of A Coruña)
Marta Witkowska(AMU)
Urszula Morzyk(AMU)
and supervised by prof.Zygmunt Vetulani (AMU)

suleyman.mencen@gmail.com
vetulani@amu.edu.pl

Countries

EUROPE: Turkey
NORTH AMERICA: Select Country
CENTRAL AMERICA: Select Country
SOUTH AMERICA: Select Country
ASIA: Select Country
OCEANIA: Select Country
CARIBBEAN: Select Country
MIDDLE EAST: Select Country
AFRICA: Select Country

Search :

All Places in Turkey
Ilıca
Ilıca
Ilıca
Ilıcalı
İnebolu
İnegöl
İskenderun
Isparta
İstanbul
İzmir
İznik
Kabaağaç
Kadahor
Kadıköy
Kadirli
Kahramanmaraş
Kâhta

All Hotels in Istanbul

Search :

Akgun Istanbul Hotel
Apartment Galata Taksim
Barcelo Eresin Topkapi
Biz Cevahir Hotel Istanbul
ByOtell Hotel Istanbul
Cevahir Hotel Istanbul Asia
Cinar Hotel
Çırağan Palace Kempinski Istanbul
Conrad Istanbul Bosphorus
Crowne Plaza Istanbul Harbiye
Crowne Plaza Istanbul Oryapark
CVK Park Bosphorus Hotel Istanbul
CVK Park Prestige Suites Bosphorus
Dedeman Bostanci Istanbul Hotel &amp; Convention Cer
Dedeman Istanbul
Divan Istanbul
Dosso Dossi Hotels Downtown
DoubleTree By Hilton Istanbul - Moda
DoubleTree By Hilton Istanbul - Old Town
DoubleTree by Hilton Istanbul - Piyalepasa
Elite World Business Hotel

Text Data Acquisition

Enter your hotel...

Uploaded Hotels

Verified hotels - Number of reviews
InterContinental Istanbul - 1239
Holiday Inn Istanbul City - 641
Elite World Istanbul Hotel - 732

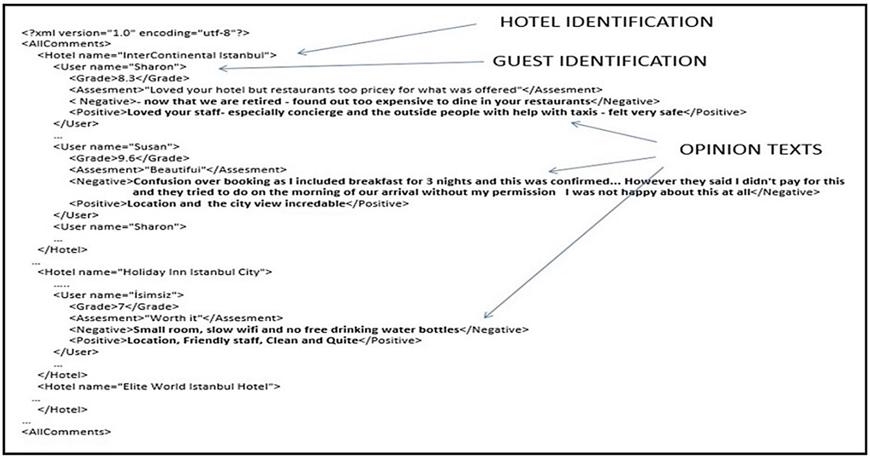
File

Please choose which type do you want!

Number of verified hotels : 3

Total number of comments: 2612

Fig. 2. The OCAS screenshots.


HOTEL IDENTIFICATION
GUEST IDENTIFICATION
OPINION TEXTS

```
<?xml version="1.0" encoding="utf-8"?>
<AllComments>
  <Hotel name="InterContinental Istanbul">
    <User name="Sharon">
      <Grade>8.3</Grade>
      <Assesment>"Loved your hotel but restaurants too pricey for what was offered"</Assesment>
      < Negative>- now that we are retired - found out too expensive to dine in your restaurants</Negative>
      <Positive>Loved your staff- especially concierge and the outside people with help with taxis - felt very safe</Positive>
    </User>
    ...
    <User name="Susan">
      <Grade>9.6</Grade>
      <Assesment>"Beautifui"</Assesment>
      <Negative>Confusion over booking as I included breakfast for 3 nights and this was confirmed... However they said I didn't pay for this
                and they tried to do on the morning of our arrival without my permission   I was not happy about this at all</Negative>
      <Positive>Location and  the city view incredable</Positive>
    </User>
    <User name="Sharon">
    ...
  </Hotel>
 ...
  <Hotel name="Holiday Inn Istanbul City">
    .....
    <User name="İsimsiz">
      <Grade>7</Grade>
      <Assesment>"Worth it"</Assesment>
      <Negative>Small room, slow wifi and no free drinking water bottles</Negative>
      <Positive>Location, Friendly staff, Clean and Quite</Positive>
    </User>
    ...
  </Hotel>
  <Hotel name="Elite World Istanbul Hotel">
 ...
  </Hotel>
...
<AllComments>
```

Fig. 3. A fragment of the XML formatted corpus presenting three different opinions about hotels in Istanbul.

OCASSC (version 1) requires two input data: (i) a corpus generated with OCAS in the XML format and (ii) a predefined list of elements that are formal indicators of the relevant phenomena (e.g., adjectives in the application case presented in Sect. 6), which may (but do not have to) appear in the corpus. Further in this paper, this list is called the *reference list*. The program retrieves the input data (here the corpus generated by OCAS) to create one or more subcorpora of the user-specified length. These subcorpora will permit one to perform a saturation test for the input reference list.

6 Corpus Evaluation Method: Saturation Tests

Despite the consensus on the fundamental importance of linguistic data corpora (texts, recordings) for investigating natural languages using the observation-based methodology of the natural sciences, there are no commonly accepted methods for the evaluation of language corpora quality. Initially, corpus size was considered a sufficient quality measure, but language engineers quickly changed their mind. It was the need of producing linguistic models for particular applications what brought their attention to specific linguistic phenomena. Consistently, corpora for language modelling are supposed to be *representative* for the concerned phenomena [9]. As corpora gathering is expensive (time, effort) and difficult (legal issues), the quality evaluation of existing corpora is important.

There are various methods to estimate the representativeness of a sample of data for a given phenomenon. What they have in common is the evaluation of a chance to find a new manifestation of the phenomenon outside the sample. In a representative sample, all relevant examples should occur at least once. If a sample is (almost) representative, then (almost) all new manifestations will be identical to some manifestations already present in the sample. In particular, the list of single manifestations of the phenomenon (list of hapax legomena) will decrease or remain the same after each new observation. This decrease results from the saturation of the considered phenomenon.

We will explore the concept of *lexical saturation* of a corpus [10, 11]. This concept appeared useful in the research on the evaluation of the size of virtual vocabulary of sublanguages (e.g., in the context of machine translation) and was used to study lexical saturation of corpora (ibid.). Informally, we say that a corpus is lexically saturated when “new lexemes appear only sporadically as a result of the extension of the corpus in a natural way” (ibid.). To study lexical saturation, we consider a corpus to be a linearly ordered set of elements (words, symbols, etc.). For its initial segments of length N, we observe the number of different words, V.[9] This function is increasing over time, and the growth of V informs us about the degree of saturation of the corpus. The function may be represented graphically by a *saturation graph*. Observations of various corpora confirm that V grows systematically with N—but more slowly. This is because the observed vocabulary becomes more and more saturate, that is, it is more and more difficult to introduce new words into discourse [11].

[9] To measure the length of a segment, we may use various units, such as characters, words, or sentences. In this paper we will use text words or opinions as the measurement units.

For a sound[10] data gathering procedure, it is crucial to have a good *stopping* criterion, that is, a criterion to stop data collection. A good stopping criterion will prevent us from collecting data beyond necessity.

Let us consider a corpus as a linearly ordered partition into segments of equal size. The observation of the number of new words, ΔV, in the last segment informs us about the degree of lexical saturation of the corpus. ΔN stands for a measure of the last segment size. Typically, ΔN is a number of words in the last segment, but other kinds of text units may appear useful, like segments, text lines, messages, or other. Below, ΔN stands by default for the number of booking.com hotel opinions.[11]

A sufficiently small value[12] of the ΔV/ΔN ratio seems to be a good candidate for a *stopping* condition.[13,14] If we intend to compare saturation degrees for corpora of different sizes, we may do it by calculating (and then comparing), for all the corpora, the ratio ΔV/ΔN for the last segment representing X% of the whole corpus, denoted ΔV/ΔN(X%) and called *X% growth ratio*.

The method using *X% growth ratio* as a stopping criterion may be generalized to evaluate representativeness of a corpus for various phenomena. For example, in order to evaluate the minimum size of a balanced opinion corpus, we performed experiments involving opinion adjectives, which for most languages plays a primary role in expressing *negative* or *positive* aspects of opinions.

6.1 Experiments

The two experiments presented below illustrate the use of the stopping criterion based on the *saturation test*. In both cases we aimed for a corpus representative for lexical instruments for expressing opinions. Our simplifying assumption was that adjectives constitute the main lexical instrument to express opinion.[15] We also took into account the fact that not all adjectives are typically used to express or support an opinion. On the other hand, some typical opinion words may be used for other purposes than opinion expression. For both experiments we selected client opinions on hotels in Turkey and in Poland.

Experiment 1: Hotels in Turkey.

The first evaluation trials were performed for Turkey. Turkey is an attractive country for tourists and businessmen, with many hotels offering services at a wide range of

[10] A data gathering procedure is considered sound with respect to the given objective if it guarantees acquisition of all data necessary to reach this objective.

[11] A choice of measure units will of course affect the value of the 10% ratio.

[12] The value is to be fixed depending on what one needs the corpus for.

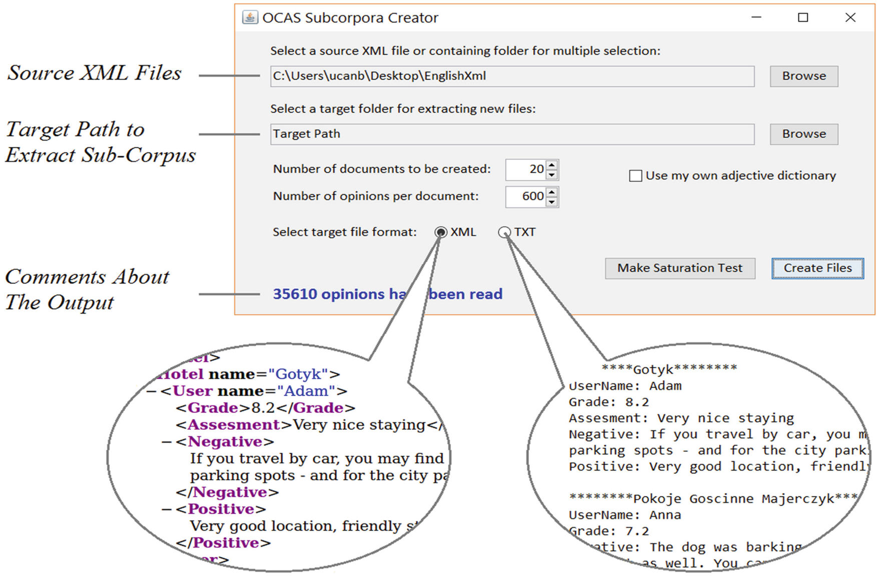
[13] According to Muller [12], in addition to the ΔV/ΔN ratio, it is also useful to consider the number (V1) of hapax legomena observed in the initial segment of the corpus of length N. For a fixed length of segments, the ratio ΔV/ΔN was shown to converge to V1/N with an increase in corpus length N [11].

[14] Note, however, that the stopping criterion considered here does not apply when a huge amount of text data is necessary to support statistical or neural-networks-based methods used to analyze texts.

[15] Julia Hartwig, a famous Polish poet known for her preference for adjectives, used to say that adjective is “the most important part of speech” [13].

standards and prices. For Turkey, Booking.com presents opinions for over 11,000 hotel service providers (mainly hotels and pensions) around the country,[16] with the major part being written in the local language (usually Turkish) or English. A test corpus was created for Booking.com opinions (1185) of about 300 semi-randomly selected hotels. In the first step, the most visited (for business or tourism) places were selected. When selecting hotels, special attention was paid to ensure a balanced representation of hotels of different categories (from 1 to 5 stars). For each hotel, up to 5 opinions of 19 words or more were selected. The variety of hotel locations and categories should make the test-corpus representative for a rich variety of phenomena. The fundamental question was when to stop constructing the corpus (stopping criterion).

To answer this question, we performed the saturation test for opinion adjectives. For the corpus of 1,185 opinions of 300 hotels, we identified 131 adjectives used as lexical expression of opinions (whether *positive* or *negative*), and we drew a vocabulary growth rate curve for these adjectives (Fig. 4). What we observe is a very slow linear increase in the number of the observed adjectives after having scanned about 20% of the corpus (60 hotels, over 230 opinions). It is hard to conclude that a reasonable stopping criterion has been satisfied. Of course, this observation does not need to be valid for other language categories or phenomena. On the other hand, it is clear that for some language phenomena, a representative corpus may be surprisingly small (Fig. 5).



Select a source XML file or containing folder for multiple selection: C:\Users\ucanb\Desktop\EnglishXml

Select a target folder for extracting new files: Target Path

Number of documents to be created: 20

Number of opinions per document: 600

□ Use my own adjective dictionary

Select target file format: ● XML ○ TXT

35610 opinions ha been read

```
  el>
 Hotel name="Gotyk">
– <User name="Adam">
    <Grade>8.2</Grade>
    <Assesment>Very nice staying</
  – <Negative>
      If you travel by car, you may find
      parking spots - and for the city pa
    </Negative>
  – <Positive>
      Very good location, friendly s
    </Positive>
   er>
```

```
    ****Gotyk********
UserName: Adam
Grade: 8.2
Assesment: Very nice staying
Negative: If you travel by car, you m
parking spots - and for the city park
Positive: Very good location, friendl

********Pokoje Goscinne Majerczyk***
UserName: Anna
Grade: 7.2
  ative: The dog was barking
      as well. You ca
```

Fig. 4. OCASSC v.1.

[16] 6,340 hotels in 28 the most visited cities.

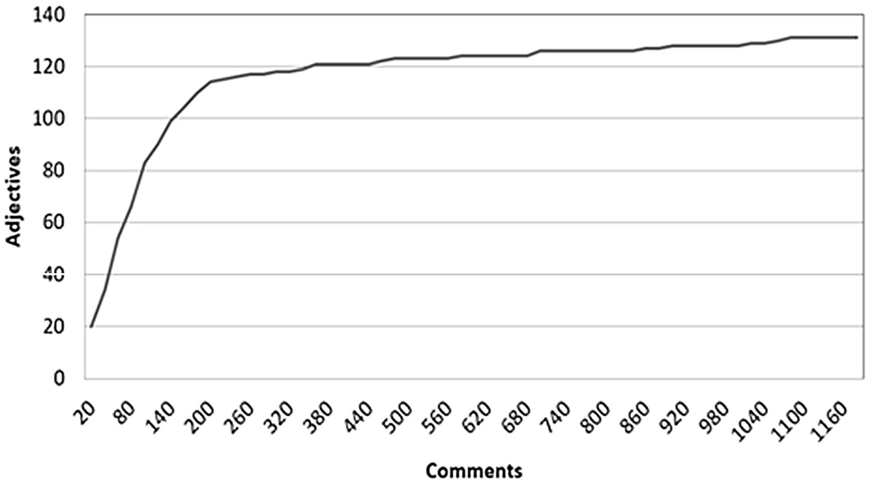


Fig. 5. Vocabulary growth curve for opinion adjectives (Experiment 1 – hotels in Turkey, 2015)

Experiment 2: Hotels in Poland.

The second experiment provided conclusions similar to those in the first one. We considered a corpus of opinions about hotels in Poland, written in English, with the same length constraint like in Experiment 1. We did not limit the number of opinions per hotel, nor did we apply any special selection of hotels. For the saturation test, we used the same OCASSC system. Its configuration requires *a priori* definition of the search space for the phenomena studied. In our case, the search space was determined by the list of words containing all adjectives that might be used to express opinions and which we considered interesting for our purposes.[17] To create this list ("dictionary"), we constructed a "teaching" corpus consisting of 2,040 opinions of hotels in the city of Poznan. We manually annotated all occurrences of adjectives used as opinion words, based on which we created a frequency list of all the annotated adjectives. This list contained 490 adjectives, which occurred 11,854 times. It is well known that the class of adjectives in English is open and that words may sometimes be used in untypical ways, in particular for another purpose than opinion expressing/supporting. New words (neologisms) and words used in an unusual way are rarely used as opinion words, and their frequency in the teaching corpus was low. In most cases, their frequencies in the corpus was 1 (or slightly higher, e.g., when a user used his or her neologism several times in his or her opinion). With such small frequencies, they can be ignored in saturation tests.

The saturation test was applied first for the corpus used in a pre-experiment, and then for various opinion subcorpora (of 2,040 opinions each) containing randomly selected comments about hotels in Poland. Figure 6 presents two pre-experiment saturation curves. The first one (continuous) represents the reference list composed of 312

[17] In OCASSC this list is called "dictionary" and is loaded by the user (see the function "use my own adjective dictionary").

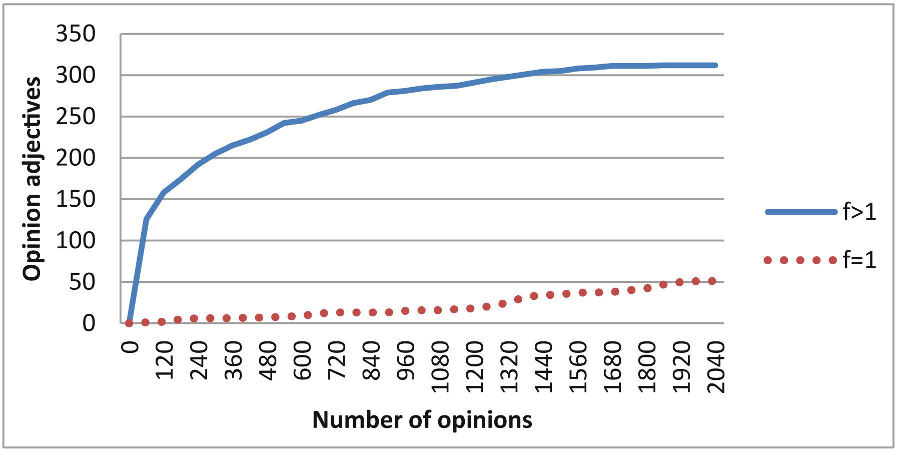


Fig. 6. Vocabulary growth rate curves for the pre-experiment subcorpus.

adjectives that occurred more than once (f > 1) in the pre-experiment corpus. The second saturation curve (dotted) represents the reference list composed of the remaining 51 opinion adjectives, that is, those that occurred just once in the pre-experiment corpus. This curve is increasing quasi-linearly, because hapaxes (i.e., words that occur once in the corpus) are distributed uniformly in the corpus, which is typical of small and medium size corpora.

Then we applied OCASSC to extract subcorpora of 2,040 opinions from the corpus of 34,800 Booking.com opinions written in English about all hotels in Poland. The corpus contained over 850,000 occurrences of 28,371 different words. To evaluate the degree of saturation for these subcorpora, we applied the *10% growth ratio* (ΔV/ΔN (10%)) (with *opinion adjectives* as a reference list and N standing for the number of opinions). In each of the observed cases, the ratio varied between 0.01 and 0.03.

The application of OCASSC to this corpus of 34,800 opinions and to the set of *all words* as the reference list shows that the corpus appears to be far from being lexically saturated, as the growth ratio ΔV/ΔN(10%) (computed for the variable N standing for opinions) is very high (0.453). Consequently, a significant growth of the observed vocabulary should be expected when proceeding to the further extensions of the corpus.

The growth ratio for words is much smaller than that for opinions: the ΔV/ΔN (10%) ratio for *all words* (the reference list as above, N standing for the number of words) in this corpus equals 0.0186. This figure is still too high to state that the corpus is lexically saturated. Compare it with the ΔV/ΔN(10%) ratio (with respect to text words) of 0.01 calculated for the well-known corpus of short meteo reports of the journal “Le Monde”, with 20,000 occurrences of 606 words [12]; but note also the difference in the absolute sizes of the corpora, that is, 850,000 and 20,000.

7 Final Remarks

In the paper we present a system for opinion corpus acquisition from the Internet. The corpora obtained using the proposed tools may serve to

- study how people use language to express their opinions,
- describe the emotional content of opinion texts, and
- study various socio-cultural factors and impact they have on the cross-language and cross-ethnic comparability of opinions.

In addition, studying opinion adjectives is relevant to various aspects of opinion texts.

A practical utility of the experiments is that they illustrate a method to decide what is a reasonable size of samples to be extracted from a general, balanced, and representative (often huge) corpus for particular language engineering works.

The corpus of hotel opinions is now under construction and will help us in future research addressing some of the above-mentioned issues.

References

1. Collins English Dictionary—Complete & Unabridged 2012 Digital Edition; © William Collins Sons & Co. Ltd. 1979, 1986 © HarperCollins Publishers (1998, 2000, 2003, 2005, 2006, 2007, 2009, 2012)
2. Charaudeau, P., Maingueneau, D.: Dictionnaire d'Analyse du Discours. Seuil, Paris (2002)
3. Stoyanov, V., Cardie, C., Litman, D., Wiebe, J.: Evaluating an opinion annotation scheme using a new multi-perspective question and answer corpus. In: Shanahan, J.G., Qu, Y., Wiebe, J. (eds.) Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series, vol. 20, pp. 77–91. Springer, Dordrecht (2006)
4. Ptaszynski, M., Rzepka, R., Araki, K., Momouchi, Y.: Automatically annotating a five-billion-word corpus of Japanese blogs for affect and sentiment analysis. In: Proceedings of the 3rd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, Jeju, Republic of Korea, pp. 89–98. Association for Computational Linguistics, Stroudsburg (2012)
5. Esuli, A., Sebastiani, F.: SentiWordNet: a publicly available lexical resource for opinion mining. In: Proceedings of the 5th Conference on Language Resources and Evaluation, LREC 2006, pp. 417–422. European Language Resources Association, Genoa (2006)
6. Vetulani, Z., Vetulani G., Kochanowski, B.: Recent advances in development of a lexicon-grammar of Polish: PolNet 3.0. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016, pp. 2851–2854. European Language Resources Association, Paris (2016)
7. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta, pp. 1320–1326. European Language Resources Association, Genoa (2010)

8. Read, J.: Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: Knight, K., Ng, H.T., Oflazer, K. (eds.) 43rd Annual Meeting of the Association of Computational Linguistics 2005, Proceedings of the Conference, University of Michigan. The Association for Computer Linguistics, New Brunswick (2005)
9. McEnry, T., Hardie, A.: Corpus Linguistics: Method. Theory and Practice. Cambridge University Press, Cambridge (2012)
10. Kittredge, R.: Semantic processing of texts in restricted sublanguage. Comput. Math Appl. **9** (1), 45–58 (1983)
11. Vetulani, Z.: Linguistic problems in the theory of man-machine communication in natural language. Universitätsverlag Dr, N. Brockmeyer, Bochum (1989)
12. Muller, Ch.: Peut-on estimer l'étendue d'un lexique? Cah. Lexicol. **27**, 3–29 (1975)
13. Legieżyńska, A.: Julia Hartwig. Wdzięczność. Wydawnictwo Uniwersytetu Łódzkiego, Łódź (in Polish) (2017)

Less-Resourced Languages

Issues and Challenges in Developing Statistical POS Taggers for Sambalpuri

Pitambar Behera[1(✉)], Atul Kr. Ojha[2], and Girish Nath Jha[1,2]

[1] Centre for Linguistics, Jawaharlal Nehru University, New Delhi, India
pitambarbehera2@gmail.com, girishjha@gmail.com
[2] Special Centre for Sanskrit Studies,
Jawaharlal Nehru University, New Delhi, India
shashwatup9k@gmail.com

Abstract. Low-density languages are also known as lesser-known, poorly-described, less-resourced, minority or less-computerized language because they have fewer resources available. Collection and annotation of a voluminous corpus for the purpose of NLP application for these languages prove to be quite challenging. For the development of any NLP application for a low-density language, one needs to have an annotated corpus and a standard scheme for annotation. Because of their non-standard usage in text and other linguistic nuances, they pose significant challenges that are of linguistic and technical in nature. The present paper highlights some of the underlying issues and challenges in developing statistical POS taggers applying SVM and CRF++ for Sambalpuri, a less-resourced Eastern Indo-Aryan language. A corpus of approximately 121 k is collected from the web and converted into Unicode encoding. The whole corpus is annotated under the BIS (Bureau of Indian Standards) annotation scheme devised for Odia under the ILCI (Indian Languages Corpora Initiative) Project. Both the taggers are trained and tested with approximately 80 k and 13 k respectively. The SVM tagger provides 83% accuracy while the CRF++ has 71.56% which is less in comparison to the former.

Keywords: Low-density language · Parts of speech tagger · SVM CRF++ · Sambalpuri · Eastern IA Language

1 Introduction

Low-density languages have fewer resources in terms of the availability of voluminous corpus [1] for NLP applications. The unavailability of corpus for a low-density language proves to bear adverse impacts on its future NLP development. As rightly pointed out by Ostler [2], languages that lack active participation in the electronic media are doomed to be endangered in the forthcoming years. Most of these languages are either dialects or languages with no government recognition. As a result, the situations of these languages in South Asia in general and in Indic languages, in particular, are 'relatively bleak' [1]. Although India is a land of more than 6000 languages with five prominent diverse language families [3] only 22 are scheduled and the rest are fighting for their survival.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 393–406, 2018.
https://doi.org/10.1007/978-3-319-93782-3_28

This paper is concerned with demonstrating the issues and challenges in developing statistical parts of speech taggers for a low-resource language nomenclatured as Sambalpuri or Kosli. The paper has broadly three objectives. Firstly, it highlights the issues in corpus collection with regard to non-uniform orthographic language standards and non-Unicode encodings of the written text. Secondly, it further attempts to bring out the issues in annotation having without any guideline. Finally, it demonstrates the challenges in developing statistical POS taggers for Sambalpuri owing to the typical, unobserved and language-specific linguistic nuances.

2 Sambalpuri: A Low-Density Eastern IA Language

Sambalpuri (ISO 639-3) is an Eastern Indo-Aryan (EIA) language and is also known as Dom, Kosli, Koshal, Koshali, Western Odia[1] etc. It is spoken in the ten districts of western and south-western Odisha which comprise Bargarh, Bolangir, Kalahandi, Sonepur, Sambalpur, Jharsuguda, Sundargarh, Deogarh, Boudh, Nuapada and Athmallik sub-division of Angul district. In comparison to its sister languages such as Maithili, Awadhi, Bhojpuri, Magahi, Angika, Bengali, Assamese, Odia and many others, Sambalpuri has not gained much attention; neither from linguists nor from the government. It is really quite obvious to affirm that it shares the genetic affinity with the Indo-Aryan language family by observing some of its linguistic features [4]. Although it has 70–76%[2] lexical similarity with Standard Oriya [5], it is syntactically a distinct language [6]. Recently, Behera and Dash [7] have reported a work-in-progress Sambalpuri dictionary which has the present size of around 500 lexicons pertaining to nouns in three domains such as flora and fauna, kinship and body parts.

3 Salient Linguistic Features

Less-described languages have some of the most interesting linguistic features that are typical and language-specific. Some of the features like agglutination, classifiers, serial verbs, multi-words, compounds etc. account for the less accuracy of any of the statistical NLP applications in general. Some of them are discussed vividly below in four sub-sections with regard to Sambalpuri: agglutination, classifiers, reduplication and compounds.

3.1 Agglutination

In an agglutinative language, words are made of a linear sequence of distinct morphemes each of which corresponds to a definite meaning[3]. In Odia, the categories such as “suffixes, postpositions, and case endings agglutinate with the verbs, nouns, adverbs or pronouns” [8–13]. Similar is the case with Sambalpuri language. Behera [14] has

[1] https://www.ethnologue.com/language/spv.

[2] However there is no satisfactory explanation about the methodology adopted and the number of lexical items analysed on the basis of which this conclusion has been arrived at.

[3] http://www.glossary.sil.org/term/agglutinative-language.

recently averred the fact that Sambalpuri has some agglutinative nominal morphology and assumes that it has some Dravidian features as Odia possesses.

For instance,
kʰɑɛbɑr-ke ‘to eat’
lʊk-ər ‘people’s’
bɑhɑr-ke ‘to outside’
mɒr-nʊ ‘from me’

In the examples instantiated above, all the case endings or markers /ke/, /ər/, /nʊ/ agglutinate with their head categories verb, noun and pronoun. /ke/ which is equivalent to the English infinitive and preposition ‘to’ is alternating here with both verb and spatial noun /kʰɑɛbɑr/ and /bɑhɑr/ respectively.

3.2 Classifiers

Classifier is one of the most prominent phenomena in Indian languages; especially in the Tibeto-Burman languages and Dravidian languages. In addition to some of the EIA languages like Bengali [15], Odia [16] and [10, 11], Bhojpuri [17], Marathi [18] and so on, it is a dominant linguistic feature in Sambalpuri as well. The classifiers mainly occur either as proper classifiers, attached to numerals or to the quantity word /ket̪e/ ‘how many, some’, or as indefinite markers, in combination with the suffix” [16] as /ʈe/, /ʈɑ/, /kʰə̃ɖe/, /ɟʰəne/ etc. in Sambalpuri. One of the rarely observed phenomena of Indian languages found in Sambalpuri is that classifiers also occur with post positions. Recently, it has been asserted that classifiers alternation with other categories takes place largely in Sambalpuri [14].

For instance, /mɒr lekʰe-ʈɑ/ ‘like me’.

3.3 Reduplication

Reduplication is the repetition of a syllable, segment or some part or whole of a lexical or phrasal unit which leads to a semantic or grammatical modification of the component in question. There are two types of reduplication: partial and total. In total reduplication, the whole part of the base is reduplicated and in the partial reduplication, some part is reduplicated [19]. In the following instances, the first two are fully reduplicated while the rest of the following are partial. In the partial reduplication, the final syllables /-nɑ/ of both the words are reduplicated like in the first example whereas the final example contains the reduplications of the initial syllables /hʊ-/.

For instance,

çɪk çɪk ‘shining’
d̪ʰɪre d̪ʰɪre ‘slowly’
ɟənɑ sʊnɑ ‘known’
hʊʈɑ hʊʈɪ ‘abusing’

When a sequence of verbs occurs in a chronology, they are called serial verbs [20] and some of them are reduplicative in nature. In the below-instantiated example, it is quite confusing as to how to annotate the verbal occurrences. Because the initial verb

/d̪əʊɽɪ/ is a non-finite verb followed by a verbal reduplication which is functioning like a manner adverb modifying the finite following verb /pəleɪgəlɑ/. The issue here is how to annotate the verbal reduplication. In these instances the morphological features of each lexical item have been taken into consideration for deciding the annotation of label.

For example,

se mɑrɪkərɪ d̪əʊɽɪ d̪əʊɽɪ pəleɪgəlɑ
he V-Nonfinite V-reduplication V-Finite
“He went away beating.”

3.4 Compounds

Compound or Sandhi is one of the most productive linguistic phenomena which is quite typical in most of the worlds’ languages in general and in Indian languages in particular. There are three basic types of compounds: vowel, consonant and visarga. In the following instance, the first word is an adjective and the second is a noun, but when get combined they comprise a nominal category. Since Sambalpuri is a head final language, the annotation label is decided on the basis of the category of the head. Here the head is a nominal element and hence the judgment goes in favor of the category of the label of the head word.

For example,

sɔd̪\JJ + pɔt̪ʰɔ\N_NN = sɔt̪pɔt̪ʰɔ\N_NN ‘good path’

So, in the above example, the decision whether to annotate the word as an adjective or noun goes for the right-headedness feature of Sambalpuri. This feature is typical to most of the IA languages and the word /sɔt̪pɔt̪ʰɔ/ is labelled as a noun.

4 Methodology

This section deals with (a) the total corpus collected in four major domains, (b) the BIS annotation guideline adapted for Sambalpuri, (c) size of the corpus for training, testing and development stages and (d) features selection for SVM and CRF++ POS taggers.

4.1 Corpus Size

The tabulated data (see Table 1) demonstrates the total corpus size collected for developing the Sambalpuri POS taggers. The whole corpus size comprises of five domains, viz. literature, sports, tourism, entertainment, and miscellaneous. The highest corpus size is registered in the domain of entertainment i.e., approximately 40 k while the ‘miscellaneous’ section accounts for the lowest number of data.

4.2 Corpus Annotation

The whole Sambalpuri corpus[4] is annotated using the ILCI Ann Tool[5] [21] following the BIS-ILCI tagset (see Table 2) devised for Odia language since there is no tagset

[4] This is the very first POS tagset developed for Sambalpuri.

[5] http://sanskrit.jnu.ac.in/ilciann/index.jsp.

Table 1. Domain-wise corpus distribution

Domains	Tokens
Literature	30, 344
Sports	21, 121
Tourism	26, 767
Entertainment	40, 554
Miscellaneous	2, 424
Total	1, 21, 210

available for it. The BIS tagset is a hierarchical set designed by the POS Standardization Committee appointed by the Department of Information and Technology, Government of India. It has a total number of 11 categorical labels at the top level and 39 fine-grained labels for the annotation. The tagset is framed keeping in view both the fineness and coarseness or flat and hierarchical structures in view. The table below contains the nomenclatures of all the categories in the second column, annotation labels in the third and categorical IPA examples in the fourth.

Table 2. BIS parts of speech tagset adapted for Sambalpuri

Sl. No.	POS Categories	Annotation Labels	Examples of Sambalpuri in IPA
1	Noun	N	
1.1	Common	N_NN	pət̪ər, pɪt̪əl, bʰabna, mʊnʊs
1.2	Proper	N_NNP	ram, hɪmaɭəj, gəŋad̪ʰər meher bɪsʊəbɪd̪jaləj, səmbəlpʊr etc.
1.3	Verbal	N_NNV	pəɽʰa, pəhə̃ra, ɖɛga, nacbarʈa
1.4	Spatial & temporal	N_NST	agke, pəcʰaɽe, pəre, pʊrʊb, etc.
2	Pronoun	PR	
2.1	Personal	PR_PRP	mʊĩ, tʊɪ, apən, se etc.
2.2	Reflexive	PR_PRF	nɪɟe etc.
2.3	Relative	PR_PRL	ɟahar, ɟahãkər, ɟenmankər
2.4	Reciprocal	PR_PRC	nɪɟər bʰɪt̪re, d̪ʊhe etc.
2.5	Wh-word	PR_PRQ	kɪe, kahar, ken mane, etc.
2.6	Indefinite	PR_PRI	ənjər, kɛnsɪ, kehɪ etc.
3	Demonstrative	DM	
3.1	Deictic	DM_DMD	ɪ, se, ɪguɽakə, seguɽakə etc.
3.2	Relative	DM_DMR	ɟenguɽakə, ɟahar
3.3	Wh-word	DM_DMQ	kaɳa, kenər, kenguɽakər etc.
3.4	Indefinite	DM_DMI	ʊnɪã, kɛnsɪ etc.
4	Verb	V	
4.1	Main	V_VM	
4.1.1	Main	V_VM	sʊ, d̪əʊɽ, d̪ɛkʰ etc.

(*continued*)

Table 2. (*continued*)

4.1.2	Non-finite	V_VNF	kʰɑɪ kərɪ, nɑcɪ nɑcɪ, etc.
4.1.3	Infinitive	V_VINF	kʰɑɛbɑrke, kʰɑɪbɑr lɑgɪ, nɑcbɑr etc.
4.1.5	Gerund	V_VNG	kʰɑɪt̪ʰɪbɑr, kʰɑut̪ʰɪbɑr etc.
4.2	Auxiliary	V_VAUX	ʊcɪt̪, d̪ərkɑr, kərɪ, t̪ʰɪbɑr etc.
5	Adjective	JJ	bʰəl, ut̪t̪əm, sũd̪ər etc.
6	Adverb	RB	
7	Postposition	PSP	sɑŋe, lekʰe, lɑgɪ etc.
8	Conjunction	CC	
8.1	Coordinator	CC_CCD	kɑ̃ hɛlɑɟe kɪ, kɑrən, ɑʊ, etc.
8.2	Subordinator	CC-CCS	ɟəd̪ɪ t̪ebe, ɟet̪ebɛlɛ set̪ebɛlɛ, ɟe, bəlɪ etc.
8.3	Quotative	CC_CCS_UT	ɑɑre, həɛ lɒ, həgɒ, ɑgjɑ̃ etc.
9	Particles	RP	
9.1	Default	RP_RPD	bʰɪ, hĩ, t̪ɔ etc.
9.2	Classifier	RP_CL	gʊʈe, d̪ʊɪʈɑ, kʰə̃ɖe etc.
9.3	Interjection	RP_INJ	ʋɑh, həɛ, ɑh, oho etc.
9.4	Intensifier	RP_INTF	ət̪ɪ, kʰʊb, bəhʊt̪, ɟəbər etc.
9.5	Negation	RP_NEG	nɑĩ, nʊhe, nɪ, nɪhɑ etc.
10	Quantifiers	QT	
10.1	General	QT_QTF	t̪ʰʊɖe, bɛsɪ, ʈɪke, gʊɖɑd̪ʊ etc.
10.2	Cardinal	QT_QTC	ek, d̪ɒ, t̪ɪn, cɑr etc.
10.3	Ordinal	QT_QTO	pəhɛlɑ, d̪ʊsrɑ, t̪ɪsrɑ etc.
11	Residuals	RD	
11.1	Foreign words	RD_RDF	languages of the other scripts except Odia
11.2	Symbol	RD_SYM	mathematical and other symbols (#, [, {, %, $, <, >, (,), *, @,)
11.3	Punctuation	RD_PUNC	(, ; : ‘’ “” :- etc.)
11.4	Unknown	RD_UNK	Tags that are left undecided
11.5	Echo word	RD_ECH	bɑgʰ-pʰɑg, kəʈɑ-cʰəʈɑ etc.

4.3 Data Size for the Taggers

The tabulated data (see Table 3) represents the different data sets applied to develop the statistical taggers. The total number of training data used for developing the taggers amounts to around 80 k. Initially, the tagger is trained with around 50 k with manually annotated data and later, the development set consists of 30 k which was automatically tagged and manually validated. After the training period, the testing is conducted with a set of approximately 13 k corpus size tokens.

Table 3. Training and testing data sets

Data sets	Tokens
Training	80, 288
Testing	12, 791

4.4 Developing POS Taggers

Two statistical taggers are developed for Sambalpuri; the first one is trained with SVM [22, 23] and the second is with CRF++ [24]. So far as the former is concerned, learning phase contains medium verbose (-V 2) and the mode of learning and tagging is set to left-right-left (LRL). The rest of the features like sliding window, feature set, feature filtering, model compression, C parameter tuning, Dictionary repairing and so on are set to the default mode. On the other hand, the latter is trained with the unigram method.

5 Issues and Challenges

This section is divided into three major sub-sections: corpus-related, human annotation-related and tagger-related issues.

5.1 Corpus-Related Issues

The issues pertaining to the corpus collection are vividly discussed: corpora collection, unavailability of Unicode encoding, non-standard usage of the language, different writing conventions and Hindi-like constructions.

Corpus Collection: A number of corpora have been developed for various languages like English and some European languages. Considering the situations in non-scheduled (lesser-known) Indian languages, it is quite unfavorable in comparison to the scheduled languages since some of the Indian institutions have either worked on or are presently developing language resources and technologies for the latter languages only. Because of the indifference of the government towards the lesser-known languages, the former are getting disempowered gradually. The institutions and projects that have worked for the corpus collection in scheduled Indian languages are IIIT-Hyderabad, CIIL-Mysore, ILCI-JNU and TDIL.

Unavailability of Unicode Encoding: Since low-density languages are less-resourceful or with no resource the software available for them are also less in number. This leads to the non-Unicode encodings which is not favorable for the development of NLP applications. The whole corpus has been converted into UTF-8 encodings using Akruti Text Converter[6]. There are different linguistic issues in the corpus itself such as non-standard usage, non-uniform Orthographic forms and Hindi-like constructions.

[6] https://22bc339da9ca3e2462414546a715752e4c2c5e0d.googledrive.com/host/0B5rBGd680WZFemVLa3RxY0preE0/AkrutiUnicode.

Non-standard Usage: Sambalpuri is not a scheduled Indian language and is written and spoken with varying standards in different regions of the western and south-western Odisha. For example: Sambalpuri, Bargadia (spoken in Bargarh), Bolangiri/a (spoken in Bolangir district), Sundargadi/ia (spoken in Sundargarh), Deogarhia (spoken in Deogarh region), Boudia (spoken in Bouddha district) etc. There are some dialectical variations among the people of Sambalpuri speaking track. The table (see Table 4) demonstrates dialectal variations of Sambalpuri with reference to negative morpheme ‘no’, adverbs ‘now’ and ‘this way’. Lexical similarity within the varieties of Sambalpuri is considerably high which ranges from 90 to 95% [5]. This similarity matrix was made by comparing Bargarhi, Bolangiri and Jharsuguda varieties with Sambalpuri.

Table 4. Dialectal variations in Sambalpuri (adapted from [25])

Variety of Sambalpuri	Negative ‘no’	Adverb ‘now’	Adverb ‘this way’
Bargarh	nʊhe/nıhe	ıhɑd̪e/ɛc^{h}ɛn	ɪɑɖe/ɪp^{h}ɑle
Bolangir	nĩ	ɛk^{h}ɛn	
Kalahandi	nĩ	ɛk^{h}ɛn	ɪbɑʈe
Sambalpur	nıhe/nʊhe	ɪc^{h}nɪ	ɪɑɖe
Sundargarh	nĩ		
Bouda		ɪgəɖɪ	ɪɑɖkʊ

Different Orthographic Conventions: A large number of words in Sambalpuri has different orthographic conventions; especially the ligatures. In Sambalpuri, there are several writing conventions used for a given word form because of the non-uniform usage of language.

For instance, in the following examples two forms are used for one word with two of them having different POS labels with the change of form.

କାଞ୍ଜେଁ N_NN, କାଁଜେ DM_DMQ

କନ୍ଡକ୍ଟର N_NN, କଣ୍ଡକ୍ଟର୍ N_NN

କାନ୍ଚନ N_NN କାଞ୍ଚନ N_NNP

This non-standard usage of the words creates issues during both manual and automatic annotation since their POS labels vary with the varying conventions.

Hindi-like Constructions: Sambalpuri is more like Hindi than Odia which accounts for the fact that the western region, where it is spoken, is situated just adjacent to Chattisgarh and Jharkhand where the influence of Hindi is largely felt. In the examples instantiated below /bɑʋəɟʊd̪/ and /ke/ are postpositions as used in Hindi while the Hindi-like indefinite and reflexive pronouns are also used.

For instance,

/bɑʋəɟʊd̪/ PSP
/hər ek/ PR_PRI /ke/ PSP
/əpnɑ/ PR_PRF /əpnɑr/ PR_PRF

5.2 Issues Pertaining to Human Annotation

One of the prominent challenges is that which pertains to the annotation of the corpus. For annotation of a voluminous corpus and to maintain consistency, one needs to have a standard tagset. Owing to the fact that a large number of languages like Sambalpuri being less-described or less-studied, it is quite daunting to devise a tagset. If one adopts and adheres to the tagset devised for a language of close proximity, then they may either compromise with the saliency of the linguistic data or may end up filling different slots for labels and not researching by delving deep into some interesting structures. For instance, there are large numbers of homophonous words that can neither be included in the reduplicated nor can they be labelled as echo.

Reduplicated Expressions: Generally, in Indian languages the reduplicated expressions follow the meaningful word. Contrastingly, in Sambalpuri many of the reduplicated parts precede the meaningful words (see Sect. 3.3). For instance, in the conjunct verb (adjective + finite verb), /cʰɪcʰɪ/ is the meaningless reduplicated part which is preceding the meaningful part /bɪcʰɪ/ 'scattered'.

For example,

cʰɪcʰɪ\RD_ECH bɪcʰɪ\JJ heɪcʰən 'have got scattered'

Similarly, in the following verbal reduplication, the meaningless part is preceding the verbal part.

For example,

kʊt̪\RD_ECH kʊt̪eɪ\V_VM d̪ɛlɑ\V_VM_VF 'has tickled'

These kinds of constructions pose significant linguistic challenges for the human annotators as to how to label them and so is for the statistical tagger.

Verb-less Constructions: In Sambalpuri and many sister languages such as Odia, Bengali, Assamese [26] verb-less constructions or covertly present verbs are commonly used. These constructions are used with adjectives in place of verbs. Therefore, the tagger also labels some of these adjectives as finite verbs because of the annotation of these constructions in the training data.

For example,

sɑsʋɔt̪\N_NNP bɑbʊr\N_NN cɑnvɑs\N_NN ʊsɑr\JJ pɪsɑr\RD_ECH .\RD_PUNC
"Saswat Babu's canvass is quite large."

In the above example, /ʊsɑr\JJ pɪsɑr\ RD_ECH/ is the reduplicated adjectival phrase which satisfies the need of the verb.

Onomatopoeic Constructions: Onomatopoeic words are the imitation of a sound associated phonetically with its describing referent. These following expressions are parts of the multi words because individually these words do not have meaning, but when combined they are manner adverbs. As per the ILCI guideline, if we annotate the first sound as noun and the following words as echo-words (RD_ECH), we are missing relevant linguistic information.

For instance,

bʰɛ̃ bʰɛ̃ ‘loudly’
ɟʰɒ ɟʰɒ ‘heavily’
d̪ʰɔ̃ pɔ̃ ‘gasping’
bʰɒ bʰɒ ‘bark’

Agglutination of Classifiers with Postpositions: Agglutination (see Sect. 3.2) is one of the common features in Odia [10] and Sambalpuri along with some IA languages like Bengali and Marathi [18]. In Sambalpuri, one of the peculiar constructions with agglutination is that the classifiers and postpositions agglutinate with each other which is also rarely found in the most agglutinating Dravidian languages. Here, to annotate these constructions as classifiers (RP_CL) or postpositions (PSP) is quite difficult.

For instance,

/bɑgɪr-ʈɑ/ ‘as-CL’
/lekʰeʈɑ/ ‘like-CL’

Similarly, in the example below, it is quite difficult to decide the annotation labels for both the human and automatic annotation. The reason is the complexity in deciding the head label of the words. The word /d̪ʊɪ-ʈɑ/ comprises of two components, a cardinal and a classifier morpheme. Both of these categories have separate labels in the BIS scheme for Sambalpuri. Therefore, if one annotates the word as cardinal, they are compromising with the other label or linguistic information.

For instance,

d̪ʊɪ-ʈɑ (d̪ʊɪ\QT_QTC ʈɑ\RP_CL)

5.3 Issues Related to Automatic Annotation

These issues are mostly pertained to tagger-related ambiguities and some other linguistic errors.

Ambiguity Issues: The data (see Table 5) represented below demonstrates that there are different types of ambiguous sets of classes and their accuracy rates. All the ambiguity classes are divided into 244 classes and they are generated automatically by the SVM tool.

Two-label Sets: This section includes the ambiguous words with two conflicting labels. The most commonly ambiguous tags are coordinating-subordinating conjunctions, coordinating conjunction-general quantifier, deictic-interrogative demonstrative and so on.

ɑʊ (CC_CCD or QT_QTF)

Table 5. Ambiguity Classes

Classes	Label Sets
2 Sets:	CC_CCD_CC_CCS, CC_CCD_QT_QTF, DM_DMD_DM_DMQ
3 Sets:	JJ_N_NST_V_VM_VF, RD_ECH_V_VM_V_VM_VNF, RP_NEG_V_VM_V_VM_VF
3 Sets>	V_VAUX_V_VM_V_VM_VF_V_VM_VNF, RP_INJ_RP_NEG_V_VM_V_VM_VNF, RD_UNK_RP_INJ_RP_RPD_V_VM_V_VM_VNF

For example,

mʊĩ ɑʊ\CC_CCD mɒr bɑpɑ “I and my father”
mət̪e ɑʊ\QT_QTF kʰɑnɑ d̪ərkɑr “I need some more food”.

In the above examples, the first one suggests that the word /ɑʊ/ is a coordinating conjunction coordinating two noun phrases while the second one states that it is a general quantifier used as a pre-modifier of the following head noun.

Three-Label Sets: This section contains the ambiguous words having three labels. The most commonly ambiguous tags are most-expectedly adjective-temporal nouns-finite, negative-main-finite verb and so on.

bɑhɑr (V_VM or N_NST or JJ)

For instance,

bɑhɑr\V_VM_VF gʰərʊ\N_NN “Come out of the house”.
se pəlɑlɑ bɑhɑr\JJ gʰərʊ\N_NN “He went away from the front room”.
bɑhɑr-ke\N_NST ɑs\V_VM_VF “Come to outside”.

In the instances mentioned above, the word form /bɑhɑr/ has three different POS labels. The first one is annotated as a finite main verb as the sentence is an imperative sentence and the covert subject is the second person pronominal. In the second example, it is labelled as an adjective as it modifies the following noun whereas the third one is a spatial noun as it refers to a location.

More than Three-Label Sets: The words having more than three labels are encapsulated in this part. For instance, main-auxiliary-nonfinite-finite verbs, unknown-interjection-default particle and so on.

kərɪ (V_VAUX or V_VM or V_VM_VF or V_VM_VNF)

For instance,

kʰɑɪ\V_VM kərɪ\V_VAUX ɑsle\V_VM_VF
kɑm\N_NN kərɪ\V_VM ɑsle\V_VM_VF
kərɪt̪ʰɪlɑ\V_VM_VF
kʰɑɪkərɪ\V_VM_VNF ɑsle\V_VM_VF

The verbal word form /kərɪ/ has more than three labels in the corpus and which is rightly so. It can be used as main, auxiliary, finite and non-finite verbs as instantiated in the above examples.

6 Results and Discussion

In spite of the different issues and challenges, statistical taggers developed for Odia achieves accuracies of 94% and 89% by SVM and CRF++ respectively (Behera [10]). The results (SVM 83% and CRF++ 71.56%) for Sambalpuri are comparatively lesser than Odia which accounts for the fact that Sambalpuri has no standardized orthographic convention and hence different regional varieties use the language in their own ways.

The first and foremost point to emphasize for a low-density language is the large-scale writing on the social media using its own script with a soul objective of developing language resources. If there is less availability of the corpus, then one can also take the assistance of mathematical modelling to achieve a higher accuracy rate. So far as the tagset-related issues are concerned, one can take the labels already used by a closely-related language spoken in the region for annotation job by incorporating it on their convenience. With regard to issues in annotation, one can take the exemplary labels from different tagsets developed for Indian languages. For example, we can incorporate WRB label from the IIIT-Hyderabad for the interrogative adverb. The reduplicative expressions need to be considered seriously as they are the most vital parts of the language and they behave quite differently in Sambalpuri. Therefore, it can be averred that labels for reduplication (RD_REDP), possessive pronouns (PR_POS) and demonstratives (DM_POS), interrogative adverbs (WRB) can be introduced. For handling agglutination, a stemmer or a lemmatizer could be used with statistical POS taggers. For punctuations, fine-grained labels should be incorporated based on their functions in a given context as they can be used as coordinators, section headers, list item markers and so on. Finally, the standardization of the language would help solve many of the issues by providing consistency in both the human and statistical annotations.

7 Conclusion

In this paper, we have discussed about different issues and challenges in terms of both corpus collection, annotation and tagger-related issues in detail for a less-resourced language, i.e. Sambalpuri. The results (SVM 83% and CRF++ 71.56%) of the statistical taggers for Sambalpuri in the present study would not only prove to be beneficial for its own future NLP research and development but also would be advantageous for any other morphologically-rich less-resourced language from around the world. At a later stage, we can further use a lemmatizer or stemmer to handle the agglutination issue and incorporate some of the solutions proposed in the research. Furthermore, these POS taggers could be potentially used for developing morph analyzer, chunker, parser and hopefully for enhancing the accuracy of machine translation. For its future development, a full-fledged online lexical dictionary using Language Explorer, Lexique Pro & Toolbox can also be prepared.

References

1. McEnery, T., Baker, P., Burnard, L.: Corpus resources and minority language engineering. In: LREC (2000)
2. Ostler, N.: Language technology and the smaller language. ELRA Newsl. **4**(2) (1999)
3. Abbi, A.: A Manual of Linguistic Fieldwork and Structures of Indian Languages, vol. 17. Lincom Europa (2001)
4. Kushal, G.: Case and agreement in Sambalpuri. M. Phil. Thesis, Centre for Linguistics, Jawaharlal Nehru University, New Delhi, Delhi (2015)
5. Mathai, E.K., Kelsall, J.: Sambalpuri of Orissa, India: A Brief Sociolinguistic Survey. SIL International (2013)
6. Tripathy, B.: Sambalpuri semantics. Graduate Thesis, Sambalpur University, Sambalpur (1984)
7. Behera, P. Dash, B.N.: Documenting Sambalpuri-Kosli: the case of a less-resourced language. Indian J. Appl. Linguist. (IJOAL). Bahri Publications (0379-0037), June 2017. (accepted)
8. Padhy, H.H., Mohanty, S.: Designing hybrid approach spell checker for Oriya. Int. J. Latest Trends Eng. Technol. **2**(4), 156–160 (2013)
9. Jena, I., Chaudhury, S., Chaudhry, H., Sharma, Dipti M.: Developing Oriya morphological analyzer using Lt-Toolbox. In: Singh, C., Singh Lehal, G., Sengupta, J., Sharma, D.V., Goyal, V. (eds.) ICISIL 2011. CCIS, vol. 139, pp. 124–129. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19403-0_20
10. Behera, P.: Odia parts of speech tagging corpora: suitability of statistical models. M. Phil. Thesis, Centre for Linguistics, Jawaharlal Nehru University, Delhi (2015)
11. Behera, P.: Evaluation of SVM-based automatic parts of speech tagger for Odia. In: Proceedings of WILDRE-3 (LREC-2016), Portoroz, Slovenia, pp. 32–38 (2016). ISBN: 978-2-9517408-8-4
12. Behera, P.: An experimentation with the CRF++ parts of speech tagger for Odia. Lang. India **17**(1) (2017). ISSN: 1930-2940
13. Ojha, A.K., Behera, P., Singh, S., Jha, G.N.: Training & evaluation of POS taggers in Indo-Aryan languages: a case of Hindi, Odia and Bhojpuri. In: Proceedings of LTC-2015, Poland, pp. 524–529 (2015)
14. Behera, P.: Issues and challenges in corpus collection and annotation of Sambalpuri: the case of a lesser-known language. Language Forum, Bahri Publications, June 2018. ISSN 0253-9071. (accepted)
15. Bhattacharya, T.: The structure of the Bangla DP. Doctoral Dissertation, University College, London (1999)
16. Neukom, L., Patnaik, M.: A Grammar of Oriya. Seminar für Allgemeine Sprachwissenschaft der University, Zürich (2003)
17. Shukla, S.: Bhojpuri Grammar. Georgetown University Press, Washington, D.C. (1981)
18. Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Jha, G.N.: A common parts-of-speech tagset framework for Indian languages. In: LREC (2008)
19. Abbi, A.: Reduplication in South Asian Languages: An Areal, Typological and Historical Study. Allied Publishers Pvt. Ltd., Chennai (1992)
20. Jha, G.N., Hellan, L., Beermann, D., Singh, S., Behera, P., Banerjee, E. Indian languages on the TypeCraft platform - the case of Hindi and Odia. In: LREC, Iceland (2014)

21. Kumar, R., Kaushik, S., Nainwani, P., Banerjee, E., Hadke, S., Jha, G.N.: Using the ILCI annotation tool for POS annotation: a case of Hindi. In: 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2012), New Delhi, India, March 2012
22. Joachims, T.: Making large scale SVM learning practical. Universität Dortmund (1999)
23. Giménez, J., Màrquez, L.: Technical Manual v1.3. Universitat Politecnica de Catalunya, Barcelona (2006)
24. Kudo, T.: CRF ++: Yet Another CRF Toolkit (2013). http://crfpp.sourceforge.net/ptojrcts/crfpp/. Accessed 10 July 2015
25. Patel, K.: A Sambalpuri Phonetic Reader. Menaka Prakashani, Sambalpur (undated)
26. Masica, C.P.: The Indo-Aryan Languages. Cambridge University Press, Cambridge (1993)

Cross-Linguistic Projection for French-Vietnamese Named Entity Translation

Ngoc Tan Le(✉) and Fatiha Sadat

Universite du Quebec a Montreal,
201 avenue President Kennedy, Montreal, QC H2X 3Y7, Canada
le.ngoc_tan@courrier.uqam.ca, sadat.fatiha@uqam.ca

Abstract. High-quality translation is time-consuming and an expensive process. Named Entity (NE) Translation, including proper names, remains a very important task for multilingual natural language processing. Most of the gold standard corpora are available for English but not for under-resourced languages such as Vietnamese. In Asian languages, this task is remained problematic. This paper focuses on a named entity translation approach by cross-linguistic projection for French-Vietnamese, a poor-resourced pair of languages. We incrementally apply a cross-projection method using a small parallel annotated corpora, such as the surface string matching measures according to probabilistic string edit distance similarity and an additional score of syllable consistence feature between the source term and the target term by a syllabification process. Evaluations on French-Vietnamese pair show a good accuracy with BLEU gain more than 4 points when translating bilingual named entities pairs.

Keywords: Named entity · Bilingual corpus · Cross-projection
Named entity translation · French-Vietnamese

1 Introduction

Due to the multiple meanings of words, expressions and also the metaphors, machine translation systems do not always offer correct translations for given contexts. They may reflect a common name written with upper case as if it is a proper name and vice versa, they translate a name having a signification in a bilingual dictionary as in the case of a common name. Named entity translation, in particular, allows to correctly identify through the proper names languages such as people's names, names of organizations and also the names of the locations. Named entity translation is a very important topic, *i.e.* statistical machine translation (SMT), cross language information retrieval, information extraction and questions & answers because named entities - particularly persons names, location names and organizations names – contain an essential meaning in natural language processing.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 407–419, 2018.
https://doi.org/10.1007/978-3-319-93782-3_29

We can see the following examples of correct and incorrect named entity translations from French to Vietnamese. Here we use Google Translate to illustrate.

(1.1) (fr) Hier soir, j'ai mangé avec Monsieur Michel Poulet. / (en) Last night, I ate with Mr. Michel Poulet.
(1.2) (vi) Đêm qua tôi đã ăn thịt gà với Michel. [*incorrect translation by Google Translate, consulted 26 June 2017*]
Literally : Hier soir, j'ai mangé du poulet avec Michel (fr) / Last night, I ate chicken with Michel. (en)
(1.3) (vi) Tối qua, tôi đã ăn với ông Michel Poulet. [*correct translation*]

In the first example above, the named entity that designates a name of a person was incorrectly translated, "Monsieur Michel Poulet" in the sentence (1.1) by "thịt gà với Michel" (*literally: chicken with Michel*) in the sentence (1.2) instead of "với ông Michel" in the sentence (1.3).

(2.1) (fr) Ma famille voyage dans le delta du fleuve Rouge. / (en) My family travels in the delta of the Red river.
(2.2) (vi) Gia đình tôi đi ở đồng bằng sông Đỏ. [*literally incorrect translation*]
(2.3) (vi) Gia đình tôi du lịch ở đồng bằng sông Hồng. [*correct translation by Google Translate, consulted 26 June 2017*]

In the second example, the named entity that designates a name of a place has been incorrectly translated "fleuve Rouge" in the sentence (2.1) by "sông Đỏ" in the sentence (2.2) instead of "sông Hồng" in the sentence (2.3). The translation error here is about the synonymy between two words "Đỏ" and "Hồng" which mean the same signification.

(3.1) (fr) Il est en train de lire L'Observateur. / (en) He is reading L'Observateur.
(3.2) (vi) Hiện anh đang đọc The Observer. [*incorrect translation by Google Translate, consulted 26 June 2017*]
(3.3) (vi) Anh ta đang đọc L'Observateur. [*correct translation*]

We can see in the third example above, sometimes a discrimination failure between proper name and common name. In this case, the named entity that designates a name of an organization "L'Observateur" (3.1) was incorrectly translated from French to Vietnamese. His translation was borrowed from english by "The Observer" (3.2) instead of keeping it intact name like "The Observer" (3.3).

One possible solution is to build a bilingual named entity dictionary. A named entity dictionary or a list of NE pairs is a base for rule-based translation and statistical transliteration method. However, this approach needs firstly a large scale of bilingual corpus with named entity annotated. Manual annotation of bilingual corpora is time-consuming and an expensive process. Unfortunately, there is very few or no researches regarding the translation of named entities for the French-Vietnamese languages pair.

In this paper we propose an iterative approach to named entity translation by cross-linguistic projection for French-Vietnamese, a poor-resourced languages pair.

The structure of this paper is as follows. Section 2 describes the related works about different methods of named entity translation. Section 3 presents our approach about named entity translation for French-Vietnamese. Sections 4 and 5 discusses the experiment setting and results. Finally, conclusion will be given in the last Sect. 6.

2 Related Work

The task of named entity translation is to translate a named entity, including proper names, temporal and numerical expression from the source language into the target language. Many researchers have tried to solve the named entity translation by several approaches. There are rule-based method, statistical method and web mining method [19].

The rule-based method uses linguistic rules to transliterate and translate named entities. Wan et al. [35] applied this method to transliterate english country names in chinese names. The statistical method uses a large scale annotated bilingual corpus as training data. It includes statistical transliteration method, comparable or parallel bilingual corpora-based method. The dominant technique is to create a NE alignment and a bilingual NE lexicon. Huang [10] combined both semantic translation and phonetic transliteration for english-chinese NE translation. Hassan et al. [9], Kim et al. [13] and Sellami et al. [30] proposed, by applying the comparable bilingual corpora-based method, the NE translation based on their context similarity, transliteration similarity and phrase-based translation similarity.

The web mining method uses a large scale of web corpora. Huang et al. [11], Jiang et al. [12], Yang et al. [36] and Mingming et al. [21] presented a new framework to names translations using web mining method. A given term is submitted to a search engine what extracts the list of translation candidates. This candidate translation list is ranked based on the surface patterns, co-occurrence feature and transliteration feature.

It is challenging to translate named entities across languages with different alphabets and pronunciations such as Arabic, Russian, Korean, Japanese, Thai, Chinese, etc. There are several studies on named entity translation for various language pairs such as English-Spanish, French-English, English-Arabic, English-Japanese, etc. However, we find very few publications on named entity translation for the French-Vietnamese, except [28] which has investigated linguistically how to deal with proper names from English and French into vietnamese. The machine translation systems face many problems with this pair such as the characteristics of the named entities and the inconsistency of their handwriting and transcription/transliteration in Vietnamese.

Since 2009, various transliteration systems have been proposed during the Named Entities Workshop evaluation campaigns[1] [5]. These campaigns consist of transliterating from English into languages with a wide variety of writing systems, including Hindi, Tamil, Russian, Kannada, Chinese, Korean, Thai and Japanese. We can see that the romanization of non-Latin writing systems remains a complex computational task that is highly dependent on a language.

Through this workshop, much progress has been made in the methodologies with an emergence of different approaches, such as grapheme in phoneme [8,22], based on statistics like automatic translation [18,24] as well as neural networks [6,7,32,33]. The variety of writing systems adds another important challenge in the extraction of named entities and automatic transliteration. All these difficulties are aggravated by the lack of bilingual dictionaries of proper names, ambiguities of transcription as well as orthographic variation in a language. Lo et al. [20] used a semi-supervised transliteration model built on a seed corpus mined from the standard parallel training data, in order to improve the Russian-English machine translation system for WMT 2016. Ngo et al. [22] proposed a statistical model for a language pair with English-Vietnamese language, with a phonological constraint on the syllables. Their system has achieved better performance than the base system, based on rules, with a 70% reduction in error rates. Cao et al. [3] also applied the statistical-based approach as automatic translation in the transliteration task for a language pair with little English-Vietnamese language, with a performance of 63% of BLEU [27].

In this work, we incrementally apply a cross-projection method using a small parallel annotated corpora, such as the surface string matching measures according to probabilistic string edit distance similarity and an additional score of syllable consistence feature between the source term and the target term by a syllabification process. In this paper, we will present a new framework that deals with the NE translation for French-Vietnamese.

3 Our Framework

We present an approach of named entity translation for French-Vietnamese. Here we discuss a morphosyntactic appearance between the named entities in the source language and the target language. Our approach relies on two following hypotheses:

Hypothesis 1. We suppose that a named entity in source language and its translated NE in target language have the same category such as person name, location name or organization name.

Hypothesis 2. Considering that person and location names are often phonetically translated and their written forms are similar to their pronunciations, we can add an additional syllables consistency feature. It means a syllabification

[1] http://workshop.colips.org/news2016/.

measure comparing the number of syllables in the word blocks or group of words between the source and the target.

Therefore, the proposed approach is composed of three main steps:

- Step 1: Extracting a list of French named entities candidates and a list of Vietnamese proper names candidates from the French-Vietnamese bilingual corpus based on sentences level.
- Step 2: Filtering French named entities candidates translated into Vietnamese by a statistical model.
- Step 3: Scoring a similarity by calculating pairs of bilingual candidates translated by the statistical model. This similarity is based on the Levenshtein string edit distance equation (1) as follows:

$$similarity(S_i, T_j) = 1 - \frac{edit\_distance(S_i, T_j)}{maxlength(|S_i|, |T_j|)} \tag{1}$$

  The edit distance function of Levenshtein or minimum edit distance is widely used as measurement between two strings. It returns the minimum weight series of edit operations that transforms source word S_i into target word T_j related to the insertion, the deletion or the substitution.

We divide the bilingual corpus into two monolingual corpus as shown in Fig. 1.

1. Firstly, the French-Vietnamese bilingual corpus is aligned at sentences level. Then, by applying a French named entity recognition module, we get a list of French named entity candidates $list\_FR\_NE$. For Vietnamese corpus, we apply a POS (*part-of-speech, grammatical categories*) annotation module and we also get a list of Vietnamese proper names candidates $list\_ref\_VI\_PN$.
2. Then a statistical model is applied in order to translate the list of French named entity candidates $list\_FR\_NE$ into Vietnamese. We obtain a list of translated named entity candidates $list\_translated\_VI\_NE$.
3. We calculate the similarity scores between this translated named entity candidates $list\_translated\_VI\_NE$ as the source S_i and the list of Vietnamese proper names candidates $list\_ref\_VI\_PN$ as a target T_j. If the value of a pair of scores of bilingual candidates is greater than the threshold value, this pair is chosen and stored in a list of bilingual NE pairs candidates.
   After analyzing the possible errors of named entities translation, we retrain this post-edited list in the statistical model to observe the possible impacts on the quality of named entities translation.

Statistical Machine Translation
Statistical Translation model has been proven in MT applications since 1990. Inspired by the noisy channel model of Claude Shannon [31], Brown and his team in IBM company proposed the first Statistical Translation System (SMT) [1,2]. The authors supposed that any sentence in a source language S, can be translated is another sentence in target language T. Thus for any pair of sentences (s, t),

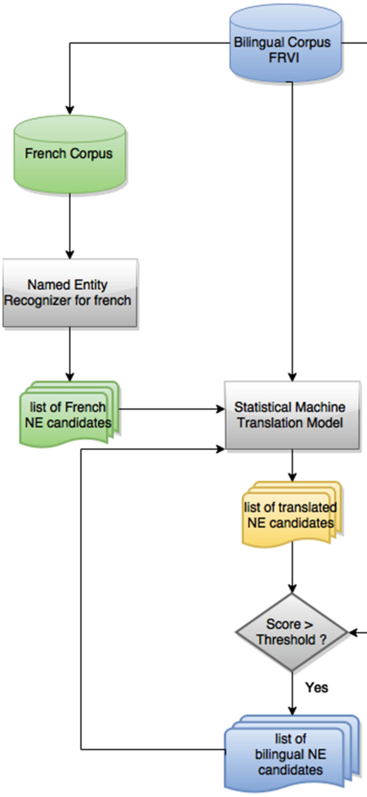

Bilingual Corpus FRVI
French Corpus
Vietnamese Corpus
Named Entity Recognizer for french
Part-of-Speech Tagger for vietnamese
list of French NE candidates
Statistical Machine Translation Model
list of vietnamese proper names candidates
list of translated NE candidates
Score > Threshold ?
Yes
list of bilingual NE candidates

Fig. 1. Framework of named entity translation system for French-Vietnamese

they assign a probability of translation $p(t|s)$. This probability is interpreted as a feature that the machine translation will perform the translation hypothesis $\hat{t}$ in the target language T given a sentence s in the source language S. The problem of SMT aims to maximize the probabilities between the translation model and the language model and choosing the best translation hypothesis $\hat{t}$. Hence, applying Bayes' theorem, mathematically, the problem is described as below:

$$p(t|s) = \frac{p(s|t)\ p(t)}{p(s)} \tag{2}$$

Because the probability $p(s)$ is independant to the source sentence t, so the Eq. (2) will be simplified :

$$\hat{t} = argmax_{t \in t^*}\ p(s|t)\ p(t) \tag{3}$$

As described in the Eq. (3), the architecture of a SMT-based system is composed of two importants components: the **translation model (TM)** $p(s|t)$ and the **language model (LM)** $p(t)$. In fact, the translation model (TM) contains the list of phrases translations, after the training phase by using a large parallel bilingual corpus and the language model is built from a monolingual corpus in the target language. The translation model gives the best translation hypothesis according to the source input text while the language model ensures that this hypothesis is syntactically correct for the target text regardless of the source input text.

Moreover, in the architecture of a SMT-based system, there is also a third important component, the **decoder**. It aims to search and to find out the best translation hypothesis $\hat{t}$ among all possibilities proposed by the system. In a non exhaustive list, there are many decoders in the litterature such as Pharaoh[2] [14], Portage [29] and Moses[3] [16]. Figure 2 describes the general architecture of the state of the art of SMT-based system.

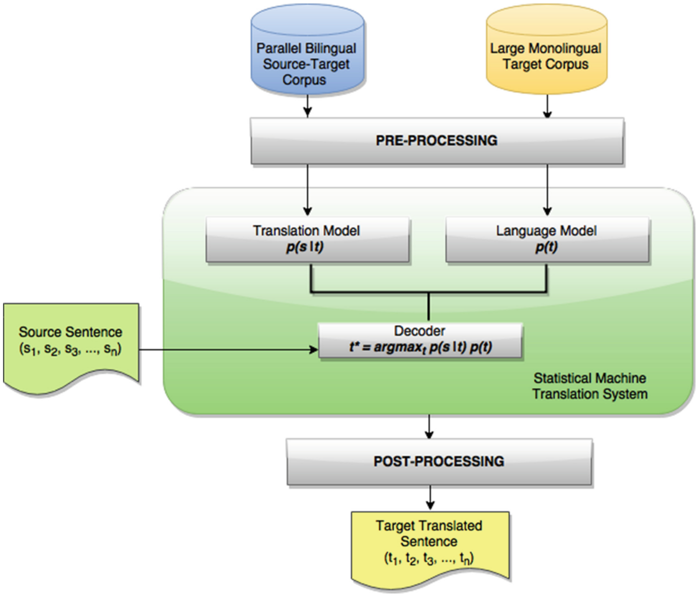


Fig. 2. General architecture of the state of the art of statistical machine translation based system

Actually, many SMT-based systems perform translation models either based on words or based on phrases. The first approach is known as the Word-Based Statistical Machine Translation (WBSMT) in which the system is based on an automatic word-to-word alignment [26]. The second approach is known as the Phrases-Based Statistical Machine Translation (PBSMT), in which the system considers the alignment unit as a contiguous sequences of words [17].

[2] http://www.isi.edu/licensed-sw/pharaoh/.
[3] http://www.statmt.org/moses.

4 Experimentation

4.1 Data Preparation

The baseline French-Vietnamese bilingual corpus is collected with 14,963 sentence pairs from the multilingual web pages for news and 5,284 sentence pairs from the Tam Dao conference, in 2009[4]. This is an international economic conference organized annually in Vietnam, where there are many named entities. Due to time constraints, we only extracted one 2009 version. In perspective we can extract more. We have 1536 named entity including 687 persons names, 713 locations names and 135 organizations names. In our experiments, we use a small test corpus with 1,060 pairs of French-Vietnamese pairs phrases.

4.2 Configuration Settings

The word segmentation is required for Vietnamese corpus. The VCL_WS tool of the VCL group [34] is used for this step. And the Vietnamese corpus is also annotated by VCL_POS tagger using the maximum entropy approach [23]. The automatic annotation system for French is a tool for recognizing named entities developed by [25]. We implement Moses[5]. This is a statistical machine translation system [15]. Moses offers all the tools needed to build a statistical model. A decoder allows to generate translation assumptions of a source text. It consists of a mechanism capable of effecting the maximization of the Eq. (3) in an acceptable time.

We performed three following experiments as follows:

- Exp1 (*baseline*) : In this experiment, we use a training data of 14k and a test data of 1,060 bilingual sentence pairs.
- Exp2 : In this experiment, we use a training data of 14k and a test data of 359 named entities extracted from 1,060 bilingual sentence pairs.
- Exp3 : In this experiment, we use a training data of 14k combining with 5k from Tam Dao 2009 conference and a test data of 359 named entities extracted from 1,060 pairs of bilingual phrases.

5 Evaluation

5.1 Results

To evaluate the named entity translation accuracy, we use the metrics such as BLEU (Bilingual Evaluation Understudy) [27] and NIST (National Institute of Standards and Technology) [4] on the test corpus. Table 1 and Fig. 3 show the results of the experiments.

Given the above three experiments, we note that the performance is clearly improving in regard to the named entities translation with 21.68% (Exp1) →

[4] http://www.tamdaoconf.com.

[5] http://www.statmt.org/moses_steps.html.

Table 1. Results of the experimentations of named entity translation for bilingual French-Vietnamese

System	BLEU (%)	NIST
Exp1(*baseline*)	21.68	5.3977
Exp2	24.05 (*+2.37*)	**3.4556**
Exp3	**26.66** (*+4.98*)	3.5600

24.05% (Exp2, *+2.37*) → 26.66% (Exp3, *+2.61*). We note that there are still unknown words in the first experiment. Hence the metric BLEU was only 21.68% and the metric NIST is 5.3977 due to noise of the unknown words. Then we decide to do the second and third experimentations only with named entities extracted from the test set (including 359 named entities). Thus, the BLEU and NIST metrics show a performance improvement with a gain of more than 2 points of BLEU and NIST over the baseline.

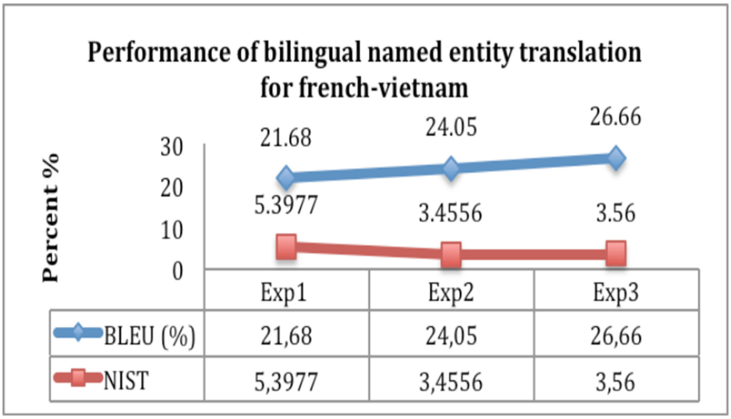


Fig. 3. Performance of bilingual named entity translation for French-Vietnamese

Moreover, we find that the named entities such as person and location names have a great similarity between French and Vietnamese. The Vietnamese tends to resemble phonetically forms of foreign names with similar pronunciation. We can measure the similarity by counting the number of syllables or the syllables consistency between source term and target term. Some examples are illustrated in Table 2.

5.2 Errors Analysis

In addition, we find that there are shortcomings and errors in the bilingual named entity translation. A major drawback of a system based on the statistical

Table 2. Some examples of the number of syllables similarity between source term and target term

French	→	Vietnamese	Comment
Phillippines	→	Phi-lip-pin	*location name with 3 syllables*
Vietnam	→	Việt Nam	*location name with 2 syllables*
Singapour	→	Sing-ga-po	*location name with 3 syllables*
Pharaon	→	Pha-ra-ông	*person name with 3 syllables*
Joseph	→	Giô-sép	*person name with 2 syllables*

machine translation model involves the amount of training data. The training data should be as large as possible in order to cover all linguistic varieties of translations.

Translation errors are categorized into three criteria: lexicon, syntax and transcription/transliteration.

1. Lexical errors concern the lack of words. The system deals with the out-of-vocabulary words, the missing words or the incorrect words.
   For example:
   (fr) fleuve Rouge → (vi) sông Đỏ #instead of « sông Hồng »
   (fr) Mékong → (vi) Mékong #instead of « Cửu Long »
   (fr) Long Biên → (vi) Dài Biên #instead of «*LongBiên*»
2. Syntax errors concern the mistranslation of names structures or word order in noun phrases.
   For example:
   (fr) Asie du Sud-Est → (vi) Á của Đông Nam #instead of « Đông_Nam_Á »
   (fr) Afrique → (vi) Phi Châu #instead of « Châu Phi »
   (fr) Asie orientale → (vi) Châu Đông_Á #instead of « Á Đông »
3. Transcription/transliteration errors relate to proper names which are poorly transcribed or transliterated by the machine translation system due to the influence of english words during manual translation.
   For example:
   (fr) Singapore → (vi) Singapore #instead of « Singapour → Sing-ga-po »
   (fr) Algeria → (vi) An-giê-ri #instead of « Algérie »
   (fr) Californie → (vi) California #instead of « Ca-li-pho-ni-a »
   (fr) Malaysie → (vi) Malaysia #instead of « Malaisie → Ma-lai-xi-a » or « Mã Lai »

6 Conclusion

In this paper we presented an approach on named entity translation by cross-linguistic projection for French-Vietnamese, a poor-resourced pair of languages. We applied a cross-projection method using a small parallel annotated corpora, and calculating the surface string matching measures according to probabilistic string edit distance similarity and an additional score of syllable consistence feature between the source term and the target term by a syllabification process. Evaluations on French-Vietnamese pair of languages show a good accuracy with BLEU gain more than 4 points when translating bilingual named entities pairs. This resulted in a small bilingual annotated corpus in a significant improvement into named entity translation.

In perspective, we will focus on collecting a large scale bilingual corpus. We will deal with different kinds of error in NE translation and propose to introduce other features (*i.e. features based on transliteration*) in order to improve the quality of the extracted NE translation. The framework can be naturally extended to other bilingual comparable corpora when the training data are available.

References

1. Brown, P.F., Cocke, J., Pietra, S.A.D., Pietra, V.J.D., Jelinek, F., Lafferty, J.D., Mercer, R.L., Roossin, P.S.: A statistical approach to machine translation. Comput. Linguist. **16**(2), 79–85 (1990)
2. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2), 263–311 (1993)
3. Cao, N.X., Pham, N.M., Vu, Q.H.: Comparative analysis of transliteration techniques based on statistical machine translation and joint-sequence model. In: Proceedings of the 2010 Symposium on Information and Communication Technology, pp. 59–63. Association for Computing Machinery (2010)
4. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)
5. Duan, X., Banchs, R.E., Zhang, M., Li, H., Kumaran, A.: Report of news 2016 machine transliteration shared task. In: ACL 2016, pp. 58–72 (2016)
6. Finch, A., Liu, L., Wang, X., Sumita, E.: Neural network transduction models in transliteration generation. In: Proceedings of NEWS 2015 The Fifth Named Entities Workshop, p. 61 (2015)
7. Finch, A., Liu, L., Wang, X., Sumita, E.: Target-bidirectional neural models for machine transliteration. In: ACL 2016, pp. 78–82 (2016)
8. Finch, A., Sumita, E.: Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model. In: Proceedings of the 2010 Named Entities Workshop, pp. 48–52. Association for Computational Linguistics (2010)
9. Hassan, A., Fahmy, H., Hassan, H.: Improving named entity translation by exploiting comparable and parallel corpora. In: AMML 2007 (2007)
10. Huang, F.: Improved named entity translation and bilingual named entity extraction. In: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, p. 253. IEEE Computer Society (2002)
11. Huang, F., Zhang, Y., Vogel, S.: Mining key phrase translations from web corpora. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 483–490. Association for Computational Linguistics (2005)

12. Jiang, L., Zhou, M., Chien, L.F., Niu, C.: Named entity translation with web mining and transliteration. In: IJCAI 2007, pp. 1629–1634 (2007)
13. Kim, J., Jiang, L., Hwang, S.w., Song, Y.I., Zhou, M.: Mining entity translations from comparable corpora: a holistic graph mapping approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1295–1304. ACM (2011)
14. Koehn, P.: Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In: Frederking, R.E., Taylor, K.B. (eds.) AMTA 2004. LNCS (LNAI), vol. 3265, pp. 115–124. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30194-3_13
15. Koehn, P.: Statistical Machine Translation. Cambridge University Press, New York (2009)
16. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pp. 177–180. Association for Computational Linguistics (2007)
17. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 48–54. Association for Computational Linguistics (2003)
18. Laurent, A., Deléglise, P., Meignier, S., Spécinov-Trélazé, F.: Grapheme to phoneme conversion using an SMT system. In: Proceedings of INTERSPEECH, pp. 708–711. ISCA (2009)
19. Liu, Y.: The technical analyses of named entity translation. In: International Symposium on Computers & Informatics, pp. 2028–2037. ISCI (2015)
20. Lo, C.k., Cherry, C., Foster, G., Stewart, D., Islam, R., Kazantseva, A., Kuhn, R.: NRC Russian-English machine translation system for WMT 2016. In: Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics (2016)
21. Mingming, Z., Yu, H., Jianmin, Y.: Research on name entity translation based on transliteration and web. In: Proceedings of the 6th National Conference on Information Retrieval, pp. 357–366 (2010)
22. Ngo, H.G., Chen, N.F., Nguyen, B.M., Ma, B., Li, H.: Phonology-augmented statistical transliteration for low-resource languages. In: Interspeech, pp. 3670–3674 (2015)
23. Nguyen, K.A., Dinh, D.: Tích hợp thông tin từ loại vào hệ dịch máy thống kê. In: National Conference, Cần Thơ, pp. 150–157 (2011)
24. Nicolai, G., Hauer, B., Salameh, M., St Arnaud, A., Xu, Y., Yao, L., Kondrak, G.: Multiple system combination for transliteration. In: Proceedings of NEWS 2015 The Fifth Named Entities Workshop, pp. 72–79 (2015)
25. Nouvel, D., Antoine, J.-Y., Friburger, N.: Pattern mining for named entity recognition. In: Vetulani, Z., Mariani, J. (eds.) LTC 2011. LNCS (LNAI), vol. 8387, pp. 226–237. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-08958-4_19
26. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Comput. Linguist. **29**(1), 19–51 (2003)
27. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics (2002)

28. Phan, T.T.T.: Machine translation of proper names from english and french into vietnamese: an error analysis and some proposed solutions. Ph.D. thesis. Université de Franche-Comté (2014)
29. Sadat, F., Johnson, H., Agbago, A., Foster, G., Kuhn, R., Martin, J., Tikuisis, A.: Portage: A phrase-based machine translation system. In: Proceedings of the ACL Workshop on Building and Using Parallel Texts, pp. 129–132. Association for Computational Linguistics (2005)
30. Sellami, R., Sadat, F., Belguith, L.H.: Mining named entity translation from non parallel corpora. In: FLAIRS Conference, pp. 219–224 (2014)
31. Shannon, C.E., Weaver, W.: The Mathematical Theory of Information. University of Illinois Press, Urbana (1949)
32. Shao, Y., Nivre, J.: Applying neural networks to English-Chinese named entity transliteration. In: Sixth Named Entity Workshop, Joint With 54th ACL (2016)
33. Thu, Y.K., Pa, W.P., Sagisaka, Y., Iwahashi, N.: Comparison of grapheme-to-phoneme conversion methods on a myanmar pronunciation dictionary. In: Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing 2016, pp. 11–22 (2016)
34. Vu, D.H.: Phân đoạn từ tiếng việt ngữ dụng. Master thesis (2011)
35. Wan, S., Verspoor, C.M.: Automatic english-chinese name transliteration for development of multilingual resources. In: Proceedings of the 17th International Conference on Computational linguistics-Volume 2, pp. 1352–1356. Association for Computational Linguistics (1998)
36. Yang, F., Zhao, J., Liu, K.: A chinese-english organization name translation system using heuristic web mining and asymmetric alignment. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1, pp. 387–395. Association for Computational Linguistics (2009)

National Language Technologies Portals for LRLs: A Case Study

Delyth Prys(✉) and Dewi Bryn Jones(✉)

Bangor University, Bangor, Wales
{d.prys,d.b.jones}@bangor.ac.uk

Abstract. The new Welsh National Language Technologies Portal is an extensive resource for researchers, developers in the ICT and digital media spheres, open source enthusiasts and code clubs who may have limited understanding of language technologies but which nevertheless have a need for incorporating linguistic data and capabilities into their own projects, products, processes and services in order to better serve their wider LRL community. It includes a repository of free, simple and accessible resources with documentation, tutorials, example code and projects. This paper describes the rationale and process of building the Portal, new novel resource dissemination mechanisms employed, such as online APIs and Docker, as well as the lessons learnt and applicability to other similar linguistic situations and communities.

Keywords: LRL · Repositories · Language resources · Welsh

1 Introduction

One of the many problems facing developers and users of language technology for less-resourced languages (LRLs) is the fragmented nature of resources and attendant information for the languages they work with. In the international field of HLT and computational linguistics, large international catalogues and repositories of language data and processing tools such as ELDA and META-SHARE are now making it much easier to distribute and promote LRs at a global level. However, these are targeted at academics and professionals who are well-versed in HLT, who often work with multiple languages, and don't need a lot of hand-holding to integrate these tools and resources into their finished products or environments.

Such repositories fail to address the needs of many less-resourced languages that historically have been of little interest to the mainstream academic and commercial markets. Researchers and developers struggle to create and disseminate tools and resources to be taken up by local companies and activists who cater for those language communities.

In such scenarios, many of those who service the LRL community are small-scale developers of apps, local media companies, webmasters, open source enthusiasts, coding clubs, and public and third sector bodies. Other potential users are global companies who integrate a wide range of languages in their multilingual offerings, provided it is not too costly to do so. This wide constituency of developers may vary in its understanding of ICT and digital media in general, but the incorporation of the

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 420–429, 2018.
https://doi.org/10.1007/978-3-319-93782-3_30

linguistic element into their products is an additional barrier to the uptake of the necessary tools and resources. In addition to the tools and resources themselves, this constituency therefore has additional requirements for help and support.

In order to help both the local constituency of developers and the LRL community that they serve, it can be beneficial to create local repositories specifically targeting them. Such repositories need to include additional tutorials and case studies on resource use, in addition to the metadata and documentation required of all LR repositories. Resources also need to be accessible, user friendly and simple to use, enabling incorporation into other products with the minimum of additional coding and development.

This paper describes one effort to create a National Language Technologies Portal for one such less-resourced language, namely Welsh, the steps taken to choose the tools and resources for inclusion, the environment created to disseminate language technology code, data and capabilities and the additional tutorials and documentation that accompanied them.

The paper concludes with considerations for the further elaboration of the National Language Technologies Portal, together with lessons that have been learnt that may be applicable to other such LRL communities.

2 Choosing Tools and Resources for Inclusion

The META-NET series of White Papers on Europe's languages in the Digital Age (Rehm and Uszkoreit 2012) provides an overview of present provision of LRs for many of Europe's languages. Welsh was a late addition to this series (Evas 2014), and was scored as the weakest of the languages covered in the series in terms of its LRs. Despite this, a significant number of LRs were in existence, many of them not cited in the META-NET volume on Welsh. This in itself was cause for concern, showing low levels of awareness, even within the Welsh national community.

Attempts had been made in the early 2000s to create a Welsh or a Celtic BLARK, emulating the Dutch exercise in creating a comprehensive taxonomy of tools and resources, and elaborating the interdependencies between types of data and applications and modules that use them (Prys 2006). However, government support was not forthcoming for such a strategic project, possibly highlighting the difference between official European languages and minority languages, even though Welsh was gaining some additional recognition at the time. A strategy document *Information Technology and the Welsh Language* was published (Welsh Language Board 2006), but this was a catalogue of various IT considerations, including terminology standardization, localization, bilingual web design, the need for corpora and speech technology, training and second language learners needs, rather than an action plan to provide a comprehensive infrastructure.

Consequently it was left to individual research teams, companies and volunteers to develop any relevant tools and resources, being guided by their own interests and the availability of funding. Many of these LRs were the result of R&D projects at the Language Technologies Unit at Bangor University, a self-financed research centre where the roadmap towards comprehensive coverage of Welsh language LTs had to be balanced by the need to attract funding for its projects. Various tools and resources were created at the LTU for internal project use, for example an online version of a Welsh spelling and grammar checker had been developed in order to gather a corpus of errors and of corrected texts (Cysill Ar-lein 2009). Some of these tools and resources had been made publicly available at various locations, but others had been originally intended purely for internal use and needed further refinement and packaging if they were to be made suitable for public distribution.

The Welsh National Language Technologies Portal was therefore conceived as a project to improve the visibility and dissemination of at least some of the available resources. Funding of £50,000 was obtained through a small Welsh Government grant, with the work having to be completed within one financial year. It was recognised that not all resources could be prepared, packaged and published during the project's initial allotted timeframe of 8 months. However, in order for such a National Language Technologies Portal to be of any worth from the outset, a core set of the right resources needed to be chosen for inclusion.

The project's first task was to create an inventory of available resources and tools, together with notes on their state of readiness, documentation, the opportunities on offer, insights and analytics gathered from any logs or public feedback, licensing issues, target audience and suggested priority (Prys et al. 2014). This differs substantially from a full taxonomy of resources as developed by Binnenpoorte et al. (2002) for HLT resources for Dutch, but was an attempt at least to find some thematic groupings from the legacy tools and resources. It was recognised that these were a rather ad hoc legacy collection, including, for example, lexical resources such as dictionaries and applications, and also tools that use these lexical resources, such as a Welsh language detector.

The inventory included over 30 candidates which were grouped into the following categories:

1. Language Proofing
2. Machine Translation
3. Speech Technology
4. Corpus Harvesting and Processing
5. Lexical Resources
6. Welsh Place-names

Each candidate resource was assigned a perceived level of importance within its category to its target audience. The following table examples one candidate resource:

Based on the content and suggestions contained in the report, as well on its own information, the government funders prioritised seven resources and tools for initial inclusion:

1. **Vocab** – a Welsh/English Dictionary Website Plugin that integrates the extensive *Geiriadur Bangor* Welsh-English dictionary into any website
2. **Cysill Ar-lein** – an on-line Welsh language spelling and grammar checker
3. **Welsh/English equivalents word list** – a simple list of corresponding words and phrases in Welsh and English

4. **Welsh Social Media Corpus** – Welsh language texts collected from Twitter and Facebook
5. **Welsh/English Alignment Tool** – based on the popular HunAlign
6. **Welsh Statistical Machine Translation** – data and scripts for facilitating machine translation with the Moses SMT system between Welsh and English
7. **Welsh Language Synthetic Speech** – Welsh text to speech voice based on the Festival Speech Synthesizer system. The project team undertook that, if time and resources allowed, the following resources should also be added to the initial resources:
8. **Welsh Language Part-of-Speech Tagger** – the PoS tagger used within the Welsh language grammar checker.
9. **Language Detection for Welsh** – as described in Table 1.

Table 1. Example candidate resource entry

5. Lexical resources		
Id	Name	Importance
5.7	Welsh Language Detection	2
Description	A resource that can detect Welsh language texts	
State of readiness	Developed so far for internal use only. Needs further evaluation and possibly training	
Documentation	Needs to be documented and packaged with example usages	
Ideas	There are many e.g. separate texts that are linguistically mixed (English and Welsh)	
Comments	The detector is based on an open source Java library. It has been trained with texts from the Language Technologies Unit's various corpus resources. It would be worthwhile to contribute the models back to the original project: http://code.google.com/p/language-detection	
Licensing	Apache	
Target audience	Software developers, (especially those which process bilingual texts and resources)	
Priority	This is important. This is a missing resource that other developers often contact us about	

This left out many other useful tools and resources, such as a parallel Welsh/English corpus of the Welsh Assembly's Record of Proceedings, despite pointing out its importance for the development of machine translation and other applications. It is however hoped that this and other resources may be added at some future time.

The project proceeded with identifying the common dissemination mechanisms between various resources which would influence National Language Technologies Portal organisation and construction.

3 Portal Construction and Organisation

The resources chosen for initial inclusion differed greatly in form, nature and in their mechanisms for dissemination. Some resources existed only in the form of large data files, such as the social media corpora, whilst others, such as the spelling and grammar checker were functionalities desirable for integration into as many software products and websites as possible. It was realised therefore that no single platform implementation would be able to fulfil all the requirements of a Welsh National Language Technologies Portal. The portal would be constructed as an initial website, serving as a superficial layer of information with resources hosted in reality in a federation of bespoke sub-websites and third party web services that would each best serve each resources' means for dissemination.

3.1 Main Website

WordPress, which is a free and open content management system, suitable for use as a web publishing platform, was chosen as the basis for the development of the initial website. The website would need to be bilingual and with the addition of multilingual plugins, WordPress is especially convenient for the production of bilingual sites, due to the ease of toggling backwards and forwards between two languages when developing new content, and of translating each page. Simple usage of WordPress menus was used to organise and present the resources according to a simplified breakdown of language technologies themes (see Table 2).

Table 2. Initial menu structure of the Welsh National Language Technologies Portal

Translation	Speech	Cloud	Corpora
Aligning Localisation Machine translation	Text to speech	API services Widgets	Corpus of Welsh tweets Corpus of Welsh Facebook texts

Each choice in the menu would take the user to a simple page containing a description of the theme as well as of each resource contained within, with links to access the true location of the resources. In some cases, links were also included to pages that provide support on how to obtain the resources via the various dissemination mechanisms employed.

3.2 Cloud API Services

The National Portal needed to disseminate resources that were functionalities desirable for integration into as many software products and websites as possible. A bespoke website was constructed for providing access to such language technology capabilities via simple online APIs.

As long as the computer or device has a connection to the internet, these capabilities, such as Welsh language text to speech, would be very easy to integrate into any software capable of communicating via simple HTTP requests and response. Thus complexities and barriers associated with downloading, porting and building code for a given deployment environment are eliminated. In addition, the API approach permits free access to capabilities provided by commercial products not amenable to free distribution of code and data but which could still be incorporated legitimately into other software products and projects, in particular open source.

The initial offering of online API Services provided by the portal were: Cysill Arlein Welsh language spelling and grammar checker, Welsh language Parts of Speech Tagger, Vocab, Welsh language Text to Speech Engine, Language Detect and Welsh language lemmatizer.

In order to use the API services, the user must register at the bespoke API Services website after which the user can choose an API, agree to the terms and conditions before receiving a 128-bit integer Globally Unique Identifier (GUID) API key for their use of that API. Terms and conditions can vary between API services. All however aim to prevent misuse and protect the service for all users. Thus users agree to accept a rate limit on the number of requests per hour for every API Service. Terms and conditions for API Services provided by commercial products, request that no attempts be made at reverse engineering.

3.3 GitHub

Some resources exist as code. The National Portal project would also create a significant amount of new code for its tutorial and example projects. The most obvious and most attractive location for hosting open source code freely is GitHub. GitHub is well known and popular amongst developers, where you can discover, use and contribute to millions of projects using a collaborative development workflow.

All code based resources provided by the National Portal would exist within a number of repositories within a specially created GitHub organisation, separate from any other ongoing projects developed on GitHub by the researchers. Repositories were used also to contain documentation, tutorials and example projects for each API service. In hosting resources on GitHub it will be possible for users to contribute additions and enhancements back into the resources.

In addition, the National Portal's main website would provide instructions as to how resources located on GitHub could be downloaded even if Git is not installed on the user's PC.

3.4 Docker

Some of the resources on offer as code and script for download and local execution are of a very complicated and sophisticated nature, none more so than the Moses-SMT machine translation system. It is not a trivial task for a developer, let alone a MT practitioner to master its complicated and lengthy build process and subsequent loading of translation models and or training.

Recent technological developments in software deployment provide opportunities for significantly simplifying the downloading, installation and execution of complex applications to the National Portal's target audience. Docker is the leading solution and service to date in this space. Docker is an open platform for building, shipping and running distributed applications. Entire applications can be packaged as images and executed in containers without worrying about inconsistencies between various development and production environments and without locking into any platform or language.

In addition the Docker Hub Registry provides a free-to-use registry of pre-prepared images submitted by users or officially by popular open source projects such as Ubuntu, Postgresql and WordPress. Thus, in addition to hosting all Welsh/English Moses-SMT resources on GitHub, an image of a Moses-SMT server packaged with scripts that facilitate fetching pre-trained Welsh/English translation models, and execution were submitted to the Docker Hub Registry. This enables a user in two very easy Docker commands to have his/her own local Welsh/English Moses-SMT machine translation environment and server.

4 Beta-Testing, Engaging and Building a Developers Community

During the period of building the National Portal and packaging the tools and resources, a number of methods were used to reach out to potential users in order to raise awareness of the forthcoming repository and also to find beta-testers who would be prepared to work with us to make sure it was fit for purpose and easy to use.

Since the National Portal's website had been implemented on WordPress, its blogging features were put to use along with our twitter feed to reach out and market resources as they became available to as wide as possible a target audience.

Fortunately also the project was able to reach out and obtain feedback from companies that had approached its members in the past enquiring whether such resources as they needed for their products existed for Welsh. The *Hacio'r Iaith* event (an annual Welsh-medium unconference for hackers, enthusiasts, developers and members of the media industry) was also used as an ideal venue to reach a worthwhile target audience.

Amongst the most interesting groups who engaged with beta-testing the resources were a class of 9 and 10 year old school children and their teacher from a small rural school in North West Wales. The school had received a number of Raspberry Pi computers through the generosity of the Raspberry Pi Foundation and Google and wanted to utilise the new Welsh language resources for activities that would teach them not only coding but strengthen their Welsh language written and oral skills. Coding club resources in Welsh are very scarce and thus an existing introduction to coding lesson plans by Raspberry Pi Foundation based on the Turing Test was translated. The lesson plans were adapted to incorporate the Welsh language Text to Speech API service, as well as expanded to suggest using some other language technologies such as the Cysill Ar-lein spelling and grammar checker, parts of speech tagger and language detection, in order to give their Turing test code the appearance of Welsh language capabilities.

Outreach activities climaxed in a one day conference to launch the National Portal and give it further publicity. This was an opportunity to bring together academics, businesses, enthusiasts and other stakeholders, and thereby break down some of the barriers that traditionally exist between them. Presentations included one on the use a freelance developer had made of the Twitter corpus, and another on analysing Welsh tweets, together with the work the schoolchildren in programming their Raspberry Pis to speak Welsh. The conference was covered by the Welsh language television and radio news with the children giving excellent answers in their interviews.

Speakers however were not confined to local participants, with international presenters from Ireland, the Basque Country and South Africa also taking part. This brought the much needed perspective of other less-resourced languages into play, hopefully helping to build up a network of like-minded people, and laying the foundation for future joint projects and collaborations.

5 Further Work and Conclusions

Although this initial project was only 8 months in duration, the National Portal itself was designed for long term sustainability. The repository will continue to be developed and used as and when other projects produce relevant tools and resources.

Already a further project on combining recent developments in Welsh language speech recognition, along with machine translation, text to speech and commercial search APIs by Google and Microsoft called *Seilwaith Cyfathrebu Cymraeg* (Welsh Communications Infrastructure), funded by the Welsh Government and S4C (the Welsh Broadcasting Authority), and a follow-on project called *Macsen* (A Welsh Personal Digital Assistant), funded by the Welsh Government, have made additional resources freely available through the National Portal and its dissemination mechanisms on GitHub, as API Services and further images in the Docker Hub Registry.

New additions include speech recognition kits for Welsh, a forced aligner and source code for the Welsh digital personal assistant. The corpora included in the Portal continue to grow dynamically, with the Twitter corpus for example having increased from 2 million to 7 million words. A link to the new Welsh META-SHARE node has been added, with Git Large File Storage (LFS) having recently been introduced to help facilitate the versioning of larger files.

The popularity of the API Services approach has been encouraging with app developers, webmasters and others normally not able to use language technologies already integrating or at least exploring enhancing their products and digital offerings for Welsh. The Vocab widget and API service (Jones et al. 2016) has been added to two popular national Welsh language news websites, with others significant services in the pipeline. The *Cysill Ar-lein* Welsh language spelling and grammar checker API is being considered for inclusion into apps for Welsh learners as well as scripts for automating proof reading Welsh language articles in Wikipedia. In the meantime other developers are asking for further language technology capabilities to be added.

Also encouraging is that the Moses-SMT image in the Docker's Hub Registry has been pulled to date 278 times, gaining four star ratings with positive comments, and appears to have created forked versions for other language communities.

The approach of National Portal web sites for other types of Welsh language resources has been already well-established, with a Welsh National Terminology Portal having been set up in 2010, and a Welsh National Corpora Portal following in 2011. Building on the National Portal 'brand' has helped in raising awareness of the new offering of a Welsh National Language Technologies Portal, as well as reflect accurately its national character and provenance of funding.

Discussions of language technology matters, the danger of 'digital extinction' for small languages such as Welsh, and especially recent developments in speech recognition technology, have been stimulated by the high visibility of the Language Technology Portal and the increased activity that surrounds it. This has led to increased engagement with the media, including coverage on radio and television. In June 2017, a debate on Technology and the Welsh Language: Risk or Opportunity? was held in the Welsh Assembly (The Record of Proceedings 28/06/17), linking the national debate to issues of both economic and linguistic revitalization.

A common danger with tools and resources for LRLs is lack of quality control, as developers are desperate for anything they can use. This can be overcome not only by the formal evaluation of such tools and resources, but also by encouraging continuous feedback and dialogue with communities of users. The former can prove challenging for LRLs, who may lack the capacity for developing resources, let alone evaluating them, and where methods devised to evaluate LRs for well-resourced languages may not always be suitable. On the other hand, relationships with communities of users, especially if activists and enthusiasts are included, can be closer and better in LRL environments, and feedback and engagement with these communities needs to be actively encouraged, rather than merely aping what works for WRLs.

Acknowledgements. The Welsh National Language Technologies Portal project reported on in this paper was made possible with the financial support of the Welsh Government, through its Technology and Digital Media in the Welsh Language Fund. The authors would also like to thank the contributors from various SMEs, hackers and communities of users that assisted the project team.

References

Binnenpoorte, D., De Friend, F., Sturm, J., Daelemans, W., Cucchiarini, C.: A field survey for establishing priorities in the development of HLT resources for Dutch. In: Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, Las Palmas de Gran Canaria (2002)

Cysill Ar-lein: Bangor University, Bangor (2009). http://www.cysgliad.com/cysill/arlein/. Accessed 17 Sept 2015

Evas, J.: The Welsh Language in the Digital Age. Metanet White Paper Series. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-45372-4

Jones, D.B., Prys G., Prys D.: Vocab: a dictionary plugin for web sites. In: Actes de la conference conjointe JEP-TALN-RECITAL, vol. 6. CLTW, Paris (2016)

Prys, D., Jones D.B., Cooper, S., Robertson, P.: Adnoddau Technolegau Iaith i'w Cynnwys mewn Porth Adnoddau Cenedlaethol [Language Technology Resources to be Included in a National Terminology Portal]. Unpublished Report to the Welsh Government (2014)

Prys, D.: The BLARK Matrix and its relation to the language resources situation for the Celtic languages in Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa (2006)

Rehm, G., Uszkoreit, H.: The Danish Language in the Digital Age. White Paper Series. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30627-3

The Record of Proceedings of the National Assembly of Wales for 28/06/2017 Beginning at 18.33 hours, Cardiff Wales. http://www.cynulliad.cymru/en/bus-home/pages/rop.aspx?meetingid=4301&assembly=5&c=Record%20of%20Proceedings&startDt=28/06/2017&endDt=28/07/2017#484320. Accessed 4 July 2017

Welsh Language Board: Information Technology and the Welsh Language/Technoleg Gwybodaeth a'r Iaith Gymraeg (2006). http://orca.cf.ac.uk/43799/1/3964.pdf. Accessed 27 Oct 2015

Welsh National Corpora Portal (2011). http://corpws.cymru/?lang=en. Accessed 9 Sept 2015

Welsh National Language Technologies Portal (2015). http://techiaith.cymru/?lang=en. Accessed 9 Sept 2015

Welsh National Terminology Portal (2010). http://termau.cymru/?lang=en. Accessed 9 Sept 2015

Raspberry Pi Learning Resources: The Turing Test. https://www.raspberrypi.org/learning/turing-test-lessons/. Accessed 9 Sept 2015

Challenges for and Perspectives on the Malagasy Language in the Digital Age

Joro Ny Aina Ranaivoarison(✉)

Centre Interdisciplinaire de Recherche Appliquée au Malgache,
Ankatso University, Antananarivo, Madagascar
jororanaivo@yahoo.fr

Abstract. This paper is a revised and extended version of my LTC 2015 contribution [26]. It shows the state of Malagasy language at the present time and deals with challenges and perspectives. To enter in the digital age, languages must provide resources and tools. The creation of useful tools such as spell checkers or machine translation systems would introduce Malagasy into the era of new technology. It encourages users to use the language more. However, it is usually the work of specialists of Natural Language Processing (NLP). For Malagasy, an agglutinative language, the collaboration between specialists of NLP and linguists is required. This paper surveys tools and resources that have been constructed for Malagasy, and, among others, a project [24, 27] based on the DELA framework [14] to construct NLP dictionaries of Malagasy by using conventional dictionaries and converting them into a structured, but readable and manually updatable, resource usable by Unitex [18]. We report on the ongoing construction of NLP dictionaries of verbs, nouns from verbs, grammatical words with the same DELA methodology and we discuss the dictionaries of simple words and multi-word units.

Keywords: Less-resourced language · Malagasy · NLP dictionary

1 Introduction

The Malagasy language is the national language of Madagascar (about 400 km East of Africa) whose official languages are French and Malagasy. It is spoken by 23 millions of people. In the 19th century, missionaries from England and France came to Madagascar and studied this language. Conventional dictionaries such as Freeman [8], Weber [29], Malzac [15], and Malagasy grammar books in English, French or Malagasy, such as Griffiths [9], Cousins [4], Andrianony [1], have been published.

Since Rajaona [22], Malagasy linguists have produced more scientific, richly documented studies of the Malagasy language. In the 1990s, starting with Rabenilaina [20] and followed by Ralalaoherivony [23], the Malagasy language entered an era of formalized study of language with the introduction of the concept of NLP dictionary.

For Malagasy language, some tools exist but they are not usable by the general public; digital media and devices are used by journalists but information written in Malagasy language is not explored efficiently.

Z. Vetulani et al. (Eds.): LTC 2015, LNAI 10930, pp. 430–444, 2018.
https://doi.org/10.1007/978-3-319-93782-3_31

In this paper, my goal and motivation are presented in Sect. 2, existing tools for Malagasy language are presented in Sect. 3, and existing resources in Sect. 4. Section 5 reports on methods of building NLP dictionaries. As for Sect. 6, it deals with NLP dictionaries of Malagasy (verbs, nouns from verbs, grammatical words). In Sect. 7, an evaluation of these dictionaries is reported. In Sect. 8, discussion about dictionaries of simple words and multi-word units is offered. Finally, in Sect. 9, as a conclusion, global perspectives about Malagasy language in the digital age are examined.

2 Motivations and Goals

Resources and tools are required for the processing of Malagasy. Krauwer [13] cites some resources and tools as BLARK, and Enguehard and Mangeot [7] cite others: adapted keyboards, spell checker, speech synthesis, machine translation, etc. Among all of them, we chose to construct a monolingual dictionary because it is likely to contribute to practically all other objectives mentioned by these authors.

Our objective is to construct a manually-updatable dictionary. We construct a dictionary of roots and since it we perform morphological analysis of a word or generation of inflected forms. The availability of a dictionary of roots facilitates the implementation of a morphological analyzer, a spell checker, and indirectly the annotation of corpora, i.e. several BLARK items. Constructing a dictionary for an agglutinative language is a major scientific challenge, and the first milestone in order to build tools and more advanced natural language applications.

3 Previous Work on Tools

Some researchers and programmers working on Malagasy language have already constructed or developed tools. A program of concordance exists for example with Pr. Jean-Yves Morin at the University of Canada but features of this program cannot be clarified (information about the product is not available). Researchers and developers at the Institut Supérieur Polytechnique de Madagascar (ISPM) led by Pr. Julien Raboanary realized for example a program of machine translation and spell checker. A demonstration of machine translation has been organized by the ISPM but no literature about the system has been found. The spell checker is developed in JAVA and it proposes corrections for errors found in the text [21]. It has no grammar checking module. Dalrymple *et al.* [5], in the framework of the Parallel Grammar Project (PARGRAM), have built a morphological analyzer for Malagasy language at Xerox. A program of recognition of named entities has been constructed too with Poibeau *et al.* [19]. These tools are not widely used and the dictionaries are not available for research.

4 Previous Work on Resources

Diwersy [6] collected a corpus of modern Malagasy newspaper texts, which is freely available under LGPL-LR license.

As for NLP dictionaries, those constructed for existing tools are not available for research. The only available one is Ranaivoarison *et al.*'s [24, 27] NLP dictionary of Malagasy simple verbs and nouns from verbs. They are a structured, but readable and updatable lexical resource based on peer-reviewed morphosyntactic information. It was inspired by Berlocher *et al.*'s [2] efficient, large-coverage morphological analyzer for Korean, which is an agglutinative language like Malagasy. Berlocher *et al.*'s analyzer too is based on readable and updatable resources: an NLP dictionary of stems, finite-state transducers of suffixes and finite-state transducers of generation of allomorphs.

This method was adapted to Malagasy, and Unitex [18] performs morphological analysis of Malagasy verbs and nouns from verbs with the resource.

5 Methodology

Our model is based on the model of handcrafted transducers and DELA dictionaries [2, 10, 14]. An alternative model of this last might be the two-level morphology model [11], which has been used to deal with agglutinative languages such as Finnish [12], Turkish [17] and Malagasy [5]. However, the resources of a two-level morphology system are less readable and less easy to update because most rules are very abstract and a priori applicable to any word. Updating one rule may affect a priori any lexical item, endangering the performance of the language system. Since adding new entries in the system may involve changes in the rules, the processing of pre-existing entries can become incorrect. In contrast, experiments with Korean dictionary DECO [16] showed that electronic dictionaries are easy to maintain and update then the resources of a two-level morphology. With DELA dictionaries, every word is explicitly assigned a specific rule, i.e. a transducer. As a result, updating a transducer in the system may only affect the corresponding words. This makes the system more reliable and the construction of the resources can be cumulative. As the construction of electronic dictionary involved collecting data, defining stem and affix classes, constructing transducers of variation of allomorphs and combination of morphemes, populating the dictionary, in this section, we outline successively these processes in this order.

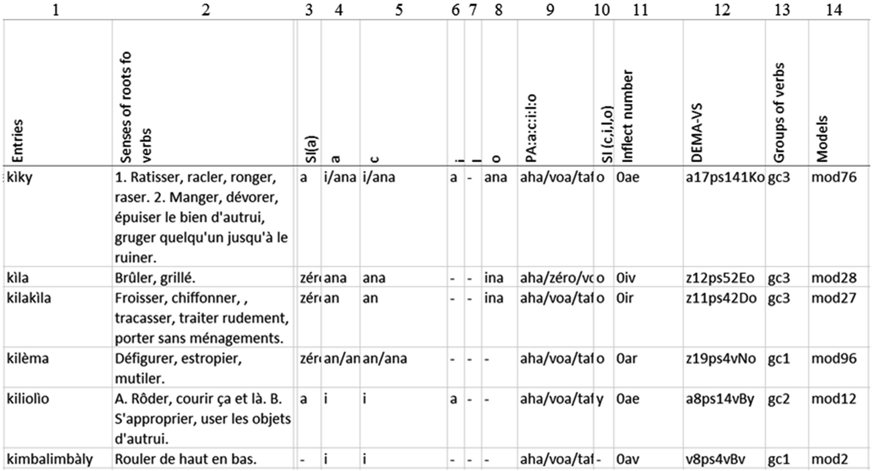
5.1 Collecting Empirical Data

Due to the large number of lexical entries, the construction of an electronic dictionary required a huge amount of linguistic information, which is organized into a table identifying the numerous morphemes which combine with roots. Table 1 shows for example the table of verbs.

The 3rd until the 10th column give morphemes attached to each root and the 11th column shows how verbs vary in contact of morphemes. The last 3 columns are information we use to populate the dictionary of verbs.

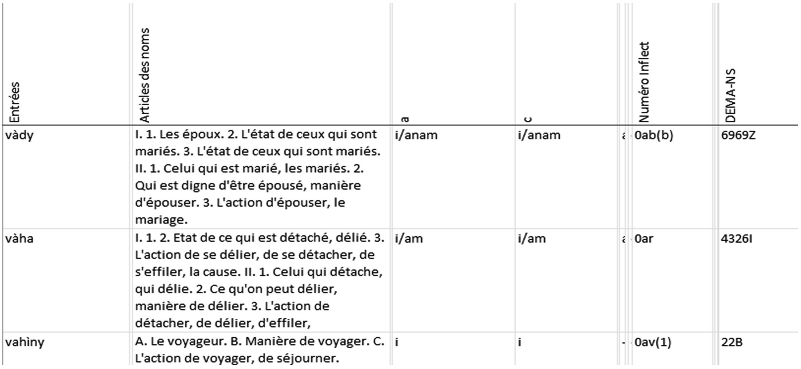
Table 2 (below) shows the table of nouns from verbs.

Table 1. Table of verbs.



1	2	3	4	5	6	7	8	9	10	11	12	13	14
Entries	Senses of roots fo verbs	SI(a)	a	c	i	l	o	PA:a:c:i:l:o	SI (c,i,l,o)	Inflect number	DEMA-VS	Groups of verbs	Models
kìky	1. Ratisser, racler, ronger, raser. 2. Manger, dévorer, épuiser le bien d'autrui, gruger quelqu'un jusqu'à le ruiner.	a	i/ana	i/ana	a	-	ana	aha/voa/taf	o	0ae	a17ps141Ko	gc3	mod76
kìla	Brûler, grillé.	zér	ana	ana	-	-	ina	aha/zéro/vc	o	0iv	z12ps52Eo	gc3	mod28
kilakìla	Froisser, chiffonner, , tracasser, traiter rudement, porter sans ménagements.	zér	an	an	-	-	ina	aha/voa/taf	o	0ir	z11ps42Do	gc3	mod27
kilèma	Défigurer, estropier, mutiler.	zér	an/an	an/ana	-	-	-	aha/voa/taf	o	0ar	z19ps4vNo	gc1	mod96
kiliolìo	A. Rôder, courir ça et là. B. S'approprier, user les objets d'autrui.	a	i	i	a	-	-	aha/voa/taf	y	0ae	a8ps14vBy	gc2	mod12
kimbalimbàly	Rouler de haut en bas.	-	i	i	-	-	-	aha/voa/taf	-	0av	v8ps4vBv	gc1	mod2

Table 2. Table of nouns from verbs



Entrées	Articles des noms	a	c		Numéro Inflect	DEMA-NS
vàdy	I. 1. Les époux. 2. L'état de ceux qui sont mariés. 3. L'état de ceux qui sont mariés. II. 1. Celui qui est marié, les mariés. 2. Qui est digne d'être épousé, manière d'épouser. 3. L'action d'épouser, le mariage.	i/anam	i/anam	a	0ab(b)	6969Z
vàha	I. 1. 2. Etat de ce qui est détaché, délié. 3. L'action de se délier, de se détacher, de s'effiler, la cause. II. 1. Celui qui détache, qui délie. 2. Ce qu'on peut délier, manière de délier. 3. L'action de détacher, de délier, d'effiler,	i/am	i/am	a	0ar	4326I
vahìny	A. Le voyageur. B. Manière de voyager. C. L'action de voyager, de séjourner.	i	i	-	0av(1)	22B

The 3rd and the 4th columns from the first column give morphemes attached to each root to form nouns from verbs. The 5th column shows how roots vary in contact of morphemes and the last column shows the combination of morphemes to form nouns. These last 2 columns are information use to populate the dictionary of nouns from verbs.

These data classify verbs and nouns from verbs according to 2 criteria: firstly, morphemes attached to each root and their combinations with each other; and, secondly, the way roots change in contact of morphemes.

5.2 Defining Affix and Stem Classes

Combinations of morphemes define classes named “affix classes” and types of variations of form of roots define “stem classes”. The table was used to identify these two classes.

Firstly, stem classes give variations of roots when they are adjacent to morphemes. When nouns are from verbs, the same stem classes used for verbs are used for nouns from verbs.

Secondly, affix classes provide the different morphemes attached to the roots. However, affix classes for verbs are different from affix classes for nouns from verbs.

These 2 classifications (stem and affix classes) cross-classify and made up a complex of linguistic data. The first classification enables Unitex to recognize morphemes combined with roots and the second allows it to generate variants of roots. This functionality uses the transducers encoded for each class.

5.3 Building Transducers of Generation of Allomorphs and Combinations of Morphemes

Firstly, transducers of generation of allomorphs concern variation of roots. Secondly, transducers of combination of morphemes concern morphemes attached to each root and their combinations with each other. These 2 types of transducers are encoded graphically with the graph editor of Unitex. They are presented below in this order.

Transducers of Generation of Allomorphs. Afinite-state transducer of generation of allomorphs is associated to each stem class and specifies the formal variants of roots found in inflected forms of verbs or nouns from verbs. Figure 1 is an example of a transducer of generation of allomorphs.

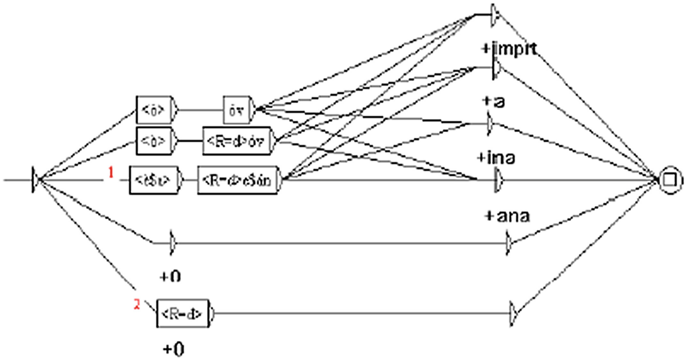


Fig. 1. Transducers of generation allomorphs V0ibe

Paths 1 and 2 allow for example generation of *dehán* and *dèha* in inflected forms:

- of verbs such as *mandeha* “to walk” (verb in present time, indicative mode, active-stative voice) *mandehana* “walk” (imperative form), *andehanana* “the circumstance of walking” (verb in present time, indicative mode, circumstancial voice), etc.
- of nouns from verbs such as *mpandeha* “traveler” (agent/profession noun), *fandeha* “way of walking” (manner noun), *fandehanana* “action of walking” (action noun).

Paths indicate respectively affixes with which they combine. The path 1 provides for example the form *dehán*; and associates to it, a coded property + ana indicating that it combines with the affix -*ana* and is found for example in the forms such as *fandehanana* “action of walking” for nouns and *andehanana* “circumstance of walking” for verbs. The other properties (+ imprt, + a, + ina) for this path are used for verbal forms [25:227]. The box with +0 indicates that after the morphological variant there is no more suffix as in path 2. Indeed, after the morphological variant *deha*, there is no more suffix, as in nominal forms (*mpandeha* and *fandeha*) or verbal forms (*mandeha).*

Transducers of Combination of Morphemes. Afinite-state transducer of combination of morphemes is associated to each affix class and specifies which morphemes combine with the roots. However, transducers of combination of morphemes for verbs are different from transducers of combinations of morphemes for nouns from verbs.

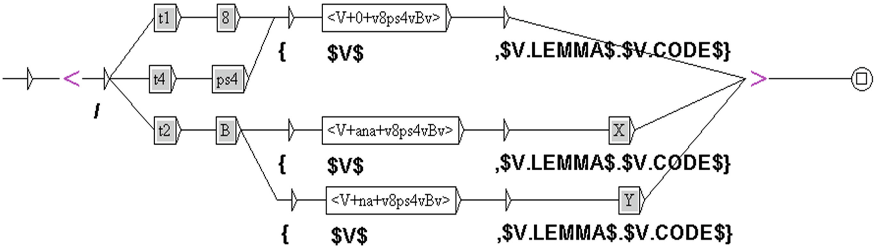

<
/
t1
8
t4
ps4
t2
B
<V+0+v8ps4vBv>
{ V ,$V.LEMMA$.$V.CODE$}
<V+ana+v8ps4vBv>
X
{ V ,$V.LEMMA$.$V.CODE$}
<V+na+v8ps4vBv>
Y
{ V ,$V.LEMMA$.$V.CODE$}
>

Fig. 2. Transducers of combination of morphemes for verbs v8ps4vBv

Figure 2 (below) shows for example a type of transducers of combination of morphemes for verbs.

This graph concerns for example *zozozòzo* “to buzz” and all verbs which have the same combination of morphemes as it (such as *àmbatra* “to speak in an inarticulate manner”). The graph allows morphological analysis of all form of verbs from it such as:

- active-stative voice *nizozozozo* (past), *mizozozozo* (present), *hizozozozo* (future)
- circumstancial voice *nizozozozoana* (past), *izozozozoana* (present), *hizozozozoana* (future)
- “accomplished” forms like *voazozozozo* (present), *ho voazozozozo* (future)

As for Fig. 3 (below), it shows for example a type of transducers of combination of morphemes for nouns from verbs.

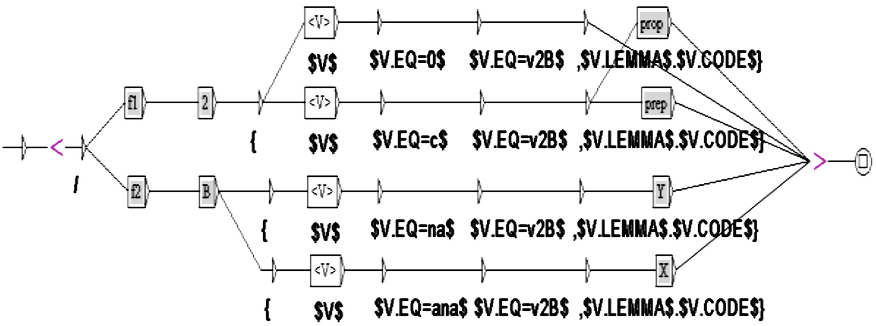


Fig. 3. Transducers of combination of morphemes for nouns from verbs v2B

This graph concerns also *zozozòzo* "to buzz". It allows morphological analysis of forms such as *fizozozozo* "way of buzzing, of rattling, of whistling" (noun of manner), *fizozozozoana* "action of buzzing, rustling, hissing" (noun of action).

5.4 Populating Dictionary

Roots as lexical items are inserted in a dictionary with their stem and affix classes which contain lexical information. These 2 pieces of information are represented by 2 codes in the dictionary entries. These 2 codes identify the transducers of Sect. 5.3.

Unitex imposes conventions of codification of entries, e.g. the 2 codes must be separated by the sign "+" without any space. However, the dictionary can be edited manually for extension and corrections.

When entries and lexical information are inserted in the dictionary, this last is inflected with a command of Unitex. Then, Unitex produces a dictionary of variants of roots. This dictionary is then compressed into a binary file with which Unitex can automatically recognize inflected forms of verbs or nouns from verbs in a text.

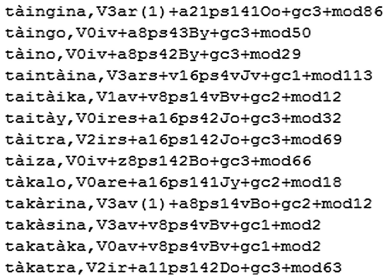
6 Results

The parts of speech that we have been working on are verbs, nouns from verbs and grammatical words. Information about them is being organized and inserted in a manually-updatable dictionary. As our work is based on these three parts of speech, three kinds of NLP dictionaries are obtained, NLP dictionary of verbs, nouns from verbs and grammatical words. To present our result, the dictionary of verbs is reviewed firstly. Secondly, the dictionary of nouns from verbs is presented. Thirdly, the dictionary of grammatical words is seen.

6.1 NLP Dictionary of Verbs

Two types of dictionary of verbs are being distinguished. On the one hand, there is the dictionary of roots of verbs and on the other hand, there is the dictionary of variants of roots of verbs. Below, these two types of dictionaries are presented successively.

Dictionary of Roots of Verbs. Entries of the dictionary of roots of verbs are roots of verbs themselves. The dictionary has 3200 entries and its structure is "Entry,VStemClass+AffixClass+GroupOfConjugation+ModelOfConjugation". Below is an excerpt from this dictionary (Fig. 4).

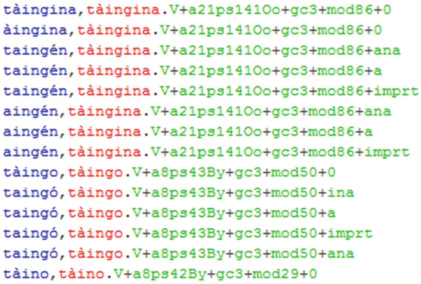


```
tàingina,V3ar(1)+a21ps141Oo+gc3+mod86
tàingo,V0iv+a8ps43By+gc3+mod50
tàino,V0iv+a8ps42By+gc3+mod29
taintàina,V3ars+v16ps4vJv+gc1+mod113
taitàika,V1av+v8ps14vBv+gc2+mod12
taitày,V0ires+a16ps42Jo+gc3+mod32
tàitra,V2irs+a16ps142Jo+gc3+mod69
tàiza,V0iv+z8ps142Bo+gc3+mod66
tàkalo,V0are+a16ps141Jy+gc2+mod18
takàrina,V3av(1)+a8ps14vBo+gc2+mod12
takàsina,V3av+v8ps4vBv+gc1+mod2
takatàka,V0av+v8ps4vBv+gc1+mod2
tàkatra,V2ir+a11ps142Do+gc3+mod63
```

Fig. 4. NLP dictionary of roots of verbs

This dictionary is already completed. A part of this dictionary is freely available on http://igm.univ-mlv.fr/∼unitex. It uses about 232 types of transducers of generation of allomorphs and about 500 types of transducers of combination of morphemes. The two transducers operate directly on the dictionary. On applying the command "Inflect…" of the menu "DELA" of Unitex, it produces automatically the dictionary of variants of roots.

Dictionary of Variants of Roots of Verbs. The dictionary of variants of roots is the electronic dictionary of verbal roots that appear in the inflected forms of verbs. It is built automatically from the dictionary of roots by a program of generation of forms of Unitex. Below, a sample of entries of this dictionary is given (Fig. 5).



```
tàingina,tàingina.V+a21ps141Oo+gc3+mod86+0
àingina,tàingina.V+a21ps141Oo+gc3+mod86+0
taingén,tàingina.V+a21ps141Oo+gc3+mod86+ana
taingén,tàingina.V+a21ps141Oo+gc3+mod86+a
taingén,tàingina.V+a21ps141Oo+gc3+mod86+imprt
aingén,tàingina.V+a21ps141Oo+gc3+mod86+ana
aingén,tàingina.V+a21ps141Oo+gc3+mod86+a
aingén,tàingina.V+a21ps141Oo+gc3+mod86+imprt
tàingo,tàingo.V+a8ps43By+gc3+mod50+0
taingó,tàingo.V+a8ps43By+gc3+mod50+ina
taingó,tàingo.V+a8ps43By+gc3+mod50+a
taingó,tàingo.V+a8ps43By+gc3+mod50+imprt
taingó,tàingo.V+a8ps43By+gc3+mod50+ana
tàino,tàino.V+a8ps42By+gc3+mod29+0
```

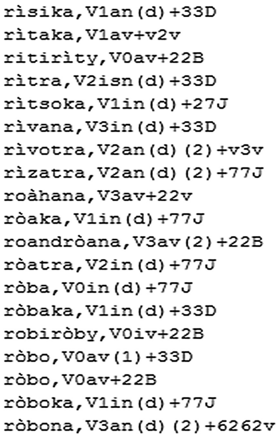
Fig. 5. NLP dictionary of variants of roots of verbs

These dictionaries both allow morphological analysis of verbs found in a text and generation of conjugated forms of verbs.

6.2 NLP Dictionary of Nouns from Verbs

As for verbs, two types of dictionary of nouns from verbs are distinguished. The first concerns the dictionary of roots of nouns from verbs and the second is the dictionary of variants of roots of nouns from verbs. Both types of dictionary are presented below in this order.

Dictionary of Roots of Nouns from Verbs. As the dictionary is a dictionary of nouns from verbs, entries of the dictionary are roots of verbs themselves but lexical information attached to each entry is different from verbs themselves on one point: affix classes for verbs are composed of six fields while for nouns from verbs are composed of three fields. An excerpt of this dictionary is presented below (Fig. 6).



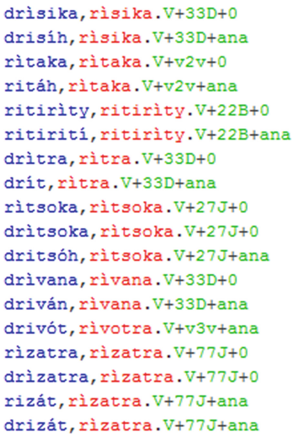
```
rìsika,V1an(d)+33D
rìtaka,V1av+v2v
ritirìty,V0av+22B
rìtra,V2isn(d)+33D
rìtsoka,V1in(d)+27J
rìvana,V3in(d)+33D
rìvotra,V2an(d)(2)+v3v
rìzatra,V2an(d)(2)+77J
roàhana,V3av+22v
ròaka,V1in(d)+77J
roandròana,V3av(2)+22B
ròatra,V2in(d)+77J
ròba,V0in(d)+77J
ròbaka,V1in(d)+33D
robiròby,V0iv+22B
ròbo,V0av(1)+33D
ròbo,V0av+22B
ròboka,V1in(d)+77J
ròbona,V3an(d)(2)+6262v
```

Fig. 6. NLP dictionary of roots of verbs forming nouns

"Entry,VStemClass+AffixClass" is the general structure of each line of the dictionary. It has about 2100 entries at the present time and is almost completed (65% completed). As for verbs, this dictionary uses 232 types of stem classes (see. Transducers of generation of allomorphs). As for the affix classes (see. Transducers of combination of morphemes), it uses about one hundred types. On inflecting the dictionary, the program of generation of terms of Unitex produces a dictionary called "dictionary of variants of roots of nouns from verbs".

Dictionary of Variants of Roots of Nouns from Verbs. The dictionary of variants of roots of nouns from verbs is the electronic dictionary of verbal roots that appear in the inflected forms of nouns from verbs. It is built automatically from the dictionary of roots. Below is a sample of entries of this dictionary (Fig. 7).

This dictionary allows morphological analysis and generation of nouns of agent/profession, manner and action such as *mpandrisika* "the one who encourages", *fandrisika* "way to encourage", *fandrisihana* "encouragement" respectively.

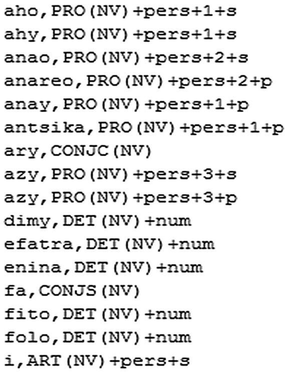


```
drìsika,rìsika.V+33D+0
drisíh,rìsika.V+33D+ana
rìtaka,rìtaka.V+v2v+0
ritáh,rìtaka.V+v2v+ana
ritirìty,ritirìty.V+22B+0
ritirití,ritirìty.V+22B+ana
drìtra,rìtra.V+33D+0
drít,rìtra.V+33D+ana
rìtsoka,rìtsoka.V+27J+0
drìtsoka,rìtsoka.V+27J+0
dritsóh,rìtsoka.V+27J+ana
drìvana,rìvana.V+33D+0
driván,rìvana.V+33D+ana
drivót,rìvotra.V+v3v+ana
rìzatra,rìzatra.V+77J+0
drìzatra,rìzatra.V+77J+0
rizát,rìzatra.V+77J+ana
drizát,rìzatra.V+77J+ana
```

Fig. 7. NLP dictionary of variants of roots of verbs forming nouns

6.3 NLP Dictionary of Grammatical Words

Grammatical words comprise pronouns, conjunctions, prepositions, interjections. Generally, they are almost invariable words except for some personal pronouns which are contracted with verbs [24]. This dictionary is 50% completed. An excerpt of this dictionary is shown below (Fig. 8).



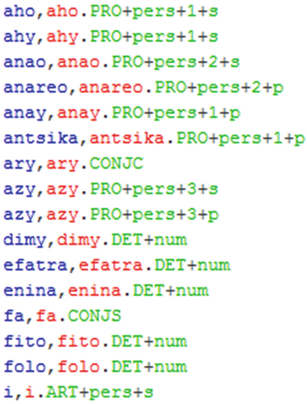
```
aho,PRO(NV)+pers+1+s
ahy,PRO(NV)+pers+1+s
anao,PRO(NV)+pers+2+s
anareo,PRO(NV)+pers+2+p
anay,PRO(NV)+pers+1+p
antsika,PRO(NV)+pers+1+p
ary,CONJC(NV)
azy,PRO(NV)+pers+3+s
azy,PRO(NV)+pers+3+p
dimy,DET(NV)+num
efatra,DET(NV)+num
enina,DET(NV)+num
fa,CONJS(NV)
fito,DET(NV)+num
folo,DET(NV)+num
i,ART(NV)+pers+s
```

Fig. 8. NLP dictionary of grammatical words

A transducer for invariable words allows to Unitex to recognize those grammatical words in a text. Below, a figure of this transducer is shown (Fig. 9).

Fig. 9. Transducer for invariable words

On using this transducer (Fig. 9) and the command "Inflect" of Unitex, Unitex produces another dictionary which will be utilized for automatic recognition of grammatical words. An excerpt of this dictionary is displayed (Fig. 10).



```
aho,aho.PRO+pers+1+s
ahy,ahy.PRO+pers+1+s
anao,anao.PRO+pers+2+s
anareo,anareo.PRO+pers+2+p
anay,anay.PRO+pers+1+p
antsika,antsika.PRO+pers+1+p
ary,ary.CONJC
azy,azy.PRO+pers+3+s
azy,azy.PRO+pers+3+p
dimy,dimy.DET+num
efatra,efatra.DET+num
enina,enina.DET+num
fa,fa.CONJS
fito,fito.DET+num
folo,folo.DET+num
i,i.ART+pers+s
```

Fig. 10. NLP dictionary of inflected forms of grammatical words

7 Evaluation

With these resources cited in Sect. 6, Unitex performs morphological analysis of inflected forms of verbs, nouns from verbs and grammatical words.

For the experimentation, we use a corpus from Diwersy [6] which has not been used to construct dictionaries. We utilize the first part of the journalistic corpus of contemporary Malagasy (cmjc1) [25] which contains words to test the coverage of dictionaries. In cmjc1, the first one hundred sentences are taken for the evaluation.

For verbs, all verbs identified in the text are all recognized and analyzed by Unitex. This dictionary is complete.

For nouns from verbs, the text is 25% covered, just some nouns are identified and analyzed, most of them are not recognized and analyzed because their roots are not already inserted in the dictionary. These roots are roots beginning with R, S, T, V. In fact, all nouns from verbs whose roots begin with Z, A until P are recognized in the portion of text but most of nouns in the text have a root beginning with R, S, T, V. This dictionary is under construction and it is about to be finished.

As for grammatical words, part of speech of personal pronouns, some locative pronouns, conjunctions, are identified and analyzed but most of them are not already inserted in the dictionary. This dictionary is under construction.

8 Discussion

On the model of Nam's [16] Korean NLP dictionary, the structure of a project of a Malagasy morphological NLP dictionary (Fig. 11) is foreseen.

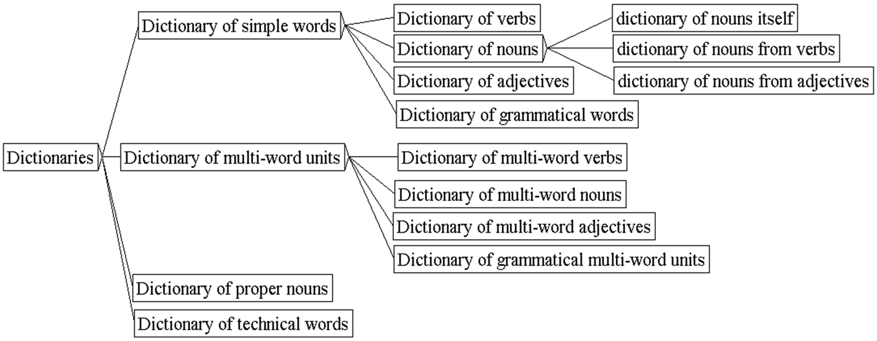

Dictionaries
Dictionary of simple words
Dictionary of verbs
Dictionary of nouns
Dictionary of adjectives
Dictionary of grammatical words
dictionary of nouns itself
dictionary of nouns from verbs
dictionary of nouns from adjectives
Dictionary of multi-word units
Dictionary of multi-word verbs
Dictionary of multi-word nouns
Dictionary of multi-word adjectives
Dictionary of grammatical multi-word units
Dictionary of proper nouns
Dictionary of technical words

Fig. 11. Structure of Malagasy NLP Dictionary

At this present time, three kinds of NLP dictionary are built: NLP dictionary of verbs, nouns from verbs and grammatical words. These dictionaries are part of the dictionary of simple words. In this section, we discuss this dictionary of simple words on the one hand and the dictionary of multi-word units on the other hand. The NLP dictionary of simple words is exposed first and the NLP dictionary of multi-word units follows.

8.1 NLP Dictionary of Simple Words

The dictionary of simple words will be composed of dictionaries of verbs, nouns, adjectives and grammatical words.

A dictionary of verbs is already available and distributed with Unitex. A large number of inflected forms of verbs are covered and recognized with this resource. As for nouns from verbs and grammatical words, dictionaries are about to be finished. There is about 1100 entries to insert in the dictionary of nouns from verbs and it will be completed. For grammatical words, it is under construction. Soon, nouns from verbs and grammatical words will be covered and recognized by Unitex.

Our challenge is to construct the other NLP dictionaries, such as NLP dictionaries of nouns themselves, nouns from adjectives, adjectives themselves and adjectives from verbs, in order to complete our NLP dictionary of simple words. Other dictionaries such as technical dictionaries are also needed.

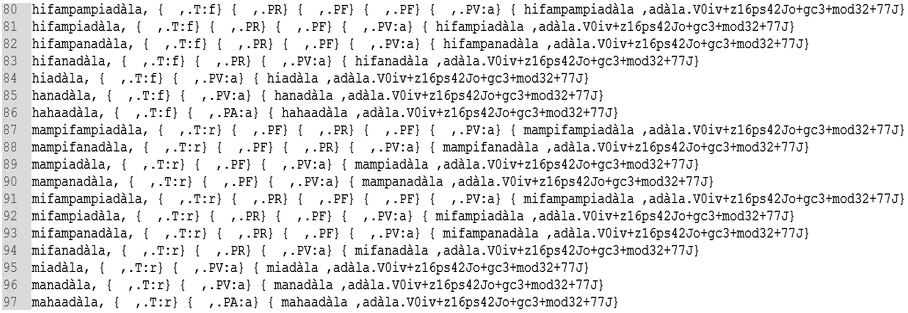
When those resources are completed, a complete dictionary of Malagasy simple words becomes available. Tools such as spell checking, machine translation and other services may become accessible more easily.

8.2 NLP Dictionary of Multi-word Units

The construction of an NLP dictionary of multi-word units is a heavy and complex task, but such a project is not unrealistic.

The construction of such a dictionary for Malagasy is different from the similar project implemented for French [28] because the entries for Malagasy dictionary are not inflected forms, but roots. However, multi-word units are in principle compositions of inflected forms. Efforts should be made to build this dictionary because it can increase the accuracy of text analysis systems.

A step we have made is to construct a program of generation of inflected forms. Entries of the program are dictionaries of roots and in the result a list of inflected forms of roots is displayed. For example, for verbs, the dictionary of roots provides all the inflected forms of verbs in the result. For the root *adàla* "to fool" for example, the program lists all the inflected forms of verbs attached to it. The list contains 97 inflected forms. This program is under construction but below we just furnish an excerpt of the results of this program for the cited example (Fig. 12).



```
80 hifampampiadàla, {  ,.T:f} {  ,.PR} {  ,.PF} {  ,.PF} {  ,.PV:a} { hifampampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
81 hifampiadàla, {  ,.T:f} {  ,.PR} {  ,.PF} {  ,.PV:a} { hifampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
82 hifampanadàla, {  ,.T:f} {  ,.PR} {  ,.PF} {  ,.PV:a} { hifampanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
83 hifanadàla, {  ,.T:f} {  ,.PR} {  ,.PV:a} { hifanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
84 hiadàla, {  ,.T:f} {  ,.PV:a} { hiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
85 hanadàla, {  ,.T:f} {  ,.PV:a} { hanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
86 hahaadàla, {  ,.T:f} {  ,.PA:a} { hahaadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
87 mampifampiadàla, {  ,.T:r} {  ,.PF} {  ,.PR} {  ,.PF} {  ,.PV:a} { mampifampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
88 mampifanadàla, {  ,.T:r} {  ,.PF} {  ,.PR} {  ,.PV:a} { mampifanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
89 mampiadàla, {  ,.T:r} {  ,.PF} {  ,.PV:a} { mampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
90 mampanadàla, {  ,.T:r} {  ,.PF} {  ,.PV:a} { mampanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
91 mifampampiadàla, {  ,.T:r} {  ,.PR} {  ,.PF} {  ,.PF} {  ,.PV:a} { mifampampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
92 mifampiadàla, {  ,.T:r} {  ,.PR} {  ,.PF} {  ,.PV:a} { mifampiadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
93 mifampanadàla, {  ,.T:r} {  ,.PR} {  ,.PF} {  ,.PV:a} { mifampanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
94 mifanadàla, {  ,.T:r} {  ,.PR} {  ,.PV:a} { mifanadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
95 miadàla, {  ,.T:r} {  ,.PV:a} { miadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
96 manadàla, {  ,.T:r} {  ,.PV:a} { manadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
97 mahaadàla, {  ,.T:r} {  ,.PA:a} { mahaadàla ,adàla.V0iv+z16ps42Jo+gc3+mod32+77J}
```

Fig. 12. Results of the program of generation of inflected forms of roots

This program should work on our other dictionaries to produce a dictionary of inflected forms of Malagasy. Technical modification in its operation are being considered.

9 Conclusion

Charon [3] discusses general information about media and devices in the digital era. For Malagasy language, journalists as native speakers of Malagasy generally present information in this language. Several types of language data in Malagasy are available (written documents, images, sounds, videos) in substantial quantities in libraries, archives, radio and television networks and even in national or private press centres. They all have in common that they cannot be really explored. Tools are not available for general public. For example tools to correct the language, to translate voices, etc.

The emerging collaboration between developers and linguists is likely to open Malagasy to the digital sphere and indirectly to enhance the teaching of this language. The construction of NLP dictionaries with root-entries enabling automatic recognition of different inflected forms of simple words would be beneficial for both goals. Spell checking, text processing, information retrieval, information extraction or translation, when available to the general public, would be useful services. The majority of people who use computers need them. The users ask for example for the availability of tools which can correct errors in their writing on computer.

For now, building readable and updatable dictionaries of nouns themselves, nouns from adjectives, adjectives themselves and adjectives from verbs is a challenge for research on Malagasy language. Such resources would help developers to construct adaptable tools for this language and would introduce the Malagasy language into the era of the digital age. They would help users to use more of the language and facilitate communication between humans and artificial intelligence.

References

1. Andrianony: Gramera na fianarana ny teny Malagasy. LMS, Antananarivo (1960)
2. Berlocher, I., Huh, H.G., Laporte, É., Nam, J.S.: Morphological annotation of Korean with directly maintainable resources. In: Poster session of LREC, Genoa (2006)
3. Charon, J.M.: Les medias à l'ère numérique. Les cahiers du journalisme **22**(23), 15–26 (2011)
4. Cousins, G.: A Concise Introduction to the Study of the Malagasy Language as Spoken in Imerina. LMS, Tananarive (1894)
5. Dalrymple, M., Liakata, M., Mackie, L.: Tokenization and morphological analysis for Malagasy. Comput. Linguist. Chin. Lang. Process. **11**(4), 315–332 (2006)
6. Diwersy, S.: Corpus journalistique du malgache contemporain. Romance Philology Department, University of Cologne (2009)
7. Enguehard, C., Mangeot, M.: LMF for a selection of African Languages (2014)
8. Freeman, J.J.: A Dictionary of the Malagasy Language: English and Malagasy. LMS, Antananarivo (1835)
9. Griffiths, D.: A Grammar of the Malagasy Language. Edward Pite, Woodbridge (1854)
10. Gross, M.: La construction de dictionnaires électroniques. Ann. télécommun. tome **44**(1–2), 4–19 (1989)
11. Koskenniemi, K.: Two-Level Morphology: A general Computational Model for Word-Form Recognition and Production. Department of General Linguistics, University of Helsinki (1983)
12. Koskenniemi, K., Church, K.W.: Complexity, two-level morphology and Finnish. In: COLLING 1988 (1988)
13. Krauwer, S.: The basic Language Resource Kit (BLARK) as the first milestone for the language resources roadmap. In: SPECOM, Russia, pp. 8–15 (2003)
14. Laporte, É.: Separating entries in electronic dictionaries of French. In: Darski, J., Vetulani, Z. (eds.) Sprache - Kommunikation - Informatik, Akten des 26, pp. 173–179. Max Niemeyer, Poznan (1993)
15. Malzac, V., Abinal, A.: Dictionnaire Malgache – Français. Imprimerie de la Mission Catholique, Tananarive (1888)
16. Nam, J.S.: Construction d'un lexique électronique des noms simples en coréen. In: Lexiques-grammaires comparés et traitements automatiques, pp. 219–245. Jacques Labelle, Université du Québec à Montréal (1994)
17. Oflazer, K.: Two-level description of Turkish morphology. In: EACL'06, Netherlands (1993)
18. Paumier, S.: Unitex 3.0. User manual. Université Paris-Est. English version. Ludwig-Maximilians-Universität, Munich (2003)
19. Poibeau, T., et al.: The multilingual named entity recognition framework. In: EACL 2003, vol 2. Association for Computational Linguistics, USA (2003)

20. Rabenilaina, R.B.: Construction du dictionnaire électronique du malgache parallèlement à celui du français. Communication au Colloque International sur les Industries de la langue, du 21 au 24 Novembre à Montréal, publiée en 1991. In: Actes du Colloque Tome 1. Office de la Langue Française et Société des Traducteurs du Québec, Montréal (1989)
21. Raboanary, J., et al.: Correction orthographique d'un texte écrit en malagasy. In: Forum de la Recherche. Ministère de l'éducation nationale, Antsiranana (2008)
22. Rajaona, S.R.: Structure du malgache. Études des formes prédicatives. Ambozontany, Fianarantsoa (1972)
23. Ralalaoherivony, B.S.: Quelques problèmes posés par la représentation des unités lexicales dans le dictionnaire morphologique du malgache. Université d'Antananarivo (2004)
24. Ranaivoarison, J.N.A., Laporte, É., Ralalaoherivony, B.S.: Formalisation of Malagasy conjugation. In: 6th Language and Technology Conference, Poznan, pp. 457–462 (2013)
25. Ranaivoarsion, J.N.A.: Modélisation de la morphosyntaxe du malgache. Construction d'un dictionnaire électronique des verbes simples. Ph. D. University of Antananarivo (2014)
26. Ranaivoarison, J.: The Malagasy language in the digital age. Challenges and perspectives. In: 7th Language and Technology Conference, Poznan, pp. 299–303 (2015)
27. Ranaivoarison, J.: Dictionnaire électronique des noms issus de verbes du malgache. Les noms issus des alternances mp- ou f-. In: Journées d'études toulousaines, JéTou 2017, Toulouse, pp. 106–112 (2017)
28. Silberztein, M.: Le dictionnaire électronique des mots composés. Lang. fr. (87), 71–83 (1990)
29. Weber, J.: Dictionnaire Malgache – Français. Notre Dame de la Ressource, Île Bourbon (1853)

Author Index