Theory and Applications of Natural Language Processing
Edited volumes

Chris Biemann
Alexander Mehler *Editors*

Text Mining

From Ontology Learning to Automated Text Processing Applications

Springer

Theory and Applications of Natural Language Processing

Aims and Scope

The field of Natural Language Processing (NLP) has expanded explosively over the past decade: growing bodies of available data, novel fields of applications, emerging areas and new connections to neighboring fields have all led to increasing output and to diversification of research.

"Theory and Applications of Natural Language Processing" is a series of volumes dedicated to selected topics in NLP and Language Technology. It focuses on the most recent advances in all areas of the computational modeling and processing of speech and text across languages and domains. Due to the rapid pace of development, the diversity of approaches and application scenarios are scattered in an ever-growing mass of conference proceedings, making entry into the field difficult for both students and potential users. Volumes in the series facilitate this first step and can be used as a teaching aid, advanced-level information resource or a point of reference.

The series encourages the submission of research monographs, contributed volumes and surveys, lecture notes and textbooks covering research frontiers on all relevant topics, offering a platform for the rapid publication of cutting-edge research as well as for comprehensive monographs that cover the full range of research on specific problem areas.

The topics include applications of NLP techniques to gain insights into the use and functioning of language, as well as the use of language technology in applications that enable communication, knowledge management and discovery such as natural language generation, information retrieval, question-answering, machine translation, localization and related fields.

The books are available in printed and electronic (e-book) form:

* Downloadable on your PC, e-reader or iPad
* Enhanced by Electronic Supplementary Material, such as algorithms, demonstrations, software, images and videos
* Available online within an extensive network of academic and corporate R&D libraries worldwide
* Never out of print thanks to innovative print-on-demand services
* Competitively priced print editions for eBook customers thanks to MyCopy service http://www.springer.com/librarians/e-content/mycopy

More information about this series at
http://www.springer.com/series/8899

Chris Biemann • Alexander Mehler
Editors

Text Mining

From Ontology Learning to Automated Text Processing Applications

Festschrift in Honor of Gerhard Heyer

Springer

Editors

Chris Biemann
Computer Science Department
Technische Universität
Darmstadt FG Language Technology
Darmstadt
Germany

Alexander Mehler
Computer Science Department
Goethe University WG Text Technology
Frankfurt am Main
Hessen
Germany

ISSN 2192-032X ISSN 2192-0338 (electronic)
ISBN 978-3-319-12654-8 ISBN 978-3-319-12655-5 (eBook)
DOI 10.1007/978-3-319-12655-5
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014959093

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Text Mining (TM), a variant of Data Mining (DM) on text data, is an important discipline of Natural Language Processing (NLP). The analysis of large amounts of linguistically pre-processed textual data is not only a prerequisite of building lexical resources such as domain-specific ontologies and language-related dictionaries. It also has direct applications in automated text processing regarding the whole range of linguistic levels such as morphology, syntax, semantics and pragmatics.

Facing the availability of huge amounts of online data and a surge in quantitative methods of analysis, this book summarizes recent trends of Text Mining. It positions Text Mining with respect to neighbouring fields, and shows Text Mining in applications as diverse as mobile devices, sentiment and deception analysis, and the digital humanities. In spite of the success that had been made during the last decades with respect to the statistical analysis of linguistic data, we are still in need of a profound understanding of Text Mining methods that allow for integrating symbolic approaches to deep linguistic analysis on the one hand, and surface-oriented statistical approaches on the other hand. This book collects several contributions into the direction of such an integrated approach.

This book is organized into two parts. In Part I, techniques and methodologies of TM are described. Eckart et al. ask the question about the ‘right’ size of a text corpus that is used for TM and propose a methodology for large corpus creation. The organization of lexical knowledge is the subject of two chapters: Tanev learns networks of linked word clusters and Kozareva learns taxonomies from lexical resources—both approaches construct an ontology-like structure from text in a bottom-up fashion. Oesterling et al. provide techniques for visually analysing document collections to make TM information visible to the user, and Mehler et al. introduce coreness as a network-based model of text vocabularies that is used to characterize Latin texts in the framework of computational humanities.

Part II contains a variety of examples where TM is used in practice—be it for the humanities, in NLP systems or in industrial applications. Bordag et al. discuss challenges regarding NLP on mobile devices and propose a TM-inspired unsupervised paradigm for their multilingual setting. Oemig and Blobel shed light on the role of NLP and TM for interoperability issues in health care. As examples of TM within NLP applications, Perez-Rosas et al. construct a deception detection system with the help of crowdsourcing, and Sonntag and Stede highlight the role of TM for sentiment analysis. The book is concluded with two projects from the Digital Humanities: Düring and van den Bosch give a case study for historical event detection, and Büchler et al. discuss historical text reuse methodologies.

This book is dedicated as a Festschrift in honour of Gerhard Heyer's 60th birthday. Coming from traditional, logic-based linguistics, Gerhard Heyer was an early promoter of Text Mining methodologies and statistical methods for natural language processing in industry and science. Being one of the first professors for language technology in a computer science department in Germany, he has had a major influence in making computational linguistics research applicable in practice. Proposing a triad of data, methods and applications, his contributions range over dictionary creation, large-scale corpora statistics, data structures, learning methods and diachronic analysis over visual analytics, digital humanities and network analysis to knowledge representation formalisms and research infrastructure projects. Playing the role of an enabler, Gerhard Heyer's diversity and broadness regarding Text Mining methods and applications are reflected in the contributions in this book.

Darmstadt, Germany Chris Biemann
Frankfurt, Germany Alexander Mehler
January 2015

List of Reviewers

- Antal van den Bosch, RU Nijmegen, The Netherlands
- Tim vor der Brück, U Frankfurt, Germany
- Christian Chiarcos, U Frankfurt, Germany
- Judith Eckle-Kohler, TU Darmstadt, Germany
- Johannes Leveling, Dublin City U, Ireland
- Andy Lücking, U Frankfurt, Germany
- Anke Lüdeling, HU Berlin, Germany
- Verena Henrich, U Tübingen, Germany
- Karo Moilainen, TheySay Ltd., UK
- Animesh Mukherjee, IIT Kharagpur, India
- Vivi Nastase, FBK-irst, Italy
- Barbara Plank, U Copenhagen, Denmark
- Simone Paolo Ponzetto, U Mannheim, Germany
- Uwe Quasthoff, U LEipzig, Germany
- Steffen Remus, TU Darmstadt, Germany
- Martin Riedl, TU Darmstadt, Germany
- Eugen Ruppert, TU Darmstadt, Germany
- Roman Scheider, IDS Mannheim, Germany
- Serge Sharoff, U Leeds, UK
- Manfred Stede, U Potsdam, Germany
- Herman Stehouwer, RZ Garching, Germany
- Fabio Massimo Zanzotto, U Rome 'Tor Vergata', Italy
- Torsten Zesch, U Duisburg-Essen, Germany

Contents

Part I Text Mining Techniques and Methodologies

Part II Text Mining Applications

Part I
Text Mining Techniques and Methodologies

Building Large Resources for Text Mining: The Leipzig Corpora Collection

Uwe Quasthoff, Dirk Goldhahn, and Thomas Eckart

Abstract Many text mining algorithms and applications require the availability of large text corpora and certain statistics-based annotations. To ensure comparability of results a standardized corpus building process is required. Particularly noteworthy are all pre-processing procedures as they are crucial for the quality of the resulting data stock. This quality can be estimated by both evaluating the corpus building process and by statistical quality measurements on the corpus. Some of these approaches are described using the example of the Leipzig Corpora Collection.

1 Introduction: The Need for Large Resources

1.1 What is the Right Size of a Corpus?

Today, the Web is one of the main resources for text corpora because of the vast amount of HTML files containing text of all kinds. However, the processing steps for building a corpus from HTML files are tricky at some points and building your own corpus for every new problem is not feasible. Prepared corpora in different languages and different sizes bring the resources closer to the user. This user also needs some kind of evaluation to estimate the resource's appropriateness for a specific task. Especially for larger corpora, their quality is not easy to determine. In this chapter we want to address the quality problem as it is done at the Leipzig Corpora Collection [14]. Apparently the quality of large corpora cannot be guaranteed by human proof reading. Instead, careful pre-processing is needed to extract text consisting of well-formed sentences in the desired language. Statistical measurements on the corpus are used to check the quality of the result.

Due to technical problems when dealing with large corpora we often prefer smaller corpora. But how to determine the smallest size which will fit our needs?

U. Quasthoff (✉) • D. Goldhahn • T. Eckart
Natural Language Processing Group, University of Leipzig, Leipzig, Germany
e-mail: quasthoff@informatik.uni-leipzig.de;
dgoldhahn@informatik.uni-leipzig.de; teckart@informatik.uni-leipzig.de

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_1

Table 1 Frequency for German phrasemes in large corpora

Multiword unit	Frequency in 26M news	Frequency in 260M Web
ein Auge zudrücken	77	592
zwischen Baum und Borke	15	204
sich auf den Hosenboden setzen	4	11
aus der Fasson geraten	0	1
mit Glanz und Gloria untergehen	0	1
letzter Rest vom Schützenfest	0	0

Table 2 Frequency for rare plural forms in German

Singular	Frequency	Plural	Frequency
Bronchitis	5,454	Bronchitiden	81
Cellulitis	391	Cellulitiden	0

Assume we want to investigate certain objects of interest (maybe words, typical patterns of words, or syntactic structures) in the corpus. To analyze a certain structure, we need a minimum number of occurrences. Often this minimum number of occurrences is between 5 and 20 because in this range the identification of different possible contexts is already feasible. Obviously for most statistics-based approaches, more than ten occurrences are needed. As the frequency of many objects of interest follows Zipf's law [31] only some of them have a reasonable high frequency, and many of them are very infrequent. Hence, it is easy to find enough examples for the high frequent objects, but we need larger and larger corpora for the infrequent constructs. The following example is taken from German phraseology: Table 1 shows the frequencies of some widely known expressions in two corpora of 26 or 260 million sentences.

We need even more data if we want to know that something is not existent. Let us consider the following example: Do the nouns *Bronchitis* and *Cellulitis* have the plural forms *Bronchitiden* and *Cellulitiden*? Unfortunately we cannot estimate the frequencies for these infrequent plurals given the frequencies for the singular forms. The results for the same 260 million sentences corpus are given in Table 2. The numbers might indicate that the plural of *Cellulitis* is not used (i.e. not existing even in larger corpora) while it is explicitly given in dictionaries like Duden [9].

1.2 How Much Text is There for a Certain Language?

Building larger corpora for the "big" languages is normally not limited by resources. This is not the case for the lesser resourced languages where the amount of available texts limits corpus building. The total amount of texts is of course increasing over time for nearly all languages, but it increases non-steadily and language dependent.

For instance, the free access to the first Zulu online newspaper *Isolezwe*[1] multiplied the number of pages in Zulu. The following parameters can be used to estimate the number of Web pages for a given language. Every value should be treated cautiously, but together they can be used to measure the amount of text in a given language in the Web.

- How many newspapers are available in that language? Newspaper directories like ABYZ News Links[2] are valuable resources.
- How many speakers does a language have? Especially if this number is low, we cannot expect to find very much text. For the converse, i.e. languages with many speakers, we have to consider the status of the language in the corresponding country and the stage of economic development.
- Number of Wikipedia articles in this language.
- Search engine counts: Take the top-5 or top-10 stop words of the language and use a search engine's number of pages to compare the values for different languages. This is known to be a very rough estimation and sometimes misleading, especially for lesser resourced languages, see [20] (Kilgarriff). But in future, these data might become more reliable.

2 Standardization and Availability

2.1 Standardized Processing

For the automatic creation of textual resources in arbitrary languages a processing chain is necessary which is, at least to a large extent, language independent. In the following paragraphs such a tool chain for building Web corpora will be presented. Focus will be on necessary steps for acquiring texts from the Web and for processing the data.

2.1.1 Crawling

With the extraordinary growth of information in the World Wide Web, online documents increasingly become the major source for creating high quality corpora of large size. The Leipzig Corpora Collection [14] combines different strategies for collecting such textual data from the WWW. The main goal is to ensure that corpora of large extent and high diversity can be created for specific languages.

[1] http://www.iol.co.za/isolezwe.

[2] http://www.abyznewslinks.com.

Generic Web Crawling

For downloading textual data from the Web, we rely on Heritrix,[3] a framework for massively parallel Web crawling. This crawler project is an open-source tool made available by the Internet Archive Community. It can be used for periodically creating snapshots of large amounts of Web pages. Heritrix is a versatile tool, providing many options to configure the desired crawling behavior. Results are outputted in the standardized Warc-format.

Heritrix can be used in several ways. On the one hand it is capable of crawling whole Top Level Domains (TLDs). In this case a small list of domains of a country of interest is used as an input. The composition of this list has only minor influence on the results since hubs of a TLD are reached rather quickly. Heritrix is then configured to follow links only within this TLD. On the other hand Heritrix can be used to download a fixed set of domains. Web pages such as ABYZ News Links[4] offer lists of language specific domains. ABYZ News Links provides more than 32,000 news sources in about 120 languages. For each domain it offers information regarding country and language. The Heritrix based Web crawler is used to download these news sources.

Distributed Web Crawling

FindLinks [19] is a distributed Web crawler utilizing a client-server architecture. The Java-based client runs on standard PCs and processes a list of URLs, which it receives from the FindLinks server. After downloading, the client extracts text and hyperlinks from the documents before returning data to the server. FindLinks has been used with community support for several years and allowed us to crawl the Web to a large extent.

Bootstrapping Corpora

In addition an approach similar to [2] Baroni and [27] Sharoff is applied. A small set of frequent terms is needed for languages in question. Therefore existing corpora of the LCC or other sources such as the Universal Declaration of Human Rights (UDHR)[5] are utilized as a resource. Based on these lists tuples of three to five high frequent words are generated. These tuples are then used to query search engines such as Google. The resulting URLs are used as a basis for the default Heritrix crawling framework.

[3] https://webarchive.jira.com/wiki/display/Heritrix/Heritrix.

[4] http://www.abyznewslinks.com.

[5] http://www.ohchr.org.

Crawling of Special Domains

Certain domains are beneficial sources for Web corpora since they contain a large amount of text in predefined languages. One example is the free Internet encyclopedia Wikipedia, which is available in more than 200 languages. Wikipedia dumps for these languages are downloaded and Wikipedia Preprocessor[6] is used for further processing and text extraction.

2.1.2 Pre-processing

After downloading documents further steps for creating corpora in multiple languages have to be taken. Among them are HTML-stripping, language identification, sentence segmentation, cleaning and sentence scrambling. Due to the amount of textual data and the number of different languages, an automatic tool chain has been implemented. It is easily configurable and only minor language-dependent adjustments, concerning e.g. abbreviations or sentence boundaries, have to be made.

HTML-Stripping

The collected data are HTML-coded. Therefore HTML-Tags, JavaScript and other elements need to be eliminated from the documents. Html2Text, a HTML-stripping-tool of the NLP-group of the University of Leipzig, is used. Further cleaning steps need to follow since the resulting text data still contains more than just well-formed sentences.

Language Identification

For most documents, only their URL but not the language of their content is known. Since our aim is to create monolingual corpora, we use LangSepa, a tool built at the NLP group of the University of Leipzig, to identify the language of a document. LangSepa compares the distribution of stop-words or character unigrams and character trigrams of various languages to the distribution within the documents. Hence, statistical knowledge concerning these distributions for many languages is a requirement. If available, corpora of the LCC were used to acquire these data. Documents such as the Universal Declaration of Human Rights were used to classify further languages—adding up to about 450 languages in total.

Sentence Segmentation

Next, sentence segmentation has to take place. This step is based on lists with typical sentence endings. These can easily be created using resources such as

[6] http://sourceforge.net/projects/wikiprep/.

sonderzeichen.de.[7] For better sentence boundary detection, abbreviation lists can be utilized. For several languages these lists already exist and more lists will be prepared.

Cleaning

The resulting elements are cleaned in additional steps. First, non-sentences are identified based on patterns that a normal sentence should obey. Strings that do not comply with these patterns are removed from the data. In a second step quality is enhanced by eliminating sentences that do not belong to the considered language [24]. Subsequently, duplicate sentences are removed. Especially newspaper reports may appear nearly unchanged on various websites. The same holds for standard boilerplates. To prevent a distortion of the statistical analysis, they are discarded.

Sentence Scrambling

To avoid copyright restrictions the collected sentences are "scrambled" by destroying the structure of the documents. This inhibits the reconstruction of the original material. This approach is considered safe with respect to German copyright legislation as there is no copyright on single sentences in Germany.

Creation of Text Databases and Statistics

The data is stored in relational databases to allow for quick and generic access. A full form dictionary with frequency information for each word is generated utilizing statistical methods. Statistical data is enhanced with information about word co-occurrences. Further analyses concerning e.g. part of speech tagging are also conducted, if tools are available for the desired language. Once the text databases have been created, they are made accessible online. Alternatively, the data can be downloaded[8] and viewed locally with the Java-based *Corpus Browser* [25]. In addition the data can be queried using a SOAP API.

2.2 Standardization in Distributed Infrastructures

The recent years have seen a tendency to understand linguistic resources not as individual assets but to emphasize on their compatibility and combinability. This enables the individual users to search and find data, use them in a variety of contexts and share their results with other researchers. Furthermore standardized formats and interfaces on different levels allow the aggregation of data to form "virtual

[7] http://www.sonderzeichen.de.

[8] http://corpora.informatik.uni-leipzig.de/download.html.

collections" that can easily exceed the largest resources currently known. Prominent members of these integrated research environments are (among others) the European projects CLARIN[9] and DARIAH.[10]

For a fully integrated environment standardization efforts are necessary on different levels:

- Standardized metadata for searching and identification of relevant resources,
- Standardized access methods for easy access to resources,
- Standardized formats to enhance compatibility and comparability of resources and
- Standardized search interfaces for building federated content search environments.

Due to its homogeneous and standardized processing procedures (as described in Sect. 2.1) the Leipzig Corpora Collection already fulfilled most of the necessary prerequisites for these tasks. This specifically means that in most cases only thin compatibility layers were necessary to integrate LCC corpora in the research environment CLARIN. These interfaces are built since 2011 including structured metadata based on the Component MetaData Infrastructure CMDI [5], RESTful webservices for all kinds of access to the corpora and the support of the annotation environment WebLicht via the Text Corpus Format TCF [18]. Furthermore a significant share of the LCC corpora can already be queried by using the Contextual Query Language (CQL) via the SRU protocol.[11]
Automatic integration procedures were created that enrich corpora with relevant metadata (if necessary), check for consistency and plausibility and allow access for external researchers with minimal manual effort. As a consequence new corpora are integrated into CLARIN on a nearly weekly basis.

3 The Leipzig Corpora Collection

3.1 Evolution of the LCC

Since the start of the Leipzig Corpora Collection LCC as "Projekt Deutscher Wortschatz"[12] in the 1990s the project was subject to many changes in its mission and employed technologies. Started as a resource provider for digital texts in German language mostly based on newspaper articles and royalty-free text material

[9] http://www.clarin.eu.

[10] http://www.dariah.eu.

[11] http://weblicht.sfs.uni-tuebingen.de/Aggregator.

[12] http://wortschatz.uni-leipzig.de.

(like literature provided by the digital library Project Gutenberg[13]) the Leipzig Corpora Collection today contains resources for text mining in several hundred languages, divided under several aspects like year of acquisition, text genre, country of origin and more.

A first major step in the direction of a massive multilingual data provider was the implementation and deployment of the international Web portal in 2005 that included corpora in 15 different languages. At that time the focus was mostly on European languages with high numbers of native speakers (like English, German, Spanish, French, and Italian). As non-European languages only Japanese and Korean were included. This portal is still in use today,[14] although with a considerably extended number of languages and sublanguages.

The following years were characterized by adaptations and improvements of existing workflows to achieve a mostly language independent processing pipeline while taking language characteristics into account (for more details see Sect. 3.2). Furthermore new ways of text acquisition were taken into operation. Naturally a focus was given on acquisition methods based on online accessible text material. As a consequence both trends combined lead to a rapid growth of the amount of corpora in different languages over the last years (c.f. Fig. 1).

The availability and extent of text material accessible via different Web protocols also allowed to extend the primary focus from newspaper articles to new kinds

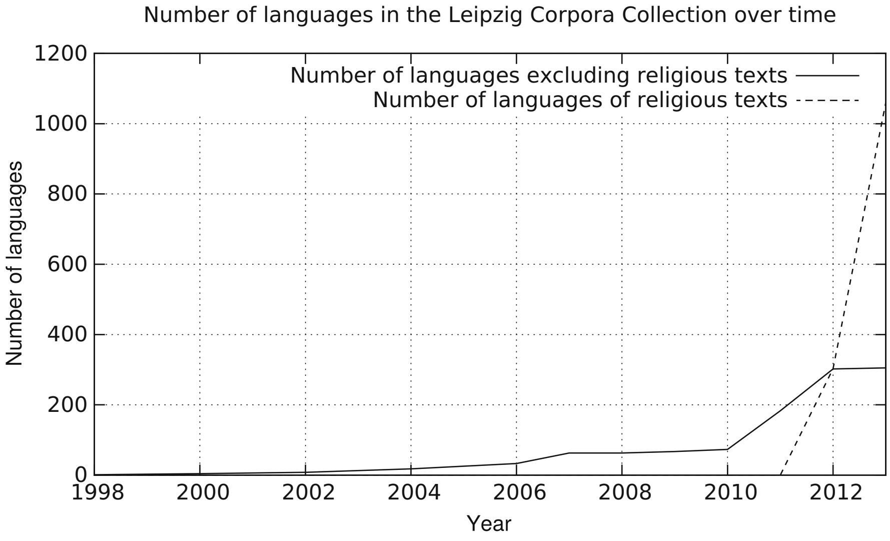


Fig. 1 Number of languages in the LCC from 1998 to 2013

[13] http://www.gutenberg.org.

[14] http://corpora.uni-leipzig.de.

Table 3 Number of sentences of crawled material in the LCC for different genres over time (in millions)

Year	Web	News	Wikipedia	Religious	Misc.	Sum
2013	3.4	10.7	0	***18.8***	9.8	42.7
2012	191.8	147.1	***63.3***	1.7	4.5	408.3
2011	131.5	***262.0***	23.2	0	***20.0***	436.7
2010	20.3	132.0	56.0	0	0	208.3
2009	0	177.8	0.2	0	5.7	183.7
2008	0	149.3	0	0	0	149.3
2007	0	108.8	49.6	0	0	158.3
2006	0	57.1	0	0	0	57.1
2005	13.1	42.7	0	0	1.2	56.9
2004	7.0	5.3	0	0	0	12.3
2003	18.2	4.6	0	0	0	22.8
2002	***657.7***	15.7	0	0	0	673.4
2001	0	3.7	0	0	0	3.7
2000	0	3.2	0	0	0	3.2
1999	0	3.4	0	0	0	3.4
1998	0	2.1	0	0	0	2.1
1997	0	2.3	0	0	0	2.3
1996	0	4.0	0	0	0	4.0
1995	0	3.4	0	0	0	3.4

of text material and genres. Table 3 gives a short summary of the amount of data acquired over the last 20 years. For each genre the year with the highest number of sentences is highlighted. Obviously both the amount of text (given in numbers of extracted sentences in million) and the diversity of material grew over time. Please note that raw material aggregated in 2013 is not processed yet completely, which explains the low numbers for this particular year.

3.2 Deep Processing

During corpus creation basic statistical information is computed including word frequencies. In a next step additional statistical evaluation takes place which is a requirement for many text mining applications. For this deeper processing we follow different approaches. For statistics such as word co-occurrences and similarities of words or sentences, techniques are used which are to a large part language independent. This allows us to compute them for all corpora. On the other hand, information about part of speech and other features of language are compiled utilizing language specific knowledge and can only be created for certain languages.

3.2.1 Word Co-occurrences

All corpora of the Leipzig Corpora Collection are annotated with statistical information about word co-occurrences. These features are computed for different window sizes of the textual environment considered. In our case they are based on co-occurrence in the same sentence or in a direct neighborhood. All word relations were generated by using the log-likelihood ratio [6] as a measure of significance. For each word the resulting values of all significant co-occurrences are stored for evaluation.

3.2.2 POS Tagging

In addition, text corpora are enriched with information about part of speech of words. For POS tagging different approaches are possible, namely supervised and unsupervised techniques. Existing unsupervised methods suffer from problems such as a low quality of results, a demand for large amounts of text or the need for manual mapping of results to actual word classes [3]. Therefore supervised POS tagging techniques are applied. While producing results of high quality they demand a manually tagged corpus to train the classifier for each language considered. We apply tools such as TreeTagger [26] and HunPos [17] in combination with training sets which are freely available on the Web. So far text corpora in 17 languages, among them many widely used languages, can be tagged. In a current advance an extension to lesser resourced languages is in progress.

3.2.3 Word Similarities

By computing word similarities text corpora can be enriched with metadata to allow for further linguistic studies. These similarities can be computed within a single corpus or across corpora and languages, opening up different possibilities for their use. When working with large corpora containing millions of different word forms a high-performance algorithm for computing similarities is necessary since words have to be compared in a pairwise manner. The Fast Similarity Search algorithm (FastSS) [4] was chosen for solving this task concerning text collections of the LCC. For word similarities it was implemented for simple string similarity based on character level.

Monolingual

For the LCC we use string similarity to analyze words of single monolingual corpora. Word pairs of a Levenshtein distance of up to two are stored for further analyses. This mainly results in pairs or groups of words which are related by morphological processes. Most prominently these processes are based on word

Table 4 Words with a Levenshtein distance of less than 3 to the German word *zahlen* ordered by word rank

Word	Word rank	Levenshtein distance
Zahlen	501	0
bezahlen	1870	2
zahlten	19311	1
Bezahlen	24361	2
zahlende	33507	2
zuzahlen	127648	2
abzahlen	142503	2
ZAHLEN	192285	0
Zahlern	224713	1
Zwahlen	336005	1
Zahlende	370584	2
anzahlen	391993	2
Abzahlen	484286	2
Anzahlen	527826	2
Zahlten	575030	1
Q-Zahlen	634670	2
zaehlen	648914	1
Zahlens	980778	1
zahle an	1137835	2
BEZAHLEN	1792380	2

formation such as inflection or derivation. This especially holds for longer words, while short words are often similar just by chance. Further reasons for words being similar are, among others, spelling errors.

In Table 4 the most frequent words with a Levenshtein Distance of less than 3 of the German word *zahlen* (*to pay*) can be found. Most of them are created by inflection or derivation.

Cross Lingual

In addition cross lingual pairs of similar words can be considered. In this case words from two different corpora of the LCC serve as input for the FastSS algorithm. Typically results consist to a large extent of proper names and cognates.

3.2.4 Sentence Similarities

Similar entities are not only relevant on word level but also on sentence level. Concerning sentences similarity can be computed using different features. On the one hand one could search for similarity on surface level and only utilize string comparison. On the other hand one could describe the sentences using different properties or statistics. The latter is done for corpora of the LCC: We generate descriptions of the sentences based solely on statistics such as POS-tags or word length of each word. Using these features the sentence *Peter likes chocolate* would be assigned the descriptions *NNP VBZ NN* or *5 5 9*. Because of computational complexity we once again use FastSS to compute similar sentences. Table 5 depicts pairs of English sentences being similar based on word lengths. Especially pairs of longer sentences often differ in only one word being inserted or changed. In Table 6 a group of German sentences with similar POS information can be found.

One application for sentence similarity is the search for near duplicate sentences. The statistics presented are by themselves already helpful for this task. For future work a combination of them might help to discover near duplicates with more precision.

Table 5 Sentence pairs with a distance of less than 3 using word length to describe the sentences

Sentence
A company that wants to frack for oil ***outside of*** Calgary city limits is facing opposition from nearby residents who fear the project will poison their tap water.
A company that wants to frack for oil ***close to*** Calgary city limits is facing opposition from nearby residents who fear the project will poison their tap water.
A complaint said Kodirov contacted an unidentified person trying to buy weapons in early July, and that person became a confidential source for the government.
A complaint said Kodirov contacted an unidentified person trying to buy weapons in early July ***2011***, and that person became a confidential source for the government.
A component of the latest builder confidence survey that measures current sales conditions rose 2 points to 51, ***the*** highest ***level*** since April 2006.
A component of the latest builder confidence survey that measures current sales conditions rose 2 points to 51, highest since April 2006.

Table 6 Sentences with a distance of less than 2 using POS to describe the sentences

Sentence
19 Einrichtungen entlassen die jungen Menschen ins richtige Leben.
19 Schüler besuchten die erste Klasse im ersten Schuljahr.
28 Holzstufen trennen das moderne Reicheneck vom 19. Jahrhundert.
1.000 Kadetten setzen ein Zeichen zur nationalen Einheit.
103.000 Euro betrug der Abgang im vergangenen Jahr.
10 Kandidaten durchlaufen ein 3monatiges Fitnessprogramm im Internet.
11.600 Kinder besuchten den Spielplatz im offenen Spielhaus.
12.000 Zuschauer verfolgten das anderthalbstündige Spiel zu guten Zwecken.
148,5 Kilogramm lautete das Wiegergebnis am heutigen Morgen.
15.29 Uhr: Ein kurzer Blick ins benachbarte Westfalen.
157 Meter ging Aschenbachs weitester Flug am ersten Wertungstag.
16,9 Minuten dauert das durchschnittliche Vorspiel in deutschen Betten.
16 Todesopfer forderte die neue Grippe laut britischer Regierung.
170 Besucher bejubelten die neuen Regenten beim Königsfrühstück.

3.3 Language and Corpus Statistics

There is a variety of applications for language and corpus statistics, especially when creating data for text mining purposes. Application of text mining always implies specific requirements on quality and composition of text material. Automatic quality assessment can be used to assure that minimum standards are met. Similar automatic approaches can be used to identify systematic differences and similarities between languages. As a consequence typological classification and corpus statistics can be correlated.

3.3.1 Quality

Quality assurance is always a relevant topic for the creation and annotation of corpora. This holds for a variety of problems in the pre-processing steps like language identification, tokenization and many more. When using Web crawling as primary data source assessment of the input material is especially a topic as hardly any assumptions about quality can be made. Manual inspection is not feasible for large amounts of text, hence statistical measures should be evaluated for their use as indicators for accurate corpora and as part of computer-assisted quality management.

Experiences showed that a variety of features can be used for this purpose where especially "outliers" (i.e. elements that in some way differ from the "norm") are interesting to identify problems. The Leipzig Corpora Collection has no dedicated focus an a specific language. In this case the applied features have to be independent from individual languages or domains but should address general properties of natural language texts. There are several features that can be exploited for this purpose, especially length distributions on several linguistic levels and typical frequency distributions (like the conformity to Zipf's law) seem to be useful [11]. Figures 2 and 3 demonstrate the usefulness of sentence length distribution (in characters) for two corpora of different quality. Figure 2 shows a typical distribution that can be found for most languages (in this case based on Hindi newspaper text). In contrast Fig. 3 shows a rather untypical result for a corpus based on Sundanese Wikipedia text. This unexpected distribution is a good indicator for problems with the input material. In this case the cause lies in (possibly machine generated) sentences which are grammatical but give a deformed impression of the language.

Other statistics also point to these quality issues. When looking at the relation of rank and frequency of words of the Sundanese corpus (Fig. 4), as described in Zipf's Law, we see an unusual behavior of the graph between rank 10 and 20. In addition character statistics can also indicate the problems described before. When looking at the most frequent character 5-grams at word beginnings (Table 7) only untypical

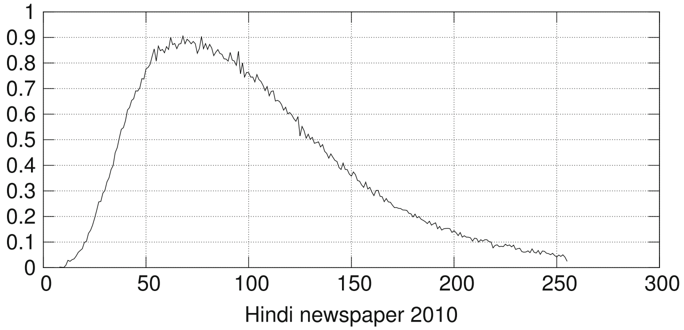


Fig. 2 Sentence length distribution for a Hindi newspaper corpus (percentage for number of characters)

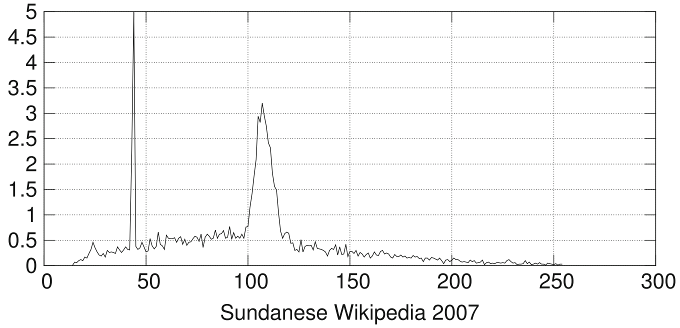


Fig. 3 Sentence length distribution for a Sundanese Wikipedia corpus (percentage for number of characters)

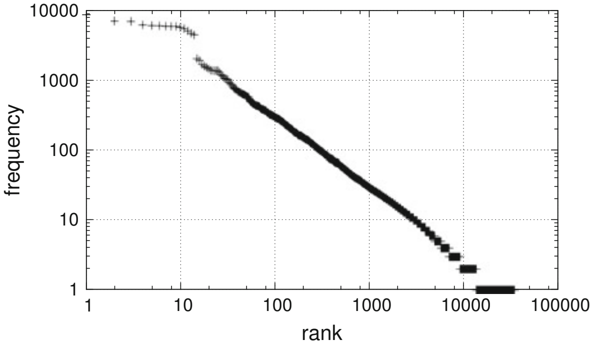


Fig. 4 Relation of rank and frequency of words of a Sundanese corpus

n-grams containing numbers and hyphens can be found. This statistic results from hundreds of identical sentences mentioning year dates (“Taun ka-1256 Maséhi dina Kalénder Grégorian.”).

Table 7 Most frequent character 5-grams at word beginnings of a Sundanese corpus

Rank	5-Gram
1	ka-12-
2	ka-13-
3	ka-10-
4	ka-14-
5	ka-18-

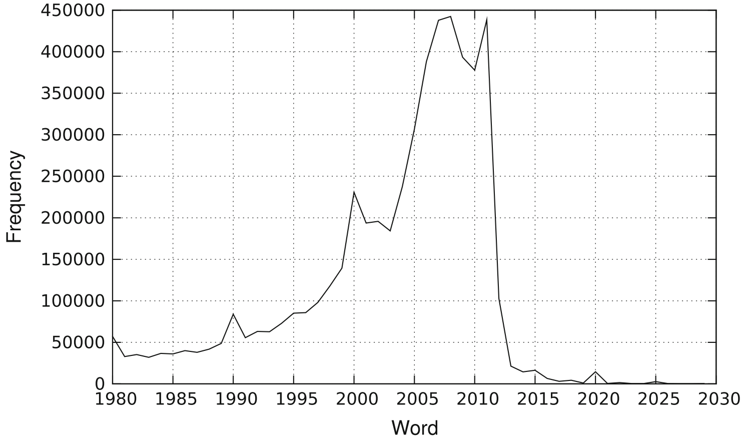


Fig. 5 Frequency of typical year dates in a French mixed corpus

3.3.2 Corpus Timeline

Another problem that occurs when dealing with crawled Web text is that it is hard to give meaningful information regarding the publication date of the text material. However, it has proven to be useful in analyzing the dates in the corpus as they characterize the time period described by the corpus. Figure 5 demonstrates this on the example of a French mixed corpus compiled by using mostly Web harvester output from the years 2005 to 2011. It shows the frequency of year dates from 1980 to 2030 where the greater share of recent texts between 2004 and 2011 stands out clearly. The higher frequency for 2000 probably results from the millennium being mentioned frequently, and not from there being a larger number of texts published in 2000. Therefore, it can be assumed that in most cases the year indicated is also the year of publication.

3.3.3 Language Description

Certain measurements on corpora vary greatly between languages. If such corpus statistics are chosen in a way to describe important features of languages they can be used to characterize languages. Possible measurements of this kind are:

- Average word length (distribution or average)

- Average sentence length in words or characters (distribution or average)
- Frequency or variability of typical word beginnings or endings (*n*-grams)
- Slope of Zipf's Law
- Text coverage (of *n* most frequent words)
- Vocabulary richness (such as Type Token Ratio, Turing's Repeat Rate)
- Entropy on word or sentence level
- Word co-occurrences (e.g. amount of next neighbor or sentence co-occurrences)
- Syllables (average number of syllables per word or sentence, average syllable length)

When using statistics to describe languages [10], it is necessary to be able to measure them for a large variety of languages. Therefore language independent measurements are advantageous.

Some of the statistics mentioned are easy to measure for most or all languages. Average word or sentence length in characters are two of them. When measuring sentence length in words the same holds as long as a tokenizer for all investigated languages exists. This is at least the case for those languages which use a white space to separate words. Information about word or sentence length can help to characterize languages in domains such as morphological complexity. Another statistic—mentioned already above—is Zipf's law [30, 31]. It makes a statement about the relationship between rank and frequency of words of a text resource. Figure 6 shows a typical Zipf distribution for a German news corpus. A possible approach for describing this distribution is to compute its average slope. For the German corpus this value is −0.99.

More complex measurements are for instance concerned with syllables. They are an important unit of language just like morphemes or phonemes and can form the basis for further linguistic investigations. It is difficult to determine the exact position of the boundaries of syllables without language specific knowledge. Measuring solely the number of syllables is advantageous since these values are

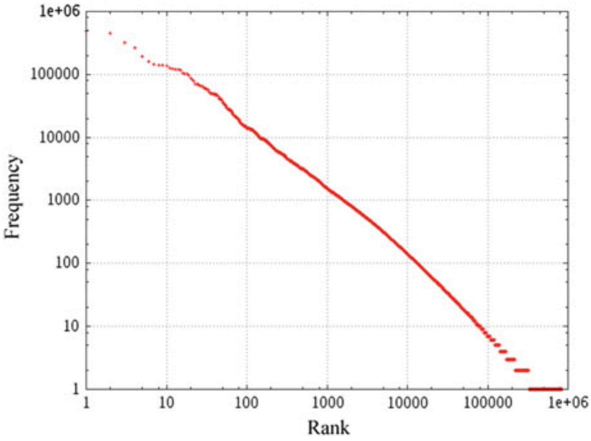


Fig. 6 Rank and frequency of words of a German news corpus. Both axes are logarithmically scaled

Table 8 Statistics based on syllable counts for a German, an English and a Czech corpus

Parameter (average)	German	English	Czech
Syllables per word (types)	2.74	2.34	2.60
Syllables per word (tokens)	1.62	1.43	1.71
Length of syllables (types)	3.03	2.91	2.78
Length of syllables (tokens)	2.99	2.83	2.67
Syllables per sentence	36.78	32.03	28.26

much easier to compute and statistics such as average length of syllables can still be determined. One way to approximate the number of syllables is to count the number of syllable peaks, since every syllable has exactly on peak. Peaks are typically vowels and in most cases consist of exactly one vowel. When ignoring diphthongs, which typically have low influence on the desired statistics for most languages, one can at least approximate syllable counts when counting vowels. Using the algorithm of (Sukhotin and Guy) [16, 28] one can identify the vowels and consonants of a particular language and then compute the desired syllable statistics. Table 8 depicts statistics based on syllable counts for a German, an English and a Czech corpus.

3.3.4 Application to Typology

A possible application of statistical measurements on corpora of many languages is *typology*. Typology is concerned with structural and functional features of languages, the distribution of these properties and the systematic relations between them. Instead of analyzing a language as a whole, often text corpora are used as a basis for typological studies. Typically few languages are analyzed and small corpora are used since manual measurements have to be taken [12]. However, recent studies used automated measurements and were thereby able to examine large corpora in several hundred languages [13, 15]. In comparison to other works in this field this process does not contain any manual steps and will be described in the following paragraphs.

When using text corpora for typological analyses different aspects have to be considered. On the one hand measurements should have a high cross-language standard deviation. On the other hand the influence of properties of the corpus such as size, genre or subject area on the measurements has to be investigated. Otherwise the statistical power of the following typological investigations might decrease. Hence, it is desirable to have a much higher cross-language *Standard Deviation* (SD) in comparison to the SD when varying these other textual properties. Examples for these comparisons can be found in Table 9.

Using corpus based measurements one can then conduct quantitative typological studies. It is possible to relate simple features of text corpora with classical typological parameters of language, which describe different levels of language such as morphology or syntax. Furthermore one can relate different measured features. This can be achieved using quantitative methods like correlation analysis

Table 9 Comparison of standard deviations (SD) of corpus-based measurements

Measurement	$\frac{SD(Lang.)}{SD(T.Size)}$	$\frac{SD(Lang.)}{SD(T.Type)}$	$\frac{SD(Lang.)}{SD(SubjectArea)}$
Average sentence length in words	107.41	8.65	13.20
Average sentence length in char.	77.03	6.23	7.67
Ratio of suffixes and prefixes	18.78	17.69	25.84
Syllables per sentence	30.25	8.22	7.33
Type-Token-Ratio	1.16	8.21	6.13
Turing's Repeat Rate	238.95	6.37	8.69
Slope of Zipf's Law	3.27	11.35	11.25
Text coverage of top 100 words	530.85	7.93	8.75

Values larger than 1 imply a higher cross-language standard deviation compared to the standard deviation when varying other features such as text size, text type or subject area

(Pearson Product Moment Correlation Coefficient) and tests of significance [22, 29] to analyze and confirm such relationships [7].

A small sample of results of [13] Goldhahn will be presented next to show the possibilities of this approach. Using these techniques it is possible to detect several correlations between measured parameters of corpora. By applying correlation analysis to comparable corpora in 730 languages, results of high statistical significance can be achieved and correlations can be confirmed or found, among them are:

- A negative correlation between average length of words and average length of sentences (in words): cor = 0.55; $p < 0.001\,\%$, sample size of 730. The longer the average word of a language is, the fewer words are usually utilized (or needed) to express a sentence. This finding is a special case of the Menzerath-Altmann-Law as described by [1] Altmann or [21] Köhler.
- A negative correlation between average number of syllables per word and average number of words per sentence: cor = 0.49; $p < 0.001\,\%$, sample size of 730. The more syllables the average word of a language has, the fewer words are typically used to express a sentence.

Relations between measured parameters and classical typological parameters can also be verified. Typological information can be taken from the World Atlas of Language Structures (WALS) [8]. Possible findings concerning different aspects of language are:

- A significant relation between measured ratio of suffixes and prefixes and position of case marking (end of word vs. beginning of word): $p < 0.001\,\%$, mean values of 10.48 and 0.7.
- A significant relation between average length of words of a language and its morphological type (concatenative vs. isolating): $p < 1\,\%$, mean values of 8.43 and 6.95.
- A significant relation between average number of syllables per sentence and word order (SOV vs. SVO): $p < 0.001\,\%$, mean values of 56.95 and 45.27.

- A significant relation between average number of syllables per word and morphological type (concatenative vs. isolating): $p < 5\,\%$, mean values of 2.06 and 1.64.

As shown by these examples, this approach to corpus-based linguistic typology allows for a wide range of analyses. Using an automatic process chain one can measure statistical features of corpora of Web text for several hundred languages. These properties can then be applied in quantitative typological analyses to detect correlations with classical typological parameters.

3.4 Multiword Units

Multiword units (MWUs) are sequences of words which are of interest for corpus exploration. From a technical point of view the type of such a MWU is rather irrelevant. For applications in text mining or linguistics we might be interested in the following kinds of multiwords:

- Proper names of persons, organizations, products, some geographic names, etc.
- Terminology
- Phraseology
- Compounds: in some languages, compounds are written using several words

Of course, simple search for a certain multiword can be done using single words and Boolean operations, but the following questions are difficult to answer without special treatment of multiwords:

- Find the frequencies for a very large number of multiwords. More specific: Rank a large list of person names by frequency.
- Which are the most significant words co-occurring with given MWUs?
- What are typical formation rules of multiword expressions in the language? How can they be compared with their equivalences in other languages?

In order to answer such questions a predefined list of MWUs is needed. Resources for multiwords are as follows:

- Titles of Wikipedia articles (mainly persons and terminology)
- Phraseology lists. They are available for many languages but need preprocessing because the dictionary form often differs from the inflected forms found in the corpus
- Terminology lists

Moreover, MWUs are generated by several algorithms. Their quality might differ (i.e. some of the proposed MWUs are not of the expected form), but often an over-generation of multiwords can be accepted. Algorithms generating multiword are, for instance:

- Named entity recognition (NER) algorithms detect proper names.

- Compounds written as separate words can often be identified by searching for an alternative continuous spelling of the multiword. In English, “after shave” may be considered as a MWU because we find the words “aftershave” and “after-shave” in the corpus.

3.5 Recent Developments and Future Trends

Additional processing steps generating more data are of interest if they are available for many languages. Moreover, the quality of the resulting data should be high enough to consider these data as useful. Interesting are, for instance, algorithms based on general machine learning techniques which can be trained for multiple languages. Moreover, larger training sets will help to achieve the desired quality. If the training data are generated by another algorithm, an additional error analysis followed by correction of the training data might be necessary.

The following data will be included next into the Leipzig Corpora Collection.

Morphological Analysis

Many POS-tagger do not return only POS tags, but also the lemmatized form of the corresponding word. These data can be used to train a lemmatizer. For a more complex morphological analysis, the corresponding training data is necessary.

Topic Modeling

For all corpora, topic models with 200 topics will be generated. A framework based on work of [23] Phan is already in testing phase on data of the LCC. The topics can be used for typical text mining tasks like document classification and disambiguation.

Interlingual Linking

Bilingual dictionaries can be used to link between words in different languages. Possible sources are Wiktionary,[15] Open Multilingual WordNet[16] and isolated bilingual dictionaries.

Acknowledgements We thank both anonymous reviewers and editors for their valuable hints and comments, which helped finalizing the chapter.

[15] http://www.wiktionary.org.

[16] http://compling.hss.ntu.edu.sg/omw.

References

1. Altmann G (1980) Prolegomena to menzerath's law. Glottometrica 2:1–10
2. Baroni M, Bernardini S (2004) BootCaT: Bootstrapping corpora and terms from the web. In: Proceedings of LREC 2004
3. Biemann C (2006) Unsupervised part-of-speech tagging employing efficient graph clustering. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: Student Research Workshop, pp 7–12. Association for Computational Linguistics
4. Bocek T, Hunt E, Hausheer D, Stiller B (2008) Fast similarity search in peer-to-peer networks. In: Network operations and management symposium, 2008. NOMS 2008, Salvador, 7–11 April 2008. IEEE, pp 240–247
5. Broeder D, Windhouwer M, van Uytvanck D, Goosen T, Trippel T (2012) CMDI: a component metadata infrastructure. In: Describing LRs with metadata: towards flexibility and interoperability in the documentation of LR workshop programme
6. Büchler M (2006) Flexibles Berechnen von Kookkurrenzen auf strukturierten und unstrukturierten Daten. Diploma Thesis, University of Leipzig
7. Cysouw M (2005) Quantitative methods in typology. In: Altmann G, Köhler R, Piotrowski R (eds) Quantitative linguistics: an international handbook. Mouton de Gruyter, Berlin, pp 554–578
8. Cysouw M (2008) Using the World Atlas of language structures. Introduction to the special issue of Sprachtypologie und Universalienforschung (STUF) 60(2):181–185
9. Duden (2009) Die deutsche rechtschreibung, Band 1, 25th edn. Dudenverlag, Mannheim/ Wien/Zürich
10. Eckart T, Quasthoff U (2013) Statistical corpus and language comparison on comparable corpora. In: BUCC - Building and using comparable corpora. Springer, Berlin
11. Eckart T, Quasthoff U, Goldhahn D (2012) Language statistics-based quality assurance for large corpora. In: Proceedings of Asia pacific corpus linguistics conference 2012, Auckland
12. Fenk-Oczlon G, Fenk A, (1999) Cognition, quantitative linguistics, and systemic typology. Linguist Typol 3:151–177
13. Goldhahn D (2013) Quantitative Methoden in der Sprachtypologie: Nutzung korpusbasierter Statistiken. Dissertation, University of Leipzig, Leipzig
14. Goldhahn D, Eckart T, Quasthoff U (2012) Building large monolingual dictionaries at the leipzig corpora collection: from 100 to 200 languages. In: Proceedings of the 8th international conference on language resources and evaluation (LREC 2012)
15. Goldhahn D, Quasthoff U, Heyer G (2014) Corpus-based linguistic typology: a comprehensive approach. In: Proceedings of konvens 2014, Hildesheim
16. Guy JB (1991) Vowel identification: an old (but good) algorithm. Cryptologia 15(3):258–262
17. Halácsy P, Kornai A, Oravecz C (2007) HunPos: an open source trigram tagger. In: Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, pp 209–212. Association for Computational Linguistics
18. Heid U, Schmid H, Eckart K, Hinrichs E (2010) A corpus representation format for linguistic web services: the D-SPIN text corpus format and its relationship with ISO standards. In: Proceedings of the seventh international conference on language resources and evaluation (LREC'10), 2010
19. Heyer G, Quasthoff U (2006) Calculating communities by link analysis of URLs. In: Innovative internet community systems. Springer, Berlin, pp 151–156
20. Kilgarriff A (2007) Googleology is bad science. Comput Linguist 33(1):147–151
21. Köhler R, Altmann G, Piotrowski R (2005) Quantitative linguistik (Quantitative linguistics). In: Ein internationales handbuch (An international handbook). De Gruyter, Berlin
22. Mann HB, Whitney DR (1947) On a test of whether one of two random variables is stochastically larger than the other. Ann Math Stat 18(1):50–60

23. Phan X, Nguyen C, Le D, Nguyen L, Horiguchi S, Ha Q (2011) A hidden topic-based framework toward building applications with short web documents. Knowl Data Eng IEEE Trans 23(7):961–976
24. Quasthoff U, Biemann C (2006) Measuring monolinguality. In: The workshop programme of LREC 2006, p 38
25. Richter M, Quasthoff U, Hallsteinsdóttir E, Biemann C (2006) Exploiting the leipzig corpora collection. In: Proceedings of the IS-LTC 2006
26. Schmid H (1994) Probabilistic part-of-speech tagging using decision trees. In: Proceedings of international conference on new methods in language processing, Manchester
27. Sharoff S (2006) Creating general-purpose corpora using automated search engine queries. In: Baroni M, Bernardini S (eds) WaCky! Working papers on the web as corpus. Gedit, Bologna
28. Sukhotin BV (1988) Optimization algorithms of deciphering as the elements of a linguistic theory. In: Proceedings of the 12th conference on computational linguistics-association for computational linguistics, vol 2, pp 645–648
29. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
30. Zipf GK (1935) The psycho-biology of language: an introduction to dynamic philology. The MIT Press, Cambridge
31. Zipf GK (1949) Human behavior and the principle of least effort. Addison-Wesley, Cambridge

Learning Textologies: Networks of Linked Word Clusters

Hristo Tanev

Abstract Ontologies have been used in different important applications like information extraction, generation of grammars, query expansion for information retrieval etc. However, building comprehensive ontologies is a time consuming process. On the other hand, building a full-fledged ontology is not necessary for every application which requires modeling of semantic classes and relations between them. In this chapter we propose an alternative solution: learning a *textology*, that is, a graph of word clusters connected by co-occurrence relations. We used the properties of the graph for the generation of grammars and also suggest a procedure for upgrading the model into an ontology. Preliminary experiments show encouraging results.

1 Introduction

Ontologies have been used in different important applications such as information extraction, generation of grammars, query expansion for information retrieval etc. However, building comprehensive ontologies is a time consuming process. For example, the Cyc ontology encompasses about one million facts and rules manually created in a period of time of about 30 years [18]. Unfortunately, the huge amount of human effort required to build such a semantic resource is a serious disadvantage from the point of view of developers. Therefore, machine learning methods for ontology learning and population are very important, since they can reduce the time of developing an ontology. Different methods for learning ontologies were proposed recently in the literature, e.g. [12, 17]. Most of these methods require some already existing resource, such as a dictionary, which can be difficult to find, for example, for low-resourced languages and specialized domains.

H. Tanev (✉)
European Commission, Joint Research Centre, Ispra, Italy
e-mail: hristo.tanev@jrc.ec.europa.eu

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_2

On the other hand, building a full-fledged ontology is not necessary for every application which requires modeling semantic classes and relations between them.

In this chapter we propose an alternative solution which brings the ontology paradigm closer to the text. That is, we propose that semantic classes are substituted by clusters of distributionally similar words and multiwords and instead of semantic relations, we acquire co-occurrence relations, labeled with textual phrases which link the words from the related clusters. We will present several algorithms of distributional semantics as well as a link generation method, which starting from a set of seed words learn clusters of co-occurring words, represented as a *textology*: a network of word clusters, connected by co-occurrence links labeled with textual phrases.

The idea about textologies is partially inspired by the textual entailment paradigm [6], where entailment relations are derived between textual expressions rather than logical formulas or other semantic representations. In a similar way, textologies model concepts and relations on a textual level, rather than in an ontological framework.

We define a *textology* in the following way:

Definition 1 A textology is a directed labeled graph $T = \{V, E, L_v, L_e\}$ with a set of vertices V and a set of directed edges (arcs) E, connecting some pairs of vertices from V, where two or more arcs from E may link the same pairs of vertices in different directions. The functions L_v and L_e are labeling functions, defined respectively on the vertices and the arcs. L_v assigns to each vertex $v \in V$ a cluster of words and multiwords, which are lexicalizations of a concept represented by v. L_e assigns to each edge e, linking vertices v_a and v_b, a set of *connector phrases*, sequences of words which tend to connect the words from the two clusters $L_v(v_a)$ and $L_v(v_b)$. The concept lexicalizations on the vertices and the connector phrases on the links are derived from a given text corpus C.

A small textology graph from the domain of public demonstrations is shown in Fig. 1. This textology graph contains different word clusters. Let us consider some of the most important ones: Cluster *demonstrators* contains lexicalisations of the *demonstrators* concept: *protestors, demonstrators, supporters*, etc. The cluster *demands* models the concept *demand/criticize*. It contains words like *demanded, requested, criticised*, etc. Another cluster named *government* contains one word *government*. All these three clusters are linked with each other, since in the above mentioned domain of protests and demonstrations, words from these clusters tend to co-occur with each other. Let us consider the arc *demands* $\rightarrow$ *government*. It means that the words from the cluster *demands* tend to co-occur with the words from the cluster *government*, and they precede them linearly. This arc will be labeled with connectors which tend to connect *demanded, criticized, requested*, etc. with *government*. For example, the first connector is "." which denotes immediate co-occurrence, such as in the bigram *demanded government*, *criticized government*

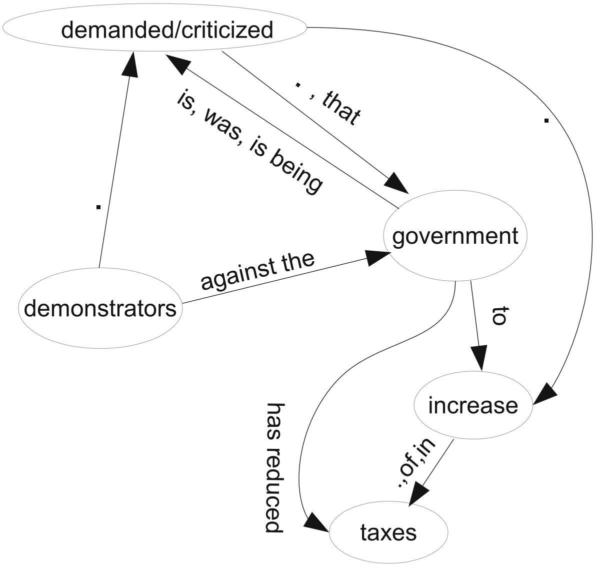


Fig. 1 Example of a textology from the domain of protests

etc., the other connector is *that*, which is used in phrases like *demanded that government*. All these collocations can be parts of phrases describing that demands or criticisms are directed towards a government. In a similar way, the opposite arc *government* → *demand* is labeled with connectors like *is*, *was* and *is being*, which are used in phrases like *government is criticized*, *government was criticized*, *government is being criticized*.

A textology encodes both semantic and syntactic information: the clusters represent semantic concepts and most of the paths in the textology graph encode valid phrases from the domain. This representation can be used to generate grammars for detecting domain-specific entities, situations and events. A textology can also be seen as an intermediate representation of algorithms that aim at learning an ontology. Along this line of thinking, the clusters can be considered to represent the classes of an ontology, while some of the textology arcs may be seen as a basis for the definition of semantic properties.

It is noteworthy that the representation under discussion is based on linear co-occurrences, which can be learnt from unannotated text corpora without using any language-specific processing. Therefore, our model is language-independent.

The structure of the chapter is as follows: In Sect. 2 we talk about related work. In Sect. 3 we present a method for building textologies. The method is based on several algorithms of distributional semantics. In Sect. 4 we describe a preliminary evaluation based on an experiment in which a textology is built to generate a grammar for event detection. Finally, we present our conclusion.

2 Related Work

Recently, different approaches for ontology learning and population have been proposed (for an overview see [3, 7] among others). Two approaches for learning expressive ontologies are presented in [17]: a lexical approach to generate complex class descriptions from definitional sentences and a logical approach to generate constructs of a general purpose-ontology such as disjointness axioms. A method for transforming dictionary glosses into ontology structures is presented in [12].

Concept and pattern learning are strongly related to ontology learning. However, they have been used also outside of the context of ontologies. Relevant to our work is the concept learning algorithm described in [14]. This algorithm finds concepts as sets of semantically similar words. It uses distributional clustering by applying a clustering algorithm, called CBC (*Clustering by Committees*). A good example for information extraction-related concept and pattern learning is presented in [15]. In this work, bootstrapping is introduced: as input, the system obtains a handful of lexicalizations for each concept and in every iteration it learns context patterns which are used, in turn, to obtain new concept lexicalizations. It was shown that this method succeeds in harvesting patterns and semantic classes with good levels of accuracy. Recently, a new concept and pattern learning architecture for *Never Ending Language Learning* (NELL) was developed and described in [4]. The NELL system uses the Web to learn concepts and patterns. It exploits a bootstrapping algorithm which is running continuously as a Web-based learning agent.

A common feature of the approaches mentioned so far is that they rely on language-specific parsers and part-of-speech taggers. Further, all of them work only for English.

The algorithms for building a textology are based on an underlying representation model called *word association graph*. This model was used in psychology to model human cognitive processes and human memory [5], [19]; in *Natural Language Processing* (NLP) it has been used by [2, 9, 10].

Relevant to our work is the association rule mining described in [8].

3 Building Textologies

In order to build a textology graph representation of a domain, we follow several basic steps, which include both running algorithms of distributional semantics, manual selection and cleaning of the learnt clusters. The methodology can be represented with the following schema, in which $CS_{textology}$ denotes the set of textology clusters, which in the beginning is an empty set:

1. Create an initial term set $TS_{initial}$ by manually selecting few terms from the target domain. The term set may contain words from different parts of speech, such as nouns, proper names, verbs, adjectives, etc. There can be both single words and multiwords. As an example consider:
   $TS_{initial} = \{demanded, criticized, protestors, demonstrators\}$

2. Create manually an initial set of semantic clusters $CS_{initial}$ from the initial term set, where a cluster represents the lexicalizations of a semantic concept, defined by the user. Each cluster contains one or more terms from the initial term set (optionally, their morphological forms can be added). In our example we will have $CS_{initial} = \{\{protestors, demonstrators\}, \{demanded, criticized\}\}$.
3. Run the automatic *cluster expansion algorithm* on the $CS_{initial}$ and expand each cluster $c \in CS_{initial}$ with additional lexicalizations of the corresponding concept. For example, our algorithm adds two more lexicalizations: *supporters* and *crowd* to the class *{protestors, demonstrators}.*
4. Set the list of the textology clusters $CS_{textology} \leftarrow CS_{textology} \cup CS_{initial}$.
5. Select manually a set of semantic clusters $Sel \subset CS_{textology}$ and run the *semantic context learning algorithm* which learns *context word clusters Ctx*, co-occurring with the selected cluster set *Sel*. For example, let us select both clusters from $CS_{initial}$, that is, in our example $Sel = CS_{initial}$. One cluster which co-occurs with the two initial clusters is *{salary, salaries, wages}.*
6. Each of the newly generated *context clusters* from $c \in Ctx$ is labeled with its two highest ranked words. On the basis of these two-word labels, the user preselects clusters which seem to be relevant to the target domain and application of the textology graph.
7. Each of the preselected *context clusters* is then manually reviewed and if it is relevant, it is manually cleaned and optionally the *cluster expansion algorithm* is run to expand it. In this way, a new set of relevant clusters is obtained *RCS*. In our example we select two new clusters, that is, *RCS={{increase, drop, increases, reduction, decrease, improvement, cuts}, {salaries, wages, salary, wage, payments, compensation}}.*
8. $CS_{textology} \leftarrow CS_{textology} \cup RCS$
9. If the user thinks this is necessary, go to step 5 or step 1.
10. Run the automatic *link detection algorithm* which detects the directed edges $E_{textology}$ among the clusters in $CS_{textology}$ and the connector phrases which label them. We assume that the words in the clusters will have a similar syntactic behavior, because of the distributional similarity among them. However, this is not always true and some connector phrases will not be valid for all the words in the clusters they connect. It is up to the user to decide if she/he has to leave such connector phrases.
11. If necessary, manually edit the labels on the arcs

The final textology graph which we obtain is a graph composed of the clusters from $CS_{textology}$ and the directed edges from $E_{textology}$. The obtained graph shows how the words from the different classes tend to co-occur with each other.

This graph can be used to generate a grammar, which we show in the following sections. The grammar built from the toy example textology graph, introduced so far, can recognize phrases like *protestors demanded increase in the salaries*, *protestors criticized the cuts in the wages* etc.

The procedure for building textologies exploits two algorithms based on distributional semantics: namely the *the cluster expansion algorithm* and *the semantic context learning algorithm*. Also, a *link detection algorithm* is exploited, which finds the links between the textology clusters and their labels.

3.1 Word Association Graph

The textology concept was inspired by the notion of a word association graph, a graph model used in psychology [19] and NLP [9, 10]. In a word association graph words are related to each other on the basis of some association measure. Associations could be acquired through interviewing people or they can be gathered more indirectly by mining word co-occurrences from a text corpus, such that one can speak of *co-occurrence graphs*. This second method provides association mining on a larger scale.

In NLP, word association (or co-occurrence) graphs have been used for automatic scoring of essays [10], personal name alias extraction [1], keyword extraction [11] and studying Web communities [13]. Properties and applications of word association graphs in relation to the problem of semantic search are discussed in [2].

In general, edges in association graphs as described in the literature represent co-occurrences, which reflect generic semantic relations. Heyer et al. [9] analyzes types of semantic relations underlying these graphs.

Before learning semantic clusters and links between them, we create a word association multigraph in which two words are connected by one or more arcs, if they co-occur on a close distance. Each arc in our word association graph represents one co-occurrence pattern, which is characterized by the left and the right word (considering the linear order of co-occurrences) and the words which occur between them. In our word graph model we allow only sequences of stop words between words. We also consider a more generic co-occurrence pattern in which two words are connected if they appear on a distance of less than five tokens without considering the tokens between them.

In our approach the word associations are used as an underlying data representation, which enables the textology learning algorithms to mine for co-occurrences, which are later used for clustering, cluster expansion and learning links. It is noteworthy that we consider as vertices in word association graphs both words and multiword units, where multiwords are selected on the basis of co-occurrence patterns of their constituents. Frequencies of the occurrences of vertices and arcs are also provided by word graphs.

Note finally that textologies can be seen as extensions of word association graphs where instead of words, vertices denote clusters of words.

3.2 Algorithms

In this section, we present several algorithms for building textologies. These algorithms include the *cluster expansion algorithm*, the *semantic context learning algorithm* and the *link detection algorithm*. As pointed out earlier, these algorithms make use of a word association graph extracted from an unannotated text corpus.

3.2.1 The Cluster Expansion Algorithm

The purpose of this algorithm is to automatically expand word clusters. The algorithm takes as input a set of *seed word clusters* $C = \{c_1, c_2, c_3, c_4, \ldots, c_n\}$ and expands each of them with other words and multiwords of a similar distribution profile as the corresponding seed cluster. Distributionally similar words in general represent similar concepts—they can be synonyms, for example, or have a common ancestor in their hypernym chains.

The algorithm is iterative; on each iteration three main steps are performed and the user(s) decide when the iterations are to stop. The algorithm steps are: (a) Finding contextual features. (b) Extracting new words and multiwords using the contextual features extracted in step (a). (c) Manually deleting inappropriate candidates. On each iteration we lower the threshold for selecting new words.

The *Cluster expansion algorithm* has a version called *cluster expansion in a context*, which expands the clusters from C in the context of a reference cluster cr. In practice, it is the same algorithm, but during the extraction of new words, we leave only those, which in the word association graph are adjacent to at least one word from the cluster cr. In this way, the newly learnt words for the clusters in C are related to cr. For example, if we want to learn paraphrases of the phrase *blocked the road*, we can expand a cluster with the word *blocked* in the context $cr = \{road\}$. In this way, we get additional words like *disrupted* and *marched*, which in combination with *road* mean *road blocking* as, for example, in *marched on the road* and *disrupted the road*. On the other hand, if we expand the cluster *blocked* without context, we get additional words like *stop* and *prevent* which may be related to *blocked*; now, they cannot be used to express a road blocking situation.

It is noteworthy that the selection threshold for this version of the algorithm is lower than in the case of the non-contextualized expansion. This is because we have an additional criterion, that is, *reference class*, which ensures a better accuracy.

Learning Contextual Features

For each semantic class c_i we iterate through its members words and multiwords, denoted by $lexicalitems(c_i)$. For each lexical item $lxi \in lexicalitems(c_i)$ we extract *contextual features* from each arc and node adjacent to lxi in the word association graph. The contextual features are two types: left and right features.

Left features are formed from the arcs which go into *lxi*; they are formed from the adjacent words or multiwords, concatenated with the connector sequence on the arc which connects them to *lxi*. Left features are n-grams which appear immediately on the left from *lxi* in the text corpus from which the word graph is created. In a similar way we extract from the word association graph the so-called right features which appear on the right side of *lxi*.

Each contextual feature of a class c_i is assigned a score which shows how well it co-occurs with the seed terms. The score is calculated as a sum of the co-occurrence score of the feature with respect to each lexical item from c_i. The co-occurrence with the single lexical items is based on the pointwise mutual information. The details of feature scoring are based on a formula described in [16]. As an example, some of the top-scoring left contextual features of the class *protestors* are *disperse the*, *tear gas at*, *force against* and some of the top-scoring right context features are *chanted*, *carried signs*, *had gathered*.

Next, we take for each cluster $c_i \in C$ the top scoring features and merge them into a contextual-feature pool, which constitutes a semantic space where the clusters are represented. The contextual features for each cluster c_i form a *context vector* $v_{context}(c_i)$ which represents the semantics of c_i through its typical contexts. Each dimension in this vector is a contextual feature, extracted for any of the considered clusters and the co-ordinates are the scores calculated with our algorithm.

Learning New Lexical Items

After contextual features are learnt for each seed cluster from C, our procedure extracts new lexical items which tend to co-occur with these contextual features. In this way, we obtain lexical items which have a similar distributional profile as the seed clusters. We use the word association graph to extract the nodes which co-occur with each contextual feature extracted on the previous step.

Our algorithm represents each lexical item *lxi* as a vector $v_{context}(lxi)$ in the space of contextual features. The dimensions of this semantic space are defined by the set of contextual features extracted before. The coordinate of a lexical item *lxi* with respect to the dimension f reflects the co-occurrence trend between *lxi* and the contextual feature f. For more details of this procedure see [16].

Our term learning approach calculates the relevance of a lexical item *lxi* for a category c, using the following formula

$$termscore(lxi, c) = \frac{v_{context}(lxi) \cdot v_{context}(c)}{|v_{context}(c)|} \tag{1}$$

This relevance value computes the projection of the lexical item vector on the category vector. Candidates which are above a certain threshold are returned to the user.

3.2.2 Semantic Context Learning

The goal of this algorithm is to find clusters of words and multiwords which tend to co-occur with the lexical items from one or more clusters. This algorithm allows us to expand the textology graph beginning from few clusters. The algorithm accepts as input a set of clusters $C = \{c_1, c_2, \ldots, c_n\}$ and a number k. It learns clusters of words which tend to co-occur with at least k input clusters. The semantic context learning works in two main steps: (a) Learn words and multiwords co-occurring with at least k clusters from C and (b) cluster these lexical items using their contextual features.

Learning Words and Multiwords Co-occurring with the Input Clusters

In this step the algorithm extracts from the word graph all those words that co-occur with the input clusters. The co-occurrence score of each lexical item for each cluster is calculated by analogy to the feature scores within the cluster expansion algorithm. The final score of each lexical item is the sum of its co-occurrence scores with respect to the different clusters under consideration. Lexical items that are co-occurring with less than k clusters are discarded.

Finding Clusters

In this step, the algorithm first finds the vectors of contextual features of the lexical items extracted in the previous step. Next, the lexical items are clustered using the cosine similarity between these vectors. The feature vector is computed the same way as in the cluster expansion algorithm. We perform agglomerative clustering based on bottom-up average-linkage. Finally, the clusters are ordered by taking into account the highest scoring lexical item in each cluster. For example, the top-scoring context clusters for $C = \{\{protest\}, \{school, schooling, education\}\}$ and $k = 2$ are *{children, kids}*, *{parents, mothers, firefighters}* and *{teachers, faculty, volunteers, professors}*, while the top scoring context clusters for $C = \{\{protestors\}\}$ and $k = 1$ are *{police, gardai}, {streets} and {disperse,quell}.*

3.2.3 Link Detection Algorithm

This algorithm detects arcs between pairs of clusters of the textology. To this end, the algorithm explores the word association graph for arcs between lexical items that belong to the textology clusters. More precisely, the algorithm takes as input a set of clusters of a textology, which are already learnt: $C = \{c_1, c_2, \ldots, c_n\}$. Then, for each pair of clusters c_i and c_j it finds all arcs from the word association graph that connect each pair of lexical items $w_a \in c_i$ and $w_b \in c_j$. The arcs, which are labeled by the same connector words and which link words from the same pair of clusters in the same direction, are merged to what we call a *generalized arc*. More precisely, we create a generalized arc from cluster c_i to cluster c_j, labeled by a connector phrase l, if there is more than one arc in the word association graph, labeled with l and linking two different pairs of lexical items (w_a, w_b), such that $w_a \in c_i$ and $w_b \in c_j$. Here, we put a restriction for more than one arc based on empirical observations. However, depending on the corpus or the application a higher threshold could be applied and more equally labeled arcs may be required in order to produce a generalized arc. The generalized arcs become arcs of the textology. Finally, the user can manually delete some of the learnt arcs if necessary. As an example, consider $C = \{c_1, c_2\}$, where $c_1 = \{protest\}$ and $c_2 = \{school, schools, education\}$. In this case, the arcs generated by our algorithm are

$$\{school, schools, education\} \xrightarrow{in} \{protest\} \tag{2}$$

$$\{protest\} \xrightarrow{against} \{school, schools, education\} \tag{3}$$

4 Using Textologies

4.1 From Textologies to Ontologies

The classes in a textology represent concepts which in the context of ontologies can be represented as classes. For example, in Fig. 1, from the textology we can generate the concepts *demand/criticize*, *government*, *demonstrators*, *increase* and *taxes*. Then, more general classes can be introduced together with the corresponding *is-a* relations between them. For example, we can create the class *people* as a hypernym of *demonstrators*, *institution* as a hypernym of *government* and *action* as a hypernym of *increase* and *demand/criticize*.

Occasionally, several clusters can be used to create one complex semantic class. For example, the clusters *government*, *increase* and *taxes* and the links between them describe phrases which may motivate the introduction of a concept named *institutional-action*. Moreover, some of the clusters, which represent binary predicates may be modeled as properties, if they express fundamental relations between concepts. This, however, depends on the ontology.

Then, some of the relations among clusters can be used as a basis for creating properties. Note that arcs in a textology represent syntagmatic relations, rather than semantic ones. Thus, some of them will not be transformed into properties. Further, many arcs may be merged into a single property. Looking at the aforementioned example, we can define the property *agent-of-demand/criticism* of the domain *demand/criticise* and range *people*. Also the property *target-of-demand/criticism* may be generated which has the same domain as the previous property while its range is *institution*.

It is noteworthy that preposition connectors usually represent different relations, however they are limited in the range of the introduced semantic relations. For example, the preposition *of* may introduce a property, e.g. *the price of the smartphone*, a *part-of* relation, e.g. *the display of the smartphone*, or ownership, e.g. *the car of Mr. Orlov*. In order to deduce the semantic relation from a surface co-occurrence like *X of Y*, one can consider the semantics of the words which fill the *X*- and the *Y*-slot of this pattern. For example, words like *length* and *price* as instances of the *X*-slot designate properties, while in cases where nouns that refer to objects occur in the *X*-slot, the pattern refers to instantiations of the *part-of*-relation, e.g. *the roof of my home* or ownership *the office of the boss*—depending on the word which fills the *Y* slot. If it is a non-animate object like *home* that occurs in the X-slot, then we have most probably a *part-of*-relation, while in the case of nouns denoting animate objects like *boss*, we most probably get an instance of the *possession*-relation.

While the method described so far is quite general, we think that creating a textology can be an important step in building an ontology.

4.2 Grammar Generation

Grammars provide a natural mechanism to parse text and link it to semantic concepts and properties. We present in this section a grammar learning algorithm, which is based on the textology model. For us a grammar is a set of production rules of two types: (1) Rules of the type $C \rightarrow (w_1, w_2, \ldots, w_n)$, where C designates a cluster and $w_1 - w_n$ are words from this cluster. (2) Rules of the type $A- > C_1 connector_1 C_2 \ldots$ which encode more complex expressions. There C_i is a symbol, referring to a cluster and $connector_i$ refers to a connector phrase. The grammars we learn can be considered to be lexicalized grammars. They do not encode the linguistic structure of the expressions and do not make reference to parts of speech or syntactic categories. Rather than that, they refer to the clusters from the textology and the lexicalizations from these clusters.

There are two important steps in creating grammars from a textology: First, each cluster can be represented as a non terminal symbol which generates the words which are elements of the cluster. Second, some arcs or paths in the textology can be transformed into context-free grammar rules representing more complex phrases. For example, in Fig. 1 the clusters *demand/crticise* and *demonstrators* can give birth to the following production rules, which generate terminal strings:

(1) DemandCriticise → (*demanded*|*criticised*|*demanding*|*seeking*| ...)*}*

(2) Demonstrators → (*demonstrators*|*protestors*| ...)

Then, the textology arc *demonstrators* → *demand/criticise* may give birth to the higher level rule (3), which parses more complex phrases like *demonstrators demanded*.

(3) DemonstratorsDemand → *Demonstrators DemandCriticise*

We use the following algorithm to generate grammar rules:

1. Manually select the set of paths P from which rules are to be generated. Suppose, we select only one path: *demonstrators* → *demanded/criticised* → *government*
2. All the clusters on any paths from P are stored in the set C. In our example, $C = \{demonstrators, demanded/criticise, government\}$.
3. For each cluster $c \in C$, create a rule, which generates as terminal strings all the words from the cluster. For example, *Demonstrators* → (*demonstrators*|*protestors*| ...).
4. For each $p \in P$, generate a grammar rule, where on the righthand side each cluster from P is represented as a non-terminal symbol, the non-terminals are ordered as the corresponding clusters appear in p and if between two clusters the arc is tagged with a symbol, different from "." (which means immediate co-occurrence), then the labels are inserted as terminal strings between the corresponding non terminals. In our example, the following rule will be generated: *A* → *Demonstrators DemandedCriticised Government*. In this rule there are no terminal strings on the righthand side, since all the arcs on the path are tagged with ".". The symbol *A* on the left was chosen arbitrarily.

Following the described procedure, one can easily generate one or more grammars from a textology by selecting different paths.

Grammars generated with this procedure have some limitations: They cannot contain recursion and also they work on only two levels—the first level generates terminal strings and the second level generates more complex phrases from these terminal strings. Although the complexity of the grammars generated from the aforementioned procedure is limited, we show here that these grammars can be used for practical purposes.

5 Experiments and Evaluation

5.1 Building a Textology

In order to test the feasibility of our method, we built a textology which represents domain knowledge about protests and civil disorders. In this experiment we concentrated in particular on the violence and demands of protestors. In order to build this textology we indexed as a word association graph about 100,000 news articles. The most important participants in this type of events are the protestors, therefore we used as an initial term set for the textology learning algorithm the set {*protestors*, *demonstrators*}.

We run the cluster expansion algorithm on this seed set with five iterations, the system suggested six new terms altogether. Five of these terms were correct, i.e. could be considered lexicalisations of the concept "protestors". In this clue, the accuracy for the expansion of this initial seed class was found to be **83 %**.

Table 1 Accuracy of cluster expansion

Algorithm	Average number correct learnt words	Average accuracy
Cluster expansion	1.9	0.74
Cluster expansion in context	2.72	0.33

Using the expanded clusters *protestors*, we run the context learning algorithm which found about 300 clusters of words which tend to co-occur with the initial seed class. We looked through the top 100 clusters, where we looked at the top two words from each cluster. In this way we chose 25 clusters for further review. From these, we found 20 clusters to be very relevant to the domain of violence in protests and civil disorders. Some of the remaining clusters had some relation with the target domain, however they were not as related as the 20 selected ones. Then, we run the cluster expansion for these 20 clusters. The accuracy of the expansion is shown in the first row of Table 1. We did not evaluate the expansion for the other 280 clusters, since the idea of our approach is that the user selects and works with only a subset of the clusters.

In order to include in our textology clusters related to protestors' demands, we performed a second learning iteration: In this case, the initial term class was set to be *{demanded, criticized}*. We run two iterations of the cluster expansion, this time it was done in the context of the class *protestors*. In this way, this algorithm learnt 16 new terms. It turned out ten of these were correct, thus the accuracy of the algorithm turned out to be **63 %**.

We run the context learning algorithm, this time using two clusters *demanded* and *protestors*. The output of our algorithm were clusters, co-occurring both with the words from *protestors* and *demands*. While many clusters were relevant to the topic *protestors' demands*, we chose 12 which were the most common reasons for demands and criticisms of the protestors. These clusters were *{deal}, {money, funds}, {law, bill}, {school}, {prosecution}, {media}, {health}, {job}, {cut}, {tax}, {pay, paid, paying}, {bank}*. We also performed an additional run with the clusters *demanded* and *protest*. We added one additional cluster to our textology, namely *wage, salary, salaries, wages, allowance, allowances*.

The clusters were expanded in one iteration, using the cluster expansion algorithm in the context of the *protestors* cluster. The accuracy of the expansion is shown in the second row of Table 1. As we mentioned earlier, the threshold of the expansion in context is lower than the expansion without context, due to the presence of a reference cluster as a constraint. Because of these differences, the accuracies of both versions are not comparable.

Finally, we ran the link extraction algorithm which found arcs connecting clusters generated so far. The obtained textology was used for generation of several grammars in the domain of protests and civil disorders.

5.2 Generating Grammars

We developed three grammars using the grammar generation algorithm described in Sect. 4.2. These grammars aimed at extracting violence-related events, road blocking and expressions of economic demands during a protest. In case of these three grammars, we experimented with one-edge paths.

Regarding the grammar about violence related to protests, we selected the arcs between 35 pairs of clusters from the set *{{disperse, quell}, {police, gardai}, {clashes, fighting,…}, {threw, hurled,…}, {tear gas, rubber bullets,…}, {fired, shooting}, {clash, clashed}, {injured, hurt, …}, {stones, grenades,…}, {rallies, demonstrations,…}, {protesting, protest}, {forces, troops}}*. Regarding the grammars about road blocking, we selected the arcs between the clusters *{blocked, blocking}* and *{road, roads, highway, highways}*. Finally, in case of the grammar about economic demands we used the arcs between 11 pairs from the set of clusters: *{{protesting, protest}, {demanded}, {protestors}, {job}, {cuts}, {tax}, {pay, paid, paying}, {jobs, employment}, {salaries, allowances}, {rallies, demonstrations}, {wage, salary,…}}*.

For each grammar, we built a small test set which contained both *positive* sentences which express the events of interest and *negative* sentences, which though being selected from the domain of protests, do not express events of interest. The corpus concerning protest-related violence contained 24 positive and 11 negative sentences; the corpus about road blocking contained 21 positive and 19 negative sentences; the corpus about economic demands contained 15 positive and 15 negative sentences.

We run each grammar on the corresponding test set and considered that a sentence is selected when at least one grammar rule matches a substring of it. In this way, we evaluated the task for selecting sentences describing events. We measured the precision, recall and F-measure for each grammar. The results are shown in Table 2. Precision is 100 % due to the unambiguous nature of the generated rules. Obviously, recall can be improved.

Table 2 Accuracy of detecting event sentences

Event type	Precision (%)	Recall (%)	F-measure (%)
Violence	100	46	63
Road blocking	100	33	50
Economic demands	100	40	57

6 Conclusion

We presented a knowledge representation model called *textology*. The model is less expressive than formal ontologies, but nevertheless maps semantic classes and their syntagmatic relations. Therefore, textologies provide an expressive format of an intermediate level which though being less expressive than ontologies, allow for tasks like grammar generation. Textologies may also be considered for upgrading them into full-fledged ontologies. We presented several algorithms for building textologies and generating context-free grammars based thereon. Our preliminary experiments show encouraging results.

In future work, we aim at experiments with other algorithms of knowledge acquisition and grammar learning. Additionally, we aim at NLP applications based on our model.

References

1. Bollegala D, Matsuo Y, Ishizuka M (2008) A co-occurrence graph-based approach for personal name alias extraction from anchor texts. In: International joint conference on natural language processing (Ijcnlp), pp 865–870
2. Bordag S, Heyer G, Quasthoff U (2003) Small worlds of concepts and other principles of semantic search. In: Böhme T, Heyer G, Unger H (eds) Iics, vol 2877. Springer, New York, pp 10–19. Retrieved from http://dblp.uni-trier.de/db/conf/iics/iics2003.html#BordagHQ03
3. Buitelaar P, Cimiano P (eds) (2008) Ontology learning and population. Bridging the gap between text and knowledge. Springer, Berlin
4. Carlson A, Betteridge J, Kisiel B, Settles B, Estevam R, Hruschka J, Mitchell T (2010) Toward an architecture for never-ending language learning. In: Proceedings of the twenty-fourth AAAI conference on Artificial Intelligence (AAAI-10), Atlanta, GA, pp 1306–1313
5. Costa ME, Bonomo F, Sigman M (2009) Scale-invariant transition probabilities in free word association trajectories. Front Integr Neurosci 3:17
6. Dagan I, Glickman O, Magnini B (2006) The pascal recognising textual entailment challenge. In: Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising tectual entailment. Springer, New York, pp 177–190
7. Drumond L, Girardi G (2008) A survey of ontology learning procedures. In: The 3rd workshop on ontologies and their applications, Salvador, Brasil, pp 13–25
8. Feldman R, Sanger J (2007) The text mining handbook. Advanced approaches in analyzing unstructured data. Cambridge University Press, Cambridge
9. Heyer G, Läuter M, Quasthoff U, Wittig T Wolff C (2001) Learning relations using collocations. In: Maedche A, Staab S, Nedellec C, Hovy EH (eds) Workshop on ontology learning, vol 38. CEUR-WS.org. Retrieved from http://dblp.uni-trier.de/db/conf/ijcai/ijcai2001ol.html#HeyerLQWW01
10. Klebanov BB, Flor M (2013, August). Word association profiles and their use for automated scoring of essays. In: Proceedings of the annual meeting of the association for computational linguistics, Sofia, Bulgaria
11. Matsuo Y, Ishizuka M (2004) Keyword extraction from a single document using word co-occurrence statistical information. Int J Artif Intell Tools 13(01):157–169
12. Navigli R, Velardi P (2008) From glossaries to ontologies: extracting semantic structure from textual definitions. In: Ontology learning and population. Bridging the gap between text and knowledge. Springer, Berlin, pp 71–87

13. Ohsawa Y, Soma H, Matsuo Y, Matsumura N, Usui M (2002) Featuring web communities based on word co-occurrence structure of communications: 736. In: Proceedings of the 11th international conference on world wide web, p 742
14. Pantel P, Lin D (2002) Discovering word senses from text. In: Proceedings of ACM SIGKDD conference on knowledge discovery and data mining, Edmonton, pp 613–619
15. Riloff E, Jones R (2002) Learning dictionaries for information extraction by multi-level bootstrapping. In: Proceedings of the sixteenth national conference on Artificial Intelligence (AAAI 99), Orlando, FL, pp 474–479
16. Tanev H, Zavarella V, Kabadjov M, Piskorski J, Atkinson M, Steinberger R (2009) Exploiting machine learning techniques to build an event extraction system for Portuguese and Spanish. Linguamatica 2:55–66
17. Völker J, Haase, P Hitzler P (2008) Learning expressive ontologies. In: Proceedings of the 2008 conference on ontology learning and population: bridging the gap between text and knowledge. IOS Press, Amsterdam, pp 45–69
18. Wikipedia: Cyc. (2014) Retrieved from http://en.wikipedia.org/wiki/Cyc
19. Zortea M, Menegola B, Villavicencio A, Salles JFD (2014) Graph analysis of semantic word association among children, adults, and the elderly. Psicologia: Reflexão e Crítica 27(1):90–99

Simple, Fast and Accurate Taxonomy Learning

Zornitsa Kozareva

Abstract Although many algorithms have been developed to extract lexical resources, few organize the mined terms into taxonomies. We propose (1) a semi-supervised algorithm that uses a root term, a seed example and lexico-syntactic patterns to learn automatically from the Web hyponyms and hypernyms subordinated to the root; (2) a Web based concept positioning test to validate the learned terms and is-a relations; (3) a graph algorithm that induces from scratch the taxonomy structure of all terms and (4) a pattern-based procedure for enriching the learned taxonomies with verb-based relations. We conduct an exhaustive empirical evaluations on four different domains and show that our algorithm quickly and accurately acquires and taxonomies the knowledge. We conduct comparative studies against WordNet and existing knowledge repositories and show that our algorithm finds many additional terms and relations missing from these resources. We conduct an evaluation against other taxonomization algorithms and show how our algorithm can further enrich the taxonomies with verb-based relations.

1 Introduction

A variety of NLP tasks, including question answering [29] and textual entailment [10, 45] rely on semantic knowledge derived from term taxonomies and thesauri such as WordNet. However, the coverage of WordNet is still limited in many regions (even well-studied ones such as the concepts and instances below Animals and People), as noted by researchers who automatically learn semantic classes [23, 34]. This happens because WordNet and most other existing taxonomies are manually created, which makes them difficult to maintain in rapidly changing domains and makes them hard to build with consistency. To surmount these problems, it would be advantageous to have an automatic procedure that can not only augment existing resources but can also produce taxonomies for existing and new domains and tasks starting from scratch.Various approaches have been developed to learn taxonomies

Z. Kozareva (✉)
Computer Science Department, Marina del Rey, CA, USA
e-mail: zornitsa@kozareva.com

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural
Language Processing, DOI 10.1007/978-3-319-12655-5_3

[41, 46]. Among the most common approach is to start with a good ontology and then to try to position the missing concepts into it [43]. Others [49] combine heterogenous features like context, co-occurrence, and surface patterns to produce a more-inclusive inclusion ranking formula. The obtained results are promising, but the problem of how to organize the gathered knowledge when there is no initial taxonomy, or when the initial taxonomy is impoverished, still remains.

The major problem with taxonomy induction from scratch is that term positioning is not trivial, because it is difficult to discover whether concepts are unrelated, subordinated, or parallel to each other. In this paper, we address the following questions: *How can one learn and induce the taxonomic organization of terms in a given domain starting from scratch?* and *How can one enrich the induced taxonomy with verb-based relations?*.

The main contributions of our research are:

- An automatic procedure for accurately extracting hyponyms and hypernyms for a given domain of interest.
- A ranking mechanism for validating the learned terms and is-a relations.
- A graph-based approach for inducing the taxonomic organization of the extracted terms starting from scratch.
- A pattern-based approach for enriching the induced taxonomies with verb-based relations.
- An experiment on reconstructing WordNet's taxonomy for given domains.
- An exhaustive evaluation on four different domains and conducting comparative studies against existing knowledge bases, resources and algorithms.

The rest of the paper is organized as follows. Next, we present related work. Section 3 describes the hyponym and hypernym extraction mechanisms which are followed in Sect. 4 by term taxonomization and Sect. 5 by taxonomy enrichment with verb-based relations. In Sect. 6 we describe the data collection and evaluation processes. We conclude in Sect. 7.

2 Related Work

Learning taxonomies involves multiple stages from term extraction and term taxonomization to taxonomy enrichment with verb-based relations. Term extraction has been relatively well understood. Researchers have developed a wide variety of approaches that learn hyponym and hypernym terms in semi-supervised or unsupervised fashion. Vast majority of approaches use either clustering or lexico-syntactic patterns to extract knowledge from text. Clustering approaches [5, 16, 25, 26] are fully unsupervised and discover relations that are not directly expressed in text. However, such methods find it challenging to assign labels to the clusters and sometimes may produce groupings that contain more or less general terms. In contrast, pattern-based approaches [11] extract information with high accuracy, but they require a set of seeds and surface patterns to initiate the learning process.

These methods are used to extract hyponyms [6, 22, 33, 38], hypernyms [23, 39], encyclopedic facts [1, 4, 35, 44], concept lists [17] and relations between terms [9, 12, 32]. Others focus on learning the selectional restrictions of semantic relations [21, 40] and their WordNet ontologizing [34].

The second stage on full fledge taxonomy induction is a much harder task. Early approaches on acquiring taxonomies from machine reading dictionaries include [2, 15, 36, 37, 47]. The most common taxonomy learning approaches start with a reasonably complete taxonomy and then insert one at a time the newly learned terms [13, 33, 43, 46, 49]. Others [43] guide the incremental approach by maximizing the conditional probability over a set of relations, while Yang and Callan [49] introduce a taxonomy induction framework which combines the power of surface patterns and clustering through combining numerous heterogeneous features. Navigli et al. [31] used graph weighting and pruning to taxonomize the terms.

Despite the large body of work, still, one would like to organize the extracted terms and relation into a taxonomic structure starting from scratch (i.e. without using an initial taxonomic structure). We propose an approach that bridges the gap between the term extraction algorithms which focus mainly on knowledge extraction but do not taxonomize and those that accept a new term and seek to enrich an already existing taxonomy. Our goals it to perform all stages at a time by starting with the term extraction and term organization, and finalizing with taxonomy enrichment with verb-based relations. Achieving this goal could provide the research community with a fast, simple and accurate automatic procedure for term taxonomization of domains that currently do not have manually created taxonomies. In the next sections we describe the term extraction, term taxonomization and taxonomy enrichment with verb-based relations procedures. The algorithms are followed by in depth evaluation and analysis of the obtained results.

3 Taxonomy Term Extraction

This section describes our semi-supervised procedure for learning sets of terms related to a domain of interest. At a higher level the knowledge harvesting algorithm has two phases, one related to *hyponym* extraction and another one related to *hypernym* extraction. Each extraction step is accompanied by a filtering procedure, which guarantees that only highly accurate terms will be included in the taxonomy induction stage.

Figure 1 shows the general framework of the knowledge extraction algorithm. As input the user provides seed examples representative of the domain of interest, as output the algorithm produces high-quality hyponym and hypernym terms of the domain interlinked with is-a relations.

At the heart of the term extraction procedures are doubly-anchored lexico-syntactic patterns (DAP) [11, 22], which produce diverse, reliable and accurate extractions. DAP has a recursive nature in which the newly learned candidate terms on the * position are used as seeds in the subsequent iteration [22].

```
1.  Given:
    a DAP hyponym pattern $P_i$={concept such as seed and *}
    a DAP$^{-}$1 hypernym pattern $P_c$={* such as $term_1$ and $term_2$}
    a root concept root
    a term called seed for $P_i$
2.  build a query using $P_i$
3.  submit $P_i$ to Yahoo! or other search engine
4.  extract terms occupying the * position
5.  take terms from step 4. and go to step 2.
6.  repeat steps 2–5 until no new terms are found
7.  rank terms by outDegree
8.  all terms with outDegree>0, build a query using $P_c$
9.  submit $P_c$ to Yahoo! or other search engine
10. extract concepts (hypernyms) occupying the * position
11. rank concepts by inDegree
12. for $\forall$ terms with inDegree>1, check subordination to the root with CPT
13. use concepts passing CPT from step 12. as temporary root and go to step 2.
14. repeat steps 2–13 until the user desires
```

Fig. 1 Hyponym–hypernym knowledge harvesting framework

This characteristic eliminates the need of humans to constantly provide seed terms [22]. Next, we describe the hyponym and hypernym extraction phases.

3.1 Hyponym Extraction and Filtering

The hyponym extraction phase is responsible for learning terms located at the lowest level of the domain taxonomy. For example, for *Animals* such terms are *lion*, *ant*, *elephant* among others. To learn the hyponyms, our algorithm takes as input from a user a root concept representative of the domain say *Animal* and a seed example say *lion*. The root and seed terms are fed into the DAP pattern "⟨*semantic class*⟩ *such as* ⟨*seed*⟩ *and* *" in order to learn from the * placeholder new terms representative of the domain. Each DAP pattern is treated as a web query and is submitted to Yahoo! Boss search API. All snippets matching the query are retrieved and part-of-speech tagged [42]. Terms (nouns and proper names) found on the * position are extracted, but only those that were previously unexplored are used as seeds in the next iteration. The hyponym term-extraction process terminates when all terms are explored. The algorithm is implemented as a breadth-first search.

Knowledge harvesting algorithms could extract erroneous terms, therefore we use graph metrics to re-rank and filter out the extracted terms. We feed the terms into a directed graph $G = (V, E)$, where each vertex $v \in V$ is a candidate term for the *semantic class* and each edge $(u, v) \in E$ indicates that the term v is extracted from the term u. A term u is ranked by

$$outDegree(u) = \frac{\sum_{\forall (u,v) \in E} (u, v)}{|V| - 1} \tag{1}$$

which represents all outgoing edges from u normalized by the total number of nodes in the graph. The intuition is that in a very large corpus such as the Web correct terms should frequently discover multiple different terms with the DAP pattern.

3.2 Hypernym Extraction and Filtering

The hypernym extraction phase is responsible for learning terms located above the lowest level and below the root level of the domain taxonomy. For example, for *Animals* such terms are *mammals*, *predators*, *herbivores* among others. To learn hypernyms, our algorithm takes as input all ⟨X,Y⟩ term pairs collected during the hyponym extraction phase and feeds them into the inverse DAP^{-1} pattern *"* such as ⟨X⟩ and ⟨Y⟩"* in order to learn from the * placeholder new hypernyms of the domain. Each DAP^{-1} pattern is treated as a web query and is submitted as to Yahoo! Boss API. For example, if the term *tiger* is learned from the DAP pattern *animals such as lions and ⟨Y⟩*, the pair *<lion,tiger>* is used to form the DAP^{-1} query * *such as lion and tiger*. This pattern extracts the hypernyms *mammals*, *carnivores* among others. The hypernym extraction phase terminates when all hyponyms have been exhaustively searched.

Once the hypernym phase terminates, all candidate hypernym terms are re-ranked in order to eliminate erroneous examples such as *toys*, *ones*. All hypernym–hyponym triples are fed into a bipartite graph $G' = (V', E')$ with a set of vertices V_{sup} representing the hypernyms and a set of vertices V_p corresponding to the ⟨X,Y⟩ hyponym term pair that extracted the hypernym. An edge $e'(u', v') \in E'$, where $u' \in V_p$ and $v' \in V_{sup}$ indicates that the pair ⟨X,Y⟩ denoted as u' extracted the hypernym v'. A vertex $v' \in V_{sup}$ is ranked by

$$inDegree(v') = \frac{\sum_{\forall (u',v') \in E'} (u', v')}{|V'| - 1} \tag{2}$$

which represents the sum of all incoming edges to the hypernym node v' from all term pairs u'. Intuitively, our confidence in a correct hypernym increases when it is discovered by multiple different hyponym pairs.

3.3 Concept Positioning Test

The hypernym graph ranking mechanism eliminates erroneously extracted terms, however, it does not determine whether a term is more or less general than the initial root concept. For instance, for *Animals* the concepts *species* is too general as it also relates to non-animal terms such as *Plants*. Since we are interested in learning terms under a specific root term (i.e. *Animals*), we need a test for domain membership which eliminates general concepts like *species*.

Following [14] a simple and yet powerful test for domain membership is the *Concept Positioning Test (CPT)*. It keeps only those terms that are located below the initial root term by testing each term with the following two queries:

(a) *RootConcept such as Concept*
(b) *Concept such as RootConcept*

where *Concept* is the extracted hypernym and *RootConcept* is the starting root term. If the system returns more Web hits for (a) than (b), this indicates that the *Concept* passes the CPT test and it is located below the root. If the system returns more Web hits for (b) than (a) this means that the concept is more general than the root and it fails the CPT test and must be excluded from the domain.

All extracted hypernyms that pass the CPT test are fed back into the hyponym extraction phase. The hypernym become a root concept and the seeds are all hyponyms that extracted the hypernym. A new cycle of hyponym extraction begins and it leads to learning new hyponym terms. The newly extracted hyponyms are passed to the hypernym extraction phase and CPT testing. This hyponym–hypernym extraction procedure repeats until all terms are extracted or until a manually set threshold is reached. In our case we ran the bootstrapping processes for ten iterations.

4 Taxonomy Induction

The hyponym–hypernym extraction phase is followed by a taxonomy induction phase in which all terms that passed the ranking criteria are hierarchically organized with respect to each other. For the purpose, we collect statistics from lexicon-syntactic patterns and use graph algorithms to induce the taxonomy structure.

4.1 Positioning Intermediate Concepts

Once the knowledge extraction and domain filtering phase terminates, we obtain the is-a relations between the root and low-level terms, as well as the is-a relations between the low-level and intermediate-level terms. However, the only information that is missing is the is-a relationship between the intermediate-level concepts. For example, the knowledge harvesting algorithm does not provide information of the hierarchical organization of concepts like *mammals*, *carnivores*, *vertebrates*, *felines*, *chordates* among others. Since the CPT test is an extremely reliable mechanism for the positioning of hypernyms with respect to the root, we decided to use the same procedure for the positioning the intermediate-level concepts. To gain more evidence from the Web, we use multiple surface patterns of the form: "X *such as* Y", "X *are* Y *that*", "X *including* Y", "X *like* Y", "*such* X *as* Y", where the *X* and *Y* corresponds to intermediate-level concepts. For instance, if we want to position the intermediate concepts *chordates* and *vertebrates* with respect to each other, we issue the CPT queries of the form: (a) *chordates such as vertebrates* and (b) *vertebrates such as chordates*. We record the counts of each pattern and estimate whether (a) returns more hits than (b). If this is the case, then *chordates* subsumes (or is broader than) *vertebrates*, otherwise *vertebrates* subsumes *chordates*.

4.2 Graph-Based Concept Reordering

Figure 2 shows the organization of the root, low-level and intermediate-level concepts according to the concept positioning test, which sometimes cannot determine the taxonomic organization between all concepts. For example, there is no is-a link between *felines* and *chordates* or between *felines* and *vertebrates*. This is due to the fact that humans tend to use examples which are from the same or closely related taxonomic levels. Therefore, CPT cannot find evidence for *chordates→felines* but it can find for *mammals→felines*.

After the concept positioning procedure has explored the positioning of all intermediate concept pairs, we observed two phenomena: (1) direct links between

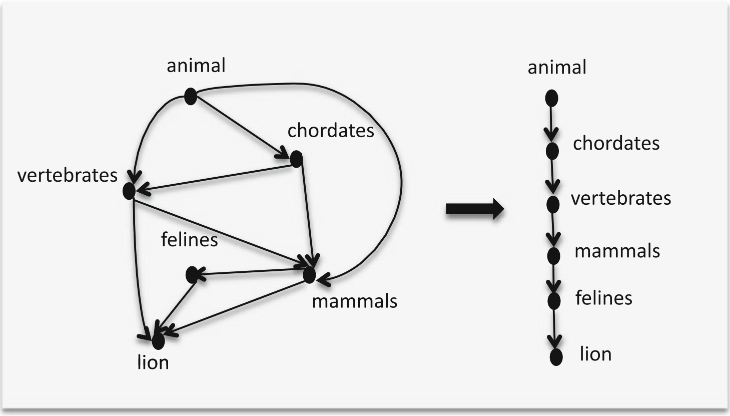


Fig. 2 Concept positioning procedure and induced taxonomy

some concepts are missing and (2) multiple paths can be taken to reach from one concept to another. To surmount these problems, we build a directed graph $G'' = (V'', E'')$ in which for a given a set of concepts (root, low, intermediate level ones), the objective is to find the longest path in the graph. In our case, the longest path would represent the taxonomic organization of the concepts as shown on the right side of Fig. 2.

In the graph G'', the nodes $V'' = \{t_1, t_2, t_3, .., t_n, r\}$ represent the harvested terms (root, low, intermediate level), the edge $(t_i, t_j) \in E''$ indicates the is-a relatedness of t_i and t_j, and the direction $t_i \rightarrow t_j$ corresponds to the term subordination according to the CPT test. If present, we eliminate all cycles in the graph. For that we use the CPT values of the terms and we use those whose weight is higher. If both terms have equal CPT values for (a) and (b), then we randomly select whether (a) or (b) subordination should remain. For each low-level term, we extract all hypernyms and is-a relations and use them to build a graph. On the top, we position the node with no predecessors p (e.g. *animal*) and at the bottom, the node with no successor s (e.g. terms like *lion*, *tiger*, *puma*). The directed graph is represented as an adjacency matrix $A = [a_{i,j}]$, where $a_{i,j}$ is 1 if (t_i, t_j) is an edge of G'', and 0. To find the longest path between p and s pair, we find all possible paths between p with s, and select the longest one among them. We use this path to represent the taxonomic organization of all concepts located between p and s. Once the taxonomization of a given low-level concept and its hypernyms terminates, we apply the same procedure to the next low-level term and its hypernyms.

5 Taxonomy Enrichment with Verb-Based Relations

While the majority of the research focuses on term extraction and taxonomy induction, less work has been done on taxonomy enrichment with verb-based relations, which aims at interlinking terms with deeper information than the is-a relations. Next, we describe how the same pattern-based bootstrapping paradigm can be used to learn verb-based relations and enrich the induced taxonomies.

5.1 Problem Formulation

We define our task as given a term from the induced taxonomy, a relation expressed by a verb and a set of prepositions: (1) learn in bootstrapping fashion new verb-based relations associated with the term; (2) form triples of the term, the harvested verbs and the initial set of prepositions to learn additional verb–preposition relations and argument fillers [18].

Figure 3 shows an example for the input term *terrorists*, the verb relation *bomb* and the recursive pattern "*terrorists bomb and* *". The verb-relation learning algorithm learns on the * position new verbs like *kill, murder, threaten, burn,*

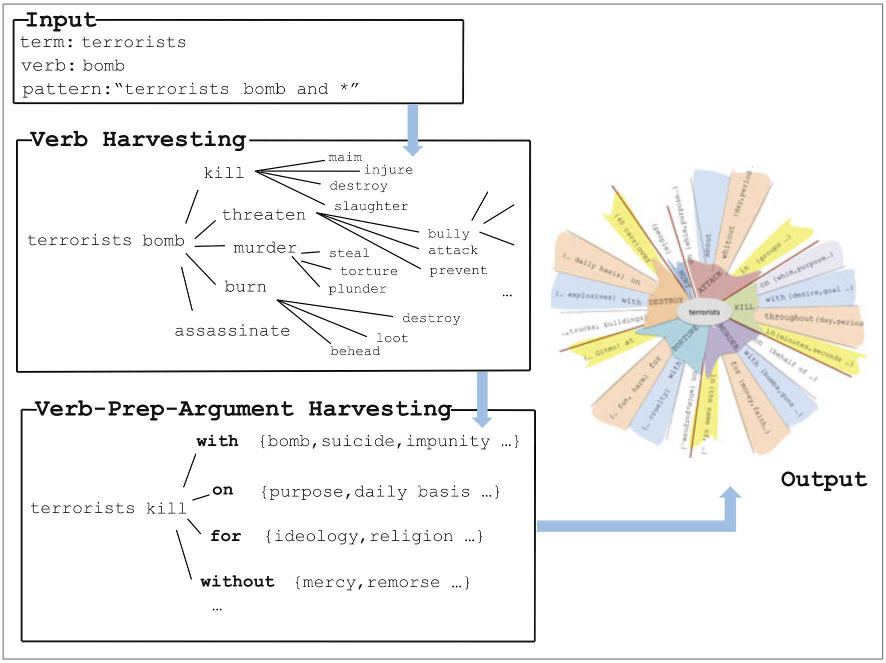


Fig. 3 Verb-based relation learning

assassinate. This phase is known as *verb* extraction. Then each learned verb is used to form triples of the type *term-verb–preposition* to learn new verb–preposition relations and argument fillers. For instance, "*terrorists kill with* *" extracts arguments like {*bombs, suicide, impunity*}. We denote this phase as *verb–preposition* extraction. The learned relations and arguments are ranked to filter out erroneous extractions. The output of the relation learning procedure is triples of the kind "*terrorists kill people*", "*terrorists kill on purpose*", "*terrorists bomb buildings*" among others. Next we describe each phase in details.

5.2 *Learning Verb Relations*

We adapt the DAP pattern for verb extraction in the following way "<*seed-term*> <*seed-verb*> and *", where <*seed-term*> is the input term, <*seed-verb*> is a seed relation expressed through a verb and * indicates the position on which new verbs will be extracted. All patterns are submitted as queries on the web and the retrieved snippets are collected. The algorithm extracts on the $*$ position verbs which are used as seeds in the next iteration. Similarly to the hyponym extraction phase, harvesting terminates when there are no more verbs to be explored. The extractions are fed into a directed graph $G = (V, E)$ with node $v \in V$ corresponding to verb candidate and edges $(u, v) \in E$ connecting two verbs. Each node u is ranked as $u = \sum_{\forall (u,v) \in E}(u, v)$. Confidence in u increases when u extracts more verbs.

5.3 Learning Verb–Preposition Relations

To learn new relations and argument fillers, we use the extracted verbs from the previous stage and pair them up with a set of 17 prepositions from the SemEval 2007 preposition disambiguation task [27]. The algorithm uses the pattern "<*seed-term*> <*verb*> <*prep*> *", where <*seed-term*> is the term for which we want to learn verb-based relations, <*verb*> are the leaned verbs from the previous phase and * is the position of the arguments, which can be nouns or proper names. Given the relation *kill* for the term *terrorists*, new relations like *terrorists kill on*, *terrorists kill with* and *terrorists kill for* are formed. All terms are fed in a bipartite graph $G' = (V', E')$ with one set of nodes representing the verbs and verb–prepositions V, and another set representing the arguments A. An edge $e'(v, a) \in E'$ between $v \in V$ and $a \in A$ shows that the verb (or verb–prep) v extracted the argument a. An argument is ranked as $a = \sum_{\forall (v,a) \in E'}(v, a)$. Confidence in a increases when a is extracted multiple times by different verbs.

6 Data Collection and Experimental Set Up

It is impossible to collect and report on results for all terms and domains. Therefore, to evaluate the performance of our knowledge harvesting, taxonomy induction and taxonomy enrichment algorithms on four domains: *Animals*, *People*, *Vehicles*, and *Plants*. We choose these domains based on their diverse nature and characteristics, as well as the fact that they have taxonomic structures that are well-represented in WordNet [19, 23, 24].

We have instantiated the knowledge harvesting procedure with the following seed terms: *lions* for *Animals*, *Madonna* for *People*, *cars* for *Vehicles*, and *cucumbers* for *Plants*. To collect the data, we have submitted the DAP patterns as web queries to Yahoo!, retrieved the top 1,000 web snippets per query, and kept only the unique ones. In total, we have collected 10 GB of text snippets. We ran the hyponym extraction algorithm until complete exhaustion, while the hyponym–hypernym replacement steps for ten iterations. The harvested data and the gold standard data used for our taxonomization evaluation can be downloaded here.[1]

In the next subsections we describe the obtained results from five different experiments we have conducted. In Experiment 1, we evaluate the performance

[1] http://kozareva.com/~kozareva/data/kozareva_taxonomy_data.zip.

of DAP for hyponym learning, in Experiment 2, we evaluate the performance of DAP^{-1} for hypernym learning, in Experiment 3, we evaluate the generated is-a relations between the concepts, in Experiment 4, we evaluate the induced taxonomic structures and in Experiment 5, we evaluate the learned verb-based relations. For each experiment we conducted a human-based evaluation and a comparative study against WordNet version 3.0. Initially, we also wanted to compare our results to knowledge bases that have been extracted in a similar way (i.e., through pattern application over unstructured text), however, it is not always possible to perform a complete comparison, because either researchers have not fully explored the same domains we have studied, or for those domains that overlap, the gold standard data was not available.

6.1 Experiment 1: Hyponym Extraction

This section describes the results of the hyponym extraction phase. In ten iterations the algorithm extracted 913 animal, 1,344 people, 1,262 plant and 1,425 vehicle terms that passed the *outDegree* filtering criteria.

Human Evaluation To evaluate the correctness of the extracted hyponyms we used two human judges, according to whom precision is 71 % for *Animals*, 95 % for *People*, 83 % for *Vehicles* and 90 % for *Plants*. The obtained results show that the hyponym step of the bootstrapping algorithm generates a large number of correct hyponyms of high quality.

WordNet Evaluation Table 1 shows the accuracy of the hyponym extraction according to WordNet (Pr_{WN}) and humans (Pr_H).

The precision for the *People* category is higher for humans, because they found that many of the correctly extracted terms were not present in WordNet. These results show that there is room for improvement in WordNet's term coverage.

Comparison Against Prior Work It is difficult to compare results against existing approaches, because either researchers have not explored the same domains or for those domains that overlap the generated data is not freely available. Still to the extend to which it is possible, we compare performance with the method of Kozareva et al. [22], which outperforms existing state-of-the-art systems [6, 33].

The approach of Kozareva et al. [22] corresponds to the first step of our bootstrapping process. The difference between the current algorithm and those of Kozareva et al. [22] is in the hyponym–hypernym stage, which feeds on each iteration the newly learned intermediate-level concepts as roots for the DAP pattern

Table 1 WordNet hyponym evaluation

	Pr_{WN}	Pr_H	*NotInWN*
Animal	0.79	0.71	48
People	0.23	0.95	986

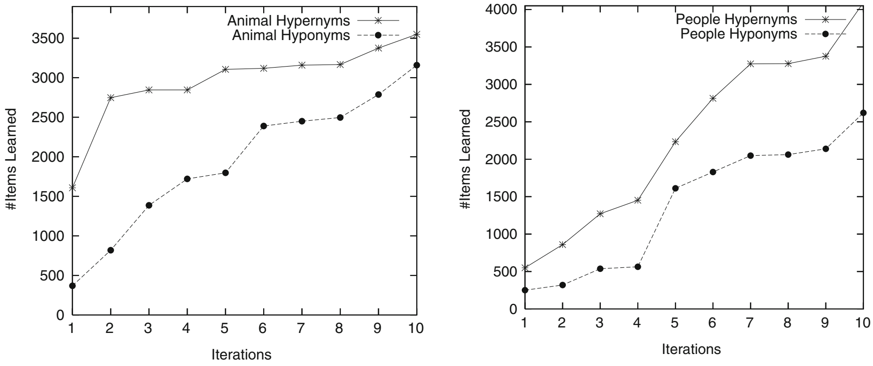


Fig. 4 Learning curves for animals and people

and instantiates the learning from the very beginning. Figure 4 shows the number of harvested terms for *Animals* and *People* for each one of the ten iterations.

Overall, the bootstrapping with intermediate concepts produced nearly five times as many low-level terms (hyponyms) compared to [22]. It is important to note that not only the recall of the extractions was improved, but also the high precision of the extractions was maintained.

6.2 Experiment 2: Hypernym Extraction

This section describes the results of the hypernym extraction phase. In ten iterations, the bootstrapping algorithm harvested 3,549 *Animal* and 4,094 *People* intermediate-level concepts. After the *inDegree* ranking was applied, we selected a random sample of intermediate-level concepts, which were used for evaluation. We conducted an extensive manual annotation following the guidelines of Hovy et al. [14] and Kozareva and Hovy [19].

Human Evaluation Kappa [8] inter-annotator agreement for *Animals* was $\kappa = 0.66$ and for *People* was $\kappa = 0.60$. The agreement scores are good enough to warrant the usage of the human judgments to estimate the precision of the algorithm, however they also showed that the task is not trivial. The accuracy for 437 *Animals* terms was 66 % and for 296 *People* terms was 85 %. After the CPT was applied only 187 *Animal* and 139 *People* terms passed. Precision increased to 84 % for *Animals* and 94 % for *People*. These results show that CPT is effective at removing undesirable general terms. Overall, the results demonstrate that our algorithm produced many high-quality intermediate concepts, with good precision.

WordNet Evaluation We also compared precision and presence of the learned hypernyms against WordNet. Table 2 shows the obtained results.

Table 2 WordNet intermediate concept evaluation

	Pr_{WN}	Pr_H	NotInWN
Animal	0.20 (88/437)	0.66 (288/437)	204
People	0.51 (152/296)	0.85 (251/296)	108

Table 3 WordNet taxonomic evaluation

ISA	Pr_{WN}	Pr_H	NotInWN
Animal	0.47 (912/1,940)	0.88 (1,716/1,940)	804
People	0.23 (318/908)	0.94 (857/908)	539

Of the learned intermediate-level concepts, WordNet contains 20 % of the *Animals* and 51 % of the *People* terms. This confirms that many of the concepts were also considered to be valuable taxonomic terms by the WordNet developers. However, our human annotators found 66 % of the *Animals* and 85 % of the *People* concepts to be correct, which suggests that the algorithm generated a substantial amount of additional concepts that could be used to enrich the taxonomy of WordNet.

6.3 Experiment 3: IS-A Taxonomic Relations

In addition to hyponyms and hypernyms, the algorithm also learns is-a relations. Given a relation *isa(X,Y)*, next we judge how often X is truly a subconcept of Y. For instance, *isa(goat, herbivore)* is correct, but *isa(goat, bird)* is not.

Human and WordNet Evaluations Table 3 shows the results for the is-a relations between all terms (intermediate and low-level ones). For each pair, we extracted the harvested links and determined whether the same links appear in WordNet. We also gave the same is-a relations to annotators.

The results show that the DAP patterns can accurately extract is-a relations. It is important to note that a substantial portion of the learned relations is not present in WordNet. For instance there are 804 *Animal* and 539 *People* links that are missing from WordNet.

6.4 Experiment 4: Reconstructing WordNet's Taxonomy

In this section, we evaluate the ability of our algorithm to induce the taxonomic structure of all learned terms. Since manual construction and evaluation of the harvested taxonomies is extremely challenging and difficult even for human experts, we decided to evaluate the performance of our algorithm by reconstructing WordNet's *Animals*, *Plants* and *Vehicles* taxonomies. We did not evaluate the taxonomy for *People*, because most of the learned terms are missing from WordNet.

Table 4 Data for WordNet reconstruction

	Animals	Plants	Vehicles
#terms	684	554	140
#is-a	4,327	2,294	412
Average depth	6.23	4.12	3.91
Max depth	12	8	7
Min depth	1	1	1

For each domain we selected the terms which were harvested by our algorithm and also present in WordNet. For each term and root concept (*Animal*, *Plant* or *Vehicle*) we retrieved all concepts located on the path between the two terms and used this information to evaluate our approach. Being able to reconstruct WordNet's taxonomy for these concepts is equivalent to evaluating the performance of our taxonomy induction approach.

Table 4 summarizes the characteristics of the taxonomies. For each domain, we show the total number of terms that must be organized, and the total number of is-a relations that must be induced.

Among the three domains we have used for our evaluation, the *Animals* one is the most complex and has the richest taxonomic structure. The maximum number of levels that must be inferred is 11, the minimum is 1 and the average taxonomic depth is 6.2. In total there are three low-level concepts (*longhorns*, *gaur* and *bullock*) with maximum depth, 20 terms (low-level and intermediate concepts) with minimum depth and 98 low-level terms (*wombat*, *viper*, *rat*, *limpkin*) with depth 6. *Plants* is also a very challenging domain, because it contains a mixture of scientific and general terms such as *magnoliopsida* and *flowering plant*.

Taxonomy Induction Evaluation To evaluate the performance of our taxonomy induction approach, we use the following measures:

$$Precision = \frac{\#is - a\ found\ in\ WordNet\ and\ by\ system}{\#is - a\ found\ by\ system} \tag{3}$$

$$Recall = \frac{\#is - a\ found\ in\ WordNet\ and\ by\ system}{\#is - a\ found\ in\ WordNet} \tag{4}$$

Table 5 shows results for the taxonomy induction of the *Vehicles* domain using different concept positioning patterns. The most productive patterns are: "X *are* Y *that*" and "X *including* Y", however the highest yield is obtained when we combine the evidence from all patterns (i.e. when we sum the retrieved Web counts from all patterns). Table 6 shows results for the taxonomization of the *Animals*, *Plants*, and *Vehicles* domains.

Figure 5 shows an example of our taxonomy induction algorithm for some low-level terms like *vipers*, *rats*, *wombats*, *ducks*, *emus*, *moths*, and *penguins* and their hypernyms. The obtained results are very encouraging given the fact that we started the taxonomy construction entirely from scratch (i.e. without the usage of a skeleton

Table 5 Evaluation of the induced vehicle taxonomy

Vehicles	Precision	Recall
X such as Y	0.99 (174/175)	0.42 (174/410)
X are Y that	0.99 (206/208)	0.50 (206/410)
X including Y	0.96 (165/171)	0.40 (165/410)
X like Y	0.96 (137/142)	0.33 (137/410)
such X as Y	0.98 (44/45)	0.11 (44/410)
All patterns	0.99 (246/249)	0.60 (246/410)

Table 6 Evaluation of the induced taxonomies

	Precision	Recall
Animals	0.98 (1643/1688)	0.38 (1643/4327)
Plants	0.97 (905/931)	0.39 (905/2294)
Vehicles	0.99 (246/249)	0.60 (246/ 410)

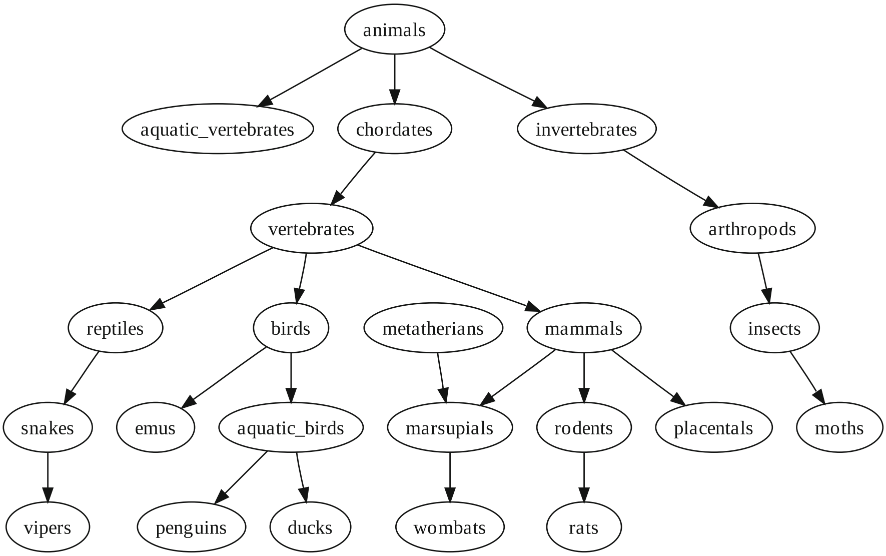


Fig. 5 An example of the induced taxonomy of our algorithm for some animal terms

structure of any existing taxonomy). The precision of the taxonomization approach is very robust. However, recall must be further improved since not all concepts were found with the lexico-syntactic patterns.

Comparison with Existing Approaches We compare the performance of our pattern-based taxonomy induction algorithm with another contemporary graph-based taxonomization algorithm developed by Roberto et al. [41]. Since they have used all of our harvested terms, is-a relations and gold standard data to evaluate the performance of their taxonomization algorithm, this is making it easy for us to conduct comparative studies and hopefully it would also encourage other

Table 7 Comparative evaluation of our taxonomy induction algorithm and the graph-based taxonomy induction algorithm of [41]

	Our approach		Navigli et al. [41]	
	Precision	Recall	Precision	Recall
Animals	0.98 (1643/1688)	0.38 (1643/4327)	0.97 (1638/1688)	0.44 (1890/4327)
Plants	0.97 (905/931)	0.39 (905/2294)	0.97 (905/931)	0.38 (879/2294)
Vehicles	0.99 (246/249)	0.60 (246/410)	0.91 (226/249)	0.49 (200/410)

researchers working on taxonomy induction to use our knowledge harvested data as a reference point for comparison.

To briefly summarize, our algorithm used CPT to find term relatedness, while [41] used graph trimming and edge weighting procedure. We induce the taxonomy using the longest path in the graph, while [41] used a Chu-Liu/Edmonds algorithm to find the optimal branching and then they applied pruning recovery to induce the final taxonomy.

Table 7 shows the obtained results of the two algorithms for the same number of terms, is-a relations and taxonomies. Our pattern-based taxonomy induction outperforms [41] for two out of the three domains. We obtained lower recall only for the *Animals* domain.

In conclusion, we can say that the beauty of our work lies not only in the simplicity of our knowledge harvesting and taxonomization algorithm, which is making it easy to implement and use by anyone, but also in our effort to create and freely distribute a taxonomization data set, which can be used as an evaluation benchmark by other unsupervised taxonomy induction algorithms.

6.5 Experiment 5: Taxonomy Verb-Based Enrichment

In this section, we evaluate the ability of our algorithm to enrich the induced taxonomy with verb-based relations.

Data Collection For the empirical evaluation, we have randomly selected 16 terms from the *People* and *Animals* domains. Table 8 shows the terms and seed verbs used to initiate the verb-based relation learning process, and summarizes the obtained results and the total number of iterations which were run to extract the verbs. *#Verbs Unique* shows the number of unique verbs after merging expressions like (*were killed*, *are killed*, *killed*). For each domain, we also show the total number of verbs used to initiate the harvesting process and the total number of learned information.

Human-Based Evaluation We select 100 verb relations and argument fillers for each term and use two annotators to mark as correct relations like "*ants bite*" and as incorrect relations like "*ants discuss*". We compute *Accuracy* as the number of *Correct* terms, divided by the total number of terms used in the annotation.

Table 8 Tested terms for verb-based relation learning and extracted information

Seed term	Seed verb	#Verbs learned	#Verbs unique	#Iter.	#Args. learned	#Args. with $a > 5$
People						
Authorities	Say	3,049	1,805	14	7,284	151
Bombers	Explode	265	224	19	9,097	344
Killers	Kill	178	163	14	6,906	217
Soldiers	Die	4,588	2,533	10	34,330	1,010
Terrorists	Kill	1,401	941	10	13,698	468
Victims	Suffer	1,861	1,263	13	21,982	767
Animals						
Ants	Eat	827	607	12	25,046	753
Birds	Eat	3,623	2,064	8	62,031	1,465
Dinosaurs	Eat	544	386	11	11,013	345
Jellyfish	Eat	12	11	4	1,120	20
Lice	Eat	42	42	8	3,330	131
Mammals	Eat	338	272	10	14,224	527
Otters	Eat	190	159	8	5,051	159
Sharks	Eat	697	500	12	16,942	598
Slugs	Eat	60	60	11	5,223	89
Vultures	Eat	36	36	5	2,757	67

Table 9 Accuracy of the extracted verb-based relations

	Accuracy verbs			Accuracy arguments		
Term	@10	@50	@100	@10	@50	@100
People						
Authorities	1	1	1	1	1	0.90
Soldiers	1	1	1	1	1	0.97
Killers	1	0.98	0.99	1	1	0.96
Av. domain	1	0.98	0.98	1	1	0.97
Animals						
Otters	1	1	0.96	1	1	0.94
Mammals	1	1	0.95	1	1	0.95
Sharks	1	1	0.98	1	1	1
Av. domain	1	0.99	0.96	1	1	0.92

Table 9 shows the accuracy of each domain at different ranks, while Table 10 shows examples of the extracted arguments.

Comparison with Existing Knowledge Bases We measure the ability of our system to learn verb-based relations of a term with respect to already existing knowledge bases, which have been created in a similar way. When we compared results against existing knowledge bases, we noticed that Yago [44] has more

Table 10 Examples of learned arguments

Term-verb	Preposition	Learned arguments
Terrorists communicate	**through**	Violence, micro technology, orkut secure channels, email, internet, internet networks, cellphones
	with	Their contacts, each other, the world, other terrorists, US citizens, Korea, governments, America
	in	Brief, code, VW, Russian, French, various ways, secret, English
	by	Mail, phone, fax, email
	without	Detection, tapping calls
Birds fly	**above**	Earth, castles, our heads, trees, lake, field, river, cloud, city
	through	Air, night, sky, park, country club, wind, storm, region, city
	around	Her, fish, house, my head, bird feeder, home, your city, ruins, place
	across	Sky, gulf, screen, rainbow, sunset, horizon, African savanna, our path, street, hometown
	into	Windows, walls, power lines, towers, sun, sea, darkness, mist, house

detailed information for the arguments of the verb relations rather than the verb relations themselves. Repositories like ConceptNet[2] [28] contain 1.6 million assertions, however they only belong to 20 relation types such as *is-a*, *part-of*, *made-of*, *effect-of* among others. This analysis shows that despite their completeness and richness, existing knowledge repositories can be further enriched with verb-based relations produced by our learning procedure.

Comparison with Existing Relation Learners For our comparative study with existing systems, we used ReVerb[3] [7], which similarly to our approach was specifically designed to learn verb-based relations from unstructured texts. Currently, ReVerb has extracted relations from ClueWeb09[4] and Wikipedia, which have been freely distributed to the public. ReVerb learns relations by taking as input any document and applies POS-tagging, NP-chunking and a set of rules over all sentences in the document to generate triples containing the verbs and the arguments associated with them. According to [7] ReVerb outperforms TextRunner [3] and the open Wikipedia extractor WOE [48] in terms of the quantity and quality of the learned relations. For comparison, we took the terms *ant, president, terrorists*

[2] http://web.media.mit.edu/~hugo/conceptnet/#overview.

[3] http://reverb.cs.washington.edu/.

[4] http://lemurproject.org/clueweb09.php/.

Table 11 Comparison of verb-based relation learners

Term	ClueWeb (ReVerb)	Web (DAP)
Ants	32	607
Presidents	32	705
Terrorists	96	941

and extracted all verbs relations found by ReVerb in the ClueWeb09 and Wikipedia triples.

Table 11 summarizes the total number of unique verbs extracted by ReVerb in ClueWeb09 (Wikipedia had even lower coverage than ClueWeb09).

We have manually validated the correctness of the extracted verbs by ReVerb and have seen that their accuracy is 100 %. However ReVerb's recall is significantly lower. This shows that our simple, yet powerful verb extraction procedure could be easily used to enrich existing taxonomies.

7 Conclusion

We have described a simple, fast and accurate pattern-based approach for taxonomy induction and enrichment. For the purpose we used variants of DAP patterns introduced by Kozareva et al. [18, 22, 23] to learn hyponyms, hypernyms, verb relations and position all concepts with respect to each other in order to induce a taxonomy. We verified the performance of our simple, yet powerful algorithms by conducting large-scale evaluation on four different domains. For each phase extraction, taxonomization and enrichment, we conducted a manual evaluation and compared the obtained results against existing repositories such as WordNet and ConceptNet. We also compared results against existing knowledge extraction and taxonomization algorithms. The overall results showed that our approach accurately extracts terms, reconstructs existing taxonomies, enriches taxonomies with verb-based relations and even finds knowledge missing from existing repositories. In [20] we have shown how the same pattern-based bootstrapping technology can learn hyponym, hypernym and relations for Spanish. In the future, we are interested in building taxonomies for languages other than English. In addition, we want to use the generated resources for various NLP applications such as Question Answering [29], textual entailment [10, 45] and noun compound interpretation [30].

References

1. Agirre E, Lopez de Lacalle O (2004) Publicly available topic signatures for all WordNet nominal senses. In: Proceedings of the 4rd international conference on Languages Resources and Evaluations (LREC), Lisbon
2. Amsler RA (1981) A taxonomy for English nouns and verbs. In: Proceedings of the 19th annual meeting on association for computational linguistics, Morristown, NJ. Association for Computational Linguistics, pp 133–138
3. Banko M, Cafarella, MJ, Soderl, S, Broadhead M, Etzionio O (2007) Open information extraction from the web. In: Proceedings of IJCAI, pp 2670–2676
4. Cuadros M, Rigau G (2008) KnowNet: building a large net of knowledge from the web. In: The 22nd international conference on computational linguistics (Coling'08), Manchester
5. Davidov D, Rappoport A (2006) Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In: Proceedings of the 21st international conference on Computational Linguistics COLING and the 44th annual meeting of the ACL, pp 297–304
6. Etzioni O, Cafarella, M, Downey D, Popescu AM, Shaked T, Soderland S, Weld DS, Yates A (2005) Unsupervised named-entity extraction from the web: an experimental study. Artif Intell 165(1):91–134
7. Fader A, Soderland S, Etzioni O (2011) Identifying relations for open information extraction. In: Proceedings of the 2011 conference on empirical methods in natural language processing, pp 1535–1545
8. Fleiss J (1971) Measuring nominal scale agreement among many raters. Psychol Bull 76(5):378–382
9. Girju R, Badulescu A, Moldovan D (2003) Learning semantic constraints for the automatic discovery of part-whole relations. In: Proceedings of the conference of the North American chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT), pp 1–8
10. Glickman O, Dagan I, Koppel M (2005) A probabilistic classification approach for lexical textual entailment. In: Proceedings of the twentieth national conference on artificial intelligence and the seventeenth innovative applications of artificial intelligence conference, pp 1050–1055
11. Hearst M (1992) Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on computational linguistics, pp 539–545
12. Heyer G, Läuter M, Quasthoff U, Wittig Th, Wolff Chr (2001) Learning relations using collocations. In: Maedche A, Staab S, Nedellec C, Hovy E (eds) Proceedings of the IJCAI workshop on ontology learning, Seattle/WA
13. Hovy EH (1998) Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In: Proceedings of the LREC conference
14. Hovy EH, Kozareva Z, Riloff E (2009) Toward completeness in concept extraction and classification. In: Proceedings of the 2009 conference on empirical methods in natural language processing (EMNLP), pp 948–957
15. Ide N, Veronis J (1994) Machine readable dictionaries: what have we learned, where do we go. In: Proceedings of the post-COLING 94 international workshop on directions of lexical research, Beijing, pp 137–146
16. Joaquim S, Kozareva Z, Noncheva V, Lopes G (2004) Proceedings of TALN, pp 19–21
17. Katz B, Lin J (2003) Selectively using relations to improve precision in question answering. In: Proceedings of the EACL-2003 workshop on natural language processing for question answering, pp 43–50
18. Kozareva Z (2012) Learning verbs on the fly. In: Proceedings of the 24th international conference on computational linguistics (COLING 2012)
19. Kozareva Z, Hovy EH (2010) A semi-supervised method to learn and construct taxonomies using the web. In: Proceedings of the 2010 conference on empirical methods in natural language processing, pp 1110–1118

20. Kozareva Z, Hovy EH (2010) Not all seeds are equal: measuring the quality of text mining seeds. In: Proceedings of the human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics, pp 618–626
21. Kozareva Z, Hovy EH (2010) Learning arguments and supertypes of semantic relations using recursive patterns. In: Proceedings of the 48th annual meeting of the association for computational linguistics, pp 1482–1491
22. Kozareva Z, Riloff E, Hovy EH (2008) Semantic class learning from the web with hyponym pattern linkage graphs. In: Proceedings of the NAACL-HLT conference, pp 1048–1056
23. Kozareva Z, Hovy EH, Riloff E (2009). Learning and evaluating the content and structure of a term taxonomy. In: Proceedings of AAAI spring symposium: learning by reading and learning to read, pp 50–57
24. Kozareva Z, Voevodski K, Teng S-H (2011) Class label enhancement via related instances. In: Proceedings of the conference on empirical methods in natural language processing, pp 118–128
25. Lin D (1998) Automatic retrieval and clustering of similar words. In: Proceedings of the 17th international conference on computational linguistics (COLING), pp 768–774
26. Lin D, Pantel P (2002) Concept discovery from text. In: Proceedings of the 19th international conference on computational linguistics (COLING), pp 1–7
27. Litkowski K, Hargraves O (2007) SemEval-2007 Task 06: Word-sense disambiguation of prepositions. In: Proceedings of the fourth international workshop on semantic evaluations, pp 24–29
28. Liu H, Singh P (2004) Focusing on ConceptNet's natural language knowledge representation. In: Commonsense reasoning in and over natural language proceedings of the 8th international conference on knowledge-based intelligent information and engineering systems (KES 2004), pp 71–84
29. Moldovan DI, Harabagiu SM, Pasca M, Mihalcea R, Goodrum R, Girju R, Rus V (1999) Lasso: a tool for surfing the answer net. In: Proceedings of the TREC conference
30. Nakov P, Kozareva Z (2011) Combining relational and attributional similarity for semantic relation classification. In: Proceedings of recent advances in natural language processing, pp 323–330
31. Navigli R, Velardi P, Cucchiarelli A, Neri F, Cucchiarelli R (2004) Extending and enriching WordNet with OntoLearn. In: Proceedings of the second Global WordNet conference 2004 (GWC 2004), pp 279–284
32. Pantel P, Pennacchiotti M (2006) Espresso: leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of 21st international conference on computational linguistics (COLING) and 44th annual meeting of the Association for Computational Linguistics (ACL)
33. Pasca M (2004) Acquisition of categorized named entities for web search. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management (CIKM), pp 137–145
34. Pennacchiotti M, Pantel P (2006) Ontologizing semantic relations. In: Proceedings of the international conference on Computational Linguistics (COLING) and the annual meeting of the Association for Computational Linguistics (ACL), pp 793–800
35. Ponzetto S, Navigli R (2010) Knowledge-rich word sense disambiguation rivaling supervised systems. In: Proceedings of the 48th annual meeting of the Association for Computational Linguistics (ACL 2010), Uppsala
36. Richardson SD, Dolan WB, Vanderwende L (1998) Mindnet: acquiring and structuring semantic information from text. In: Proceedings of the 36th annual meeting of the Association for Computational Linguistics and 17th international conference on computational linguistics (ACL '98), vol 2. Association for Computational Linguistics, Stroudsburg, PA, pp 1098–1102
37. Rigau G, Rodriguez H, Agirre E (1998) Building accurate semantic taxonomies from monolingual MRDs. In: Proceedings of the 36th annual meeting of the Association for Computational Linguistics and 17th international conference on computational linguistics (ACL '98), vol 2. Association for Computational Linguistics, Stroudsburg, PA, pp 1103–1109

38. Riloff E, Shepherd J (1997) A corpus-based approach for building semantic lexicons. In: Proceedings of the second conference on empirical methods in natural language processing (EMNLP), pp 117–124
39. Ritter A, Soderland S, Etzioni O (2009) What is this, anyway: automatic hypernym discovery. In: Proceedings of the AAAI spring symposium on learning by reading and learning to read
40. Ritter A, Mausam, Etzioni O (2010) A latent Dirichlet allocation method for selectional preferences. In: Proceedings of the Association for Computational Linguistics conference (ACL)
41. Roberto N, Velardi P, Faralli S (2011) A graph-based algorithm for inducing lexical taxonomies from scratch. In: Proceedings of IJCAI 2011, pp 1872–1877
42. Schmid H (1994) Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the international conference on new methods in language processing, pp 44–49
43. Snow R, Jurafsky D, Ng AY (2006). Semantic taxonomy induction from heterogenous evidence. In: Proceedings of the international conference on computational linguistics (COLING) and the annual meeting of the Association for Computational Linguistics (ACL)
44. Suchanek FM, Kasneci G, Weikum G (2007). Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web (WWW), pp 697–706
45. Szpektor I, Dagan I, Bar-Haim R, Goldberger J (2008) Contextual preferences. In: Proceedings of the annual meeting of the Association for Computational Linguistics (ACL), pp 683–691
46. Widdows D (2003) Unsupervised methods for developing taxonomies by combining syntactic and statistical information. In: Proceedings of the HLT-NAACL conference
47. Wilks Y, Fass D, ming Guo C, Mcdonald JE, Plate T, Slator BM (1988) Machine tractable dictionaries as tools and resources for natural language processing. In: Proceedings of the 12th conference on computational linguistics, Morristown, NJ. Association for Computational Linguistics, pp 750–755
48. Wu F, Weld D (2010) Open information extraction using Wikipedia. In: Proceedings of the 48th annual meeting of the Association for Computational Linguistics, pp 118–127
49. Yang H, Callan J (2009) A metric-based framework for automatic taxonomy induction. 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 (ACL-IJCNLP), vol 1, pp 271–279

A Topology-Based Approach to Visualize the Thematic Composition of Document Collections

Patrick Oesterling, Christian Heine, Gunther H. Weber, and Gerik Scheuermann

Abstract The thematic composition of document collections is commonly conceptualized by clusters of high-dimensional point clouds. However, illustrating these clusters is challenging: typical visualizations such as colored projections or parallel coordinate plots suffer from feature occlusion and noise covering the whole visualization. We propose a method that avoids structural occlusion by using topology-based visualizations to preserve primary clustering features and neglect geometric properties that cannot be preserved in low-dimensional representations. Abstracting the input points as nested dense regions with individual properties, we provide the user with intuitive landscape visualizations that illustrate the high-dimensional clustering structure occlusion-free.

1 Introduction

During the last decades, increased storage and growing computational power enabled text data to serve as an important information source to mine. A consequence of the excessive supply of information is that it cannot be consumed in its entirety; neither the existing material nor new portions that are added every day. Automated methods are necessary to preprocess, classify, and visually summarize coherent parts. To help users navigate this massive amount of data, researchers constantly look for appropriate models for complex linguistic features

P. Oesterling (✉) • G. Scheuermann
Image and Signal Processing Group, Institute of Computer Science, Leipzig University, Leipzig, Germany
e-mail: oesterling@informatik.uni-leipzig.de; scheuermann@informatik.uni-leipzig.de

C. Heine
Scientific Visualization Group, Department of Computer Science, ETH Zürich, Zürich, Switzerland
e-mail: cheine@inf.ethz.ch

G.H. Weber
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA
e-mail: ghweber@lbl.gov

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_4

and relationships, such as entities, emerging trends, temporal peaks in what people discuss, or segmentations into occurring (sub-)topics.

To analyze a document collection's structure in terms of the topics it contains, a well-known approach transforms texts into high-dimensional vectors using the vector space model [30] in combination with, e.g., the tf-idf [32] weighting. This method results in a point cloud where accumulations represent documents that share vocabulary and word significances, i.e., that share topics. The clustering structure of this point cloud describes the documents' topical composition: the number of clusters reflects topic count, sub-clusters and their nesting describe sub-topics, cluster size represents how many documents share a topic, and cluster compactness and separateness indicate topical generality and preciseness, respectively. Comparing these properties supports qualitative and quantitative statements about occurring topics (Fig. 1).

Providing a user with an adequate presentation of high-dimensional point clouds to facilitate visual cluster analysis is not an easy task. Established visualizations for point data include projections and axis-based techniques, such as parallel coordinates plots [14]. Although being used frequently, these techniques suffer severe drawbacks, including *structural occlusion* when clusters overlap in the final plot and *visual complexity* when dealing with large or noisy data, i.e., if noise covers the whole visualization.

In this article, we describe an alternative view on point data to advance the analysis and visualization of its high-dimensional clustering structure. We

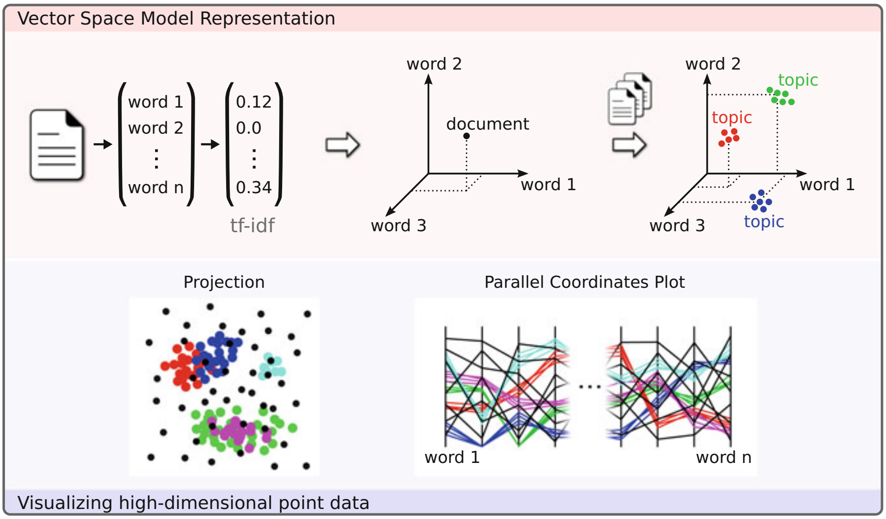


Fig. 1 In the vector space model, a document resides as a vector in a high-dimensional space with words as dimensions. Multiple documents then accumulate to clusters if they share similar content. Inspecting the topical composition by a direct visualization of the clustering structure is difficult because projection artifacts and noise easily obscure intrinsic features

provide a solution to the problem of structural occlusion of features in high-dimensional data by neglecting geometric properties that cannot be preserved in low-dimensional visualizations. These properties include relative positions, absolute distances between points and clusters, and cluster shape. Taking a topological-structural view on the points, we can extract primary clustering properties and preserve them without loss in the final visualization. Our technique is independent from the points' dimensionality and also supports brushing-and-linking individual structural features to visualizations specialized for local analysis, such as projections and axis-based techniques. Note that linking only subsets to the latter significantly improves their usability and readability, and that feature selection in these visualizations would otherwise be complicated by occlusion and noise.

For demonstration purposes, we apply our method to document visualization. More precisely, we represent text data with the vector space model and reveal the thematic composition by accurate depiction of the high-dimensional point cloud's clustering structure. This includes accurate identification and comparison of topic characteristics and also to relate single documents to occurring topics. Finally, meta-information like topic descriptors, document titles or provided class names are used to color and label documents and topics. Compared to classic text-mining approaches, the insights to be taken from this representation are rather limited. The focus is primarily on the thematic overview aspect, rather than on sophisticated linguistic analysis. Still, from a visual analysis point of view, providing an initial topical overview of the documents is fundamental and often the starting point of further, local analysis. Therefore, we compete against those techniques that have issues with accurate depiction of documents represented in the vector space model. As we cannot address every detail in the scope of this article, we refer to prior publications [23–25] for further reading, in-depth explanations and evaluations.

2 Related Work

The work presented in this article is primarily about the illustration of document collections as an example application of the analysis and visualization of multidimensional feature vectors—or simply high-dimensional points. There are multiple alternatives to represent, process and visualize textual data, each with their own merits and drawbacks. We refer the interested reader to comprehensive survey papers like Silic et al. [33] and only mention some of the most commonly known concepts related to our approach.

2.1 Visualization of High-Dimensional Point Data

Classic solutions to visualize high-dimensional point data include projections and axis-based techniques. In the former case, projections aim to map high-dimensional points to the two- or three-dimensional space for presentation on the screen. Popular dimension reduction approaches are often based on singular value decomposition, like, among others, principal component analysis (PCA) [17] or latent semantic indexing [4]; on least square approximations [29]; on multidimensional scaling [21], e.g. Glimmer [12] or Sammon's mapping [31]; or on neuro-computational algorithms like Kohonen's self-organizing maps [20]. For moderate data dimensionality, scatterplot matrices provide views from all axis-aligned directions; here "Rolling the dice" [6] helps navigate through the matrix using animations. In case of supervised projections, not only the distances in the original domain, but also additional information are employed to optimize low-dimensional embeddings. For example, linear discriminant analysis (LDA) [2] uses a segmentation of the input data into k classes to perform a supervised projection into an optimal $(k-1)$-dimensional space by minimizing and maximizing the intra- and interclass, respectively, distances of all points in the reduced dimensional space.

Axis-based techniques for high-dimensional point data deviate from the Cartesian norm to align axes perpendicular to each other. Prominent examples are parallel coordinate plots (PCP) [14], where axes are arranged in parallel next to each other, or star plots [16], where axes are arranged circularly around a center point. In both cases, data points are represented as line segments that connect the axes at the values assigned to each dimension. If vectors share similar values and dimensions, line segments accumulate to visual line bundles that represent point accumulations in the high-dimensional space. Without pairwise perpendicularity, potentially many dimensions or attributes can be visualized at the same time; though not necessarily as intuitive and familiar as in well-known 2-D or 3-D scatterplots. Since features in PCPs can sometimes be hard to interpret, Inselberg et al. [13] elaborate on the relation between line segment constellations and their counterparts in the vector space.

Depending on the input data's size and dimensionality, both visualization techniques can have severe issues when it comes to accurate depiction of high-dimensional point cloud structure. They delegate feature identification by letting the user inspect visually whether the drawing exhibits noticeable accumulations of remarkable shape or distance to each other. Therefore, if the drawing contains all data items, for larger data sets, this leads to significant visual complexity and the screen-resolution bounds the number of presentable items. Also note that global noise typically covers the whole visualization because all points are handled equally. In axes-based techniques, line segments require many pixels for a single data point and they suffer from occlusion and crossings. Feature identification is also complicated by the order in which axes are arranged. The projection error and the immanent information loss of projective techniques can obscure existent structure, but also produce visual artifacts, i.e. illusions of structure that is not present. We will revisit these limitations in Sect. 3 based on example data, and we address other problems that arise if high-dimensional spaces are analyzed with distance-based algorithms.

2.2 Representation and Visualization of Textual Data

Because unstructured text is difficult to analyze and not suitable for visualization, text data is usually transformed into other representations. An early model is *bag-of-words*, an instance of the vector space model [30], where texts, such as sentences or documents, are transformed into vectors by counting the occurrence of words or word forms. This produces high-dimensional vectors in a space where each dimension represents a word. Using the vector representation, grammar and word order are disregarded and vector (dis-)similarity is often defined as (Euclidean) distance or the cosine between two vectors. Instead of using term frequencies, it is common to weight terms by other schemes, of which tf-idf [32] is popular. In essence, the more dimensions (words) two points (documents) share and the more similar the (tf-idf) values are for these dimensions, the closer both points will be to each other. That is, the compactness of a cluster depends on how similar word use is throughout the documents of this cluster. Documents using rare words consequently reside on a cluster's border or between other clusters; they are thus not strictly related to any other topic, or related partially to several ones.

Although not always appropriate for this purpose (see above), the structure of texts represented as high-dimensional vectors is often depicted via a projection that optimizes some general or method-specific criterion. Prominent implementations include Sammon visualization [31], the text map explorer [27] or the projection explorer (PEx) [28]. Instead of illustrating text items as points or graphs, more intuitive metaphors have been introduced that depict features in the data using heat maps in WEBSOM [19], landscapes (SPIRE/IN-SPIRE [37]), islandlike depictions using wavelets (topic islands [22]), or height fields (VxInsight [3]). Note that if such metaphors are based on lower dimensional representations of the input data, they still suffer from feature aggregation and thus from information loss.

3 Pitfalls of Distance-Based Analysis and Projective Visualization

The common solution to provide the user with a projection of high-dimensional data aims at delegating feature identification to the human visual system. That is, the user identifies structural features as a certain number of coherent or accumulated groups of significant size, shape and separation in the drawing. To this end, the projection has to preserve (dis-)similarities between points, typically defined as Euclidean distance, as much as possible to present a reasonably good idea of a high-dimensional point cloud's structure. However, this approach involves dealing with two problems, namely identifying (dis-)similarities between input data items in the original domain and accurately presenting them to the user on the screen. This section gives attention to these challenges and explains why we finally reject projections as visualization and use topological concepts instead.

3.1 Distance-Based Analysis in High-Dimensional Spaces

Almost 50 years ago, Richard Bellman first spoke about *"a malediction that has plagued the scientist from the earliest days"*. While his statement, basically, refers to the problems caused by increasing the number of independent variables in different fields of application, especially for metric spaces, where the problem often is termed the *curse of dimensionality*, this means an exponential increase of volume and data sparsity with each additional dimension. Particularly for distance-based approaches, it has been shown [10, 18, 34] that depending on the chosen metric, distances between points either depend on the dimensionality (L_1 norm), approach a constant (L_2 norm), or zero ($L_{d \geq 3}$ norm). That is, the ratio between the distances to a point's farthest and closest point approaches one. As a consequence, distance variation vanishes in higher dimensions and some distance-based relationships such as *nearest neighbors* become fragile in those spaces. Of course, if distances become uniform, every distance-based approach is affected by this phenomenon. To illustrate the effects for proximity-based problems like similarity and clustering search, we consider the Medline data set (provided in [2]). It consists of 1,250 vectors in a 22,095-dimensional space, divided into five equally sized clusters. The black graph in Fig. 2a shows the distance distribution between any two points. As can be seen,[1] distances are spread from 0.0 to 1.414, but around 98.1 % of them are greater than 1.37. The key issue is that both the inter-cluster distances (green) and the intra-cluster distances (red), which we obtain from given clustering information, show the same behavior. However, for clustered data such a graph typically shows

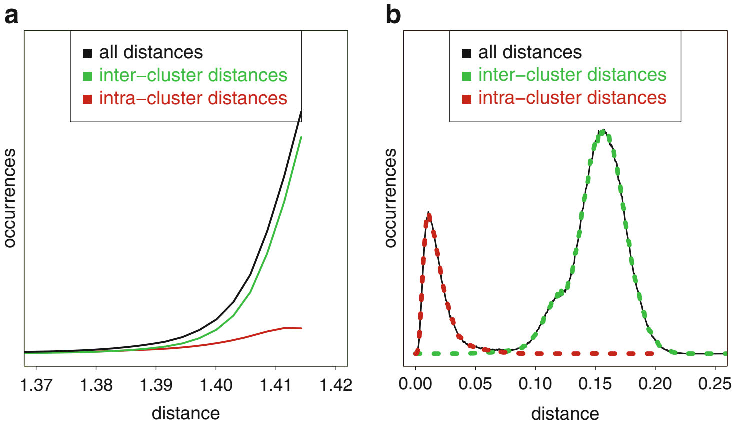


Fig. 2 Medline data: two plots of the distances between every two points (*black*) and their partition into intercluster (*green*) and intracluster (*red*) distances, (**a**) in the original space and (**b**) after applying the LDA

[1] The diagrams in Figs. 2 and 3 can be arbitrarily magnified in the electronic version of this article.

two peaks: one for the distances inside the clusters and another one representing the average distance between the clusters [34]. Consequently, because only one peak is present, any purely distance-based approach will have problems with finding the underlying clustering of this data set.

3.2 Projections to Visualize High-Dimensional Clusterings

We consider the Reuters document collection, kindly provided in [2], to demonstrate the difficulties that projections have even with moderately dimensional data. The example consists of 800 vectors in an 11,941-dimensional space, assigned equally to the following $k = 10$ classes (the letters are used in Fig. 3): acquisitions ('a'), corn ('c'), earn ('e'), grain ('g'), interest ('i'), money-fx ('m'), crude ('r'), ship ('s'), trade ('t'), and wheat ('w'). Note that unless the points are embedded in a small subspace of all 11,941 dimensions, both the projection error as well as computation times, e.g. for singular value decomposition, may be significant for such high dimensionality. Therefore, we exploit given classification information and use a supervised projection to consider an optimized projection of the Reuters data. The Rank-2 linear discriminant analysis (LDA) as proposed by Choo et al. [2] consists of two subsequent LDA projections: the first one from the original space into an intermediate $(k-1)$-dimensional space and a second projection down to two dimensions. LDA assumes a relation between clusters and classes and therefore strives to maximize and minimize, respectively, the intercluster and intracluster distances in the optimal $(k-1)$-dimensional space. For the technical part, we refer the interested reader to [2].

As shown in Fig. 3a, the Rank-2 LDA preserves the clustering well. However, two clusters on the right-hand side and in the upper left-hand corner contain points of different classes. The pivotal question is why. There are two explanations: clusters

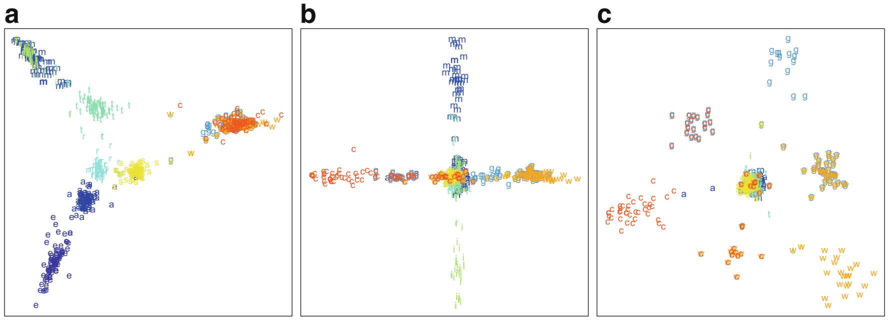


Fig. 3 Reuters data: (**a**) Scatterplot of the Rank-2 LDA with clusters that contain points of mixed classes. Looking at the same 9-D point cloud (**b**) from the 7th and 8th and (**c**) from the 7th and 9th dimension reveals that these (alleged) clusters actually consist of well-separated accumulations

are indeed mixed in the original domain or we face overdrawing artifacts due to projection errors of the second LDA projection. Since the first LDA preserves cluster relationships in the $(k-1)$-dimensional space, we can look into the nine-dimensional data using a scatterplot matrix. Figure 3b, c reveal that our second assumption is true. If we consider the points from the directions of the 7th and 8th (Fig. 3b) or the 7th and 9th (Fig. 3c) dimensions, we see that both mixed clusters in Fig. 3a consist of clusters that are actually separated in the intermediate 9-D space. Granted, this result is not completely surprising because the second LDA only uses two axes to discriminate the classes which contribute most to the optimization criterion. Nevertheless, due to the lack of any information about the intermediate space, the user will most likely tend to assume mistakenly that clusters are really mixed.

3.3 Rethinking: How to Present What to the User

Identifying proximities in high-dimensional spaces and subsequently presenting them in 2-D are two challenges on their own. Using distance-based techniques, a reduction of the overall dimensionality prior to any analysis is always reasonable; to escape the curse of dimensionality, but also to reduce runtimes. For documents, this means as few as possible words should serve as dimensions. This includes using highly descriptive words, the application of word stemming, but also a restriction to words with specific meaning, e.g. *volatile* words as proposed by Teresniak et al. [11, 35]. Using classification information, supervised projections like LDA also support finding an accurate, but lower dimensional representation of the point cloud in an optimal-dimensional space—provided that classes and clusters correlate. Substituting the Euclidean distance by application-driven metrics is another solution to bypass the curse of dimensionality. However, this also reduces the algorithm's independence from the underlying application.

Presenting high-dimensional structure in a 2-D image is the more challenging part, especially if structural features should not overlap. Arranging 2-D points in a way that their distances reflect (dis-)similarities in the original domain is perfectly fine—if the original domain is also two-dimensional. If the data is not intrinsically embedded in a 2-D subdomain, it is unlikely to preserve all pairwise distances without loss. It turns out that projections seem suboptimal to convey high-dimensional structure. On the one hand, drawing all data points to let them *simulate* high-dimensional proximities is not necessarily needed to present a structural overview. On the other hand, this approach must be questioned if the presented structure is not even guaranteed to be correct in the sense that all true features are clearly visible and no false ones are introduced.

To find another solution to visualize high-dimensional structure, we have to step back and first define what we are interested in. Talking about clusterings, the primary subject of interest are point accumulations surrounded by empty regions or noise. That is, one is interested in how many coherent groups there are and whether they contain each other (sub-clusters). Cluster properties then include their quantitative size, distinctness or compactness, or how points are distributed in a cluster. It turns out that to convey this information, we neither need the data points themselves nor their pairwise distances. In fact, taking a closer look, it is the point distances and properties derived from them that prevent accurate identification of other fundamental properties in the first place. So if we can do without geometric properties like cluster shape, relative position, or precise distance between clusters, we could better focus on the remaining properties. This is a relatively small price to pay, especially regarding the fact that the preservation of these secondary properties could otherwise destroy the overall clustering depiction in the 2-D image.

Putting it all together, instead of using lossy projections, we propose to neglect geometric properties altogether and use topological concepts. We represent the point cloud by a high-dimensional density function and analyze its topology. That is, we consider point accumulations as dense regions surrounded by low density and the function's topology tells us everything about the relation between dense regions and their properties: region count and nesting reflect the quantity of (sub-)clusters, absolute density translates into cluster significance, the number of a region's points gives a quantitative measure, the point distribution inside a cluster reflects compactness and distinctness, and the whole approach is also robust with respect to noise. But we cannot say anything about geometry anymore; not about a dense region's shape, its geometric extent, its relative position or how far away it is from another separated region. If we wanted to preserve such information in a 2-D image, we would instantly be back in the realm of projection errors and information loss. But instead we can compute the density function and its topology in arbitrary dimensions and we can represent this information without loss using topology-based visualizations that do not suffer from structural occlusion. Furthermore, by considering density-size ratios, we can still approximate geometric extent: while only a few points of high density must be rather compact, many points of low density must be spread. In the end, we can even select dense regions and link them individually to projections or axis-based techniques for further local analysis, like subspaces or approximated shape. Note that selecting single features in these visualizations would otherwise be complicated in the presence of noise or if features overlap.

4 Topological Representation of Clustering Structure

We represent a high-dimensional point cloud's clustering structure by the topology of its density function because this structure can be preserved in a visualization. The density estimation may be thought of as reconstructing a probability space for documents from the samples, i.e. documents we observed. This section describes how we turn point data into a density function, how a topological structure called the join tree encodes the clustering, which cluster properties it preserves, and which parameters the user can vary to steer the analysis and to simplify the structural view. Due to space limitations, we refer the interested reader to [24] for further technical details and evaluations.

4.1 From Point Data to a High-Dimensional Density Function

Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of points in a fixed-dimensional Euclidean space $\mathbb{R}^d$, density estimation is generally performed by the application of a filter kernel, e.g. the Gaussian filter

$$f(x) = \frac{1}{n(\sigma\sqrt{2\pi})^d} \sum_{i=1}^{n} exp\left(-\frac{\delta(x, p_i)^2}{2\sigma^2}\right),$$

with $\delta(x, p_i)$ being the Euclidean distance between sample x and point p_i.

Because it is infeasible to compute the topology of P's density function analytically, we need to construct a mesh on the given points. In lower dimensions, the function can be sampled on a regular grid of sufficient resolution, but in higher dimensions, this approach is impractical because regular grids grow exponentially in size with every additional dimension. Therefore, we aim for a simpler function f' of very similar topology that is a piecewise-linear interpolation on a complex of simplices. Informally, a simplicial complex is a mesh of simplices, i.e. generalizations of triangles and tetrahedra to arbitrary dimensions.

The theoretically best simplicial complex is the high-dimensional Delaunay triangulation. If we use the Delaunay triangulation on P and assume $f'(p_i) = f(p_i), \forall p_i \in P$, there are two problems: a prohibitive runtime of $O(n^{d/2})$ [8] to construct it and a rather coarse approximation of f since the density between two points can be lower than at those points. In this case, f' would lack topological features which indicate that two points (or regions) are separated by a region of low density.

Coarseness can be countered by adding all topologically relevant points of f to the Delaunay triangulation, which are very hard to compute. Fortunately, the topology is merely a tool to identify dense regions and their nesting. So as long as we find all dense regions accurately, we do not really care about exact positions of f's topological features. Therefore, we use a heuristic that adds a further sample on the midpoint m of each mesh edge and require $f'(m) = f(m)$ if the density at this position is lower than at the edge's endpoints. This process is referred to as *upsampling*.

The prohibitive runtime to construct the Delaunay triangulation in arbitrary dimensions can be mitigated by using subsets instead. Because we only need the edges to look for topological events, instead of a Delaunay triangulation we can work with *neighborhood graphs* [15], also called *proximity graphs*. The literature knows many neighborhood graphs, like the *Delaunay graph* (DG), which is only the vertices and the edges of the Delaunay triangulation, the *Gabriel graph* (GG) [9], the *relative neighborhood graph* (RNG) [15], the *Euclidean minimum spanning tree* (EMST), or the *nearest neighborhood graph* (NNG) [15]. These graphs share an important subset relationship in Euclidean spaces:

$$NNG \subseteq EMST \subseteq RNG \subseteq GG \subseteq DG.$$

Two important properties follow directly from this subset relationship: each supergraph of the NNG also contains the edge of each vertex to its nearest neighbor. Second, because it is a connected graph, all supergraphs of the EMST are connected as well—connectedness is necessary to determine the topology.

In summary, to approximate the Delaunay triangulation, we can work with the Gabriel graph, the relative neighbor neighborhood graph, or the Euclidean minimum spanning tree instead. The cost of this simplification is that, starting from the GG, these graphs more and more omit (long) edges and, therefore, coarsen the topology to be found. However, the differences turn out to be rather small [24], and, on the other hand, the times to construct these graphs are significantly shorter; namely $O(dn^3)$ and $O(dn^2)$ for the GG/RNG and for the EMST, respectively. Our final approximation of the density function is an upsampled neighborhood graph, with $f'(v) = f(v)$ for the positions of all graph vertices v_i.

4.2 The Topology of the Density Function

The concepts we use primarily originate from *scalar field topology*, a research area that enjoys great popularity in many disciplines and that is also of vital importance, e.g., in scientific visualization. Although we do not require the reader to have an in-depth knowledge and experience in this field, we still need to provide a brief introduction to clarify what kind of information we actually extract.

A scalar field is simply a function $f : \Omega \to \mathbb{R}$, with Ω being a d-dimensional observation space. A *superlevel set* of f at some function value h is the set $\{x \in \mathbb{R}^d \mid f(x) \geq h\}$ and may consist of zero, one, or more connected components. If the function value h is thought of as time, we can watch the evolution of superlevel sets with decreasing h, seeing them appear, grow, and join. As a metaphor, think of the function f as a landscape initially fully submerged by water. Now, the water is drained slowly and at certain points in time land masses, corresponding to superlevel sets, emerge from the water. This happens at *local maxima* of f, but there are also times when land masses join. This reflects a join of two or more superlevel sets into a larger one at a point called *saddle*. The process stops when all the water is

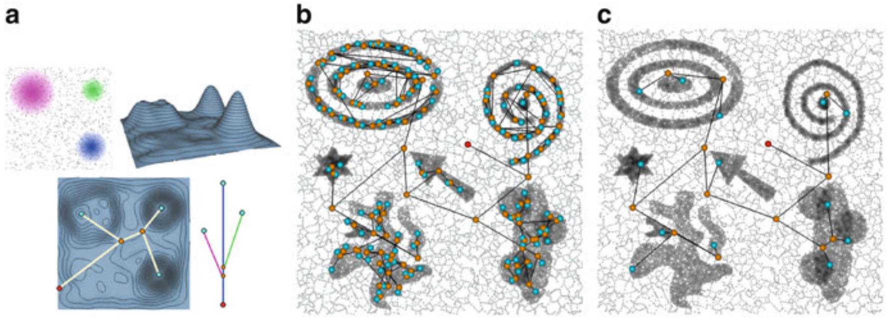


Fig. 4 Artificial data: (**a**) 2-D noisy clustering with density function and contours (*top*), the function from above with the (simplified) join tree on top and the branch decomposition (*bottom*). (**b**) Depending on the filter radius σ, the topological analysis finds accumulations of arbitrary shape as saddle-maximum pairs that take the form of the clusters. (**c**) Insignificant fluctuations in the function are removed with topological simplification; leaving one saddle-maximum pair per feature

drained, i.e. h reaches *f's global minimum*, and there remains only one superlevel set; it covers the full space. The join tree of a scalar function encodes this evolution: local maxima become leaves, saddles become inner vertices, the global minimum becomes the root, and edges connect accordingly. As we are only interested in areas of high density and their nesting, the join tree is sufficient for our purposes. Figure 4a shows a density scalar function. The rings are the boundaries of superlevel sets and the overlayed tree shows the topological changes at critical points. Carr et al. [1] present a simple and fast graph algorithm to compute the join tree based on a simplicial mesh's vertices and edges.

In essence, the structural description of the input data's clustering structure consists of a tree, namely the join tree. Each node of the tree represents a neighborhood graph node, which in turn represents an input data point or an upsampled point. For each node we also store its actual density. The tree consists of the following nodes: the leaves which are the degree-1 *maximum nodes*, one for each density maximum inside the dense regions; the degree-3 *saddle nodes* between dense regions, to be found either at upsampled positions between clusters, on cluster borders, or in noisy regions; the degree-2 *regular nodes* which are simply non-critical points inside a dense region; and a single *minimum node*, which is the tree's root node and represents the global density minimum. In well-separated clusterings, the root is similar to saddles, i.e. it connects separated regions at an upsampled position where the density is typically zero. Otherwise, it is just the data point with the least density and has no special meaning.

4.3 Cluster Properties and Algorithm Parameters

The join tree describes the clustering structure of the underlying point cloud. To extract properties of single clusters, we need two alternative join tree representations: the *unaugmented join tree* and its *branch decomposition* [26]. The unaugmented join tree is obtained by removing all its regular nodes and merging their edges. That is, the tree only consists of critical nodes, but we can still store implicit regular nodes together with each edge to associate them to their dense regions. The join tree can also be decomposed into *branches*, which are paths monotonic with respect to the function values. The *branch decomposition* is an organization of branches as a tree: each node is a branch; the root is the branch of highest priority. One node is a child of another node if the branch it represents has lower priority than the other node's branch and both branches are connected by a saddle. That is, branches are paths through the tree that reflect the nesting of dense regions in terms of a chosen priority, or property. In our work, we use three branch properties: its *persistence* [5] is the difference between the branch's highest and lowest function value, i.e. between the dense region's maximum and its saddle density. Relative to surrounding densities, persistence indicates how distinct a dense region actually is. The number of implicitly stored regular nodes gives a branch's *size*. Note that the relation between a branch's persistence and its size can disclose a dense region's compactness. If only a few points produce high persistence, they must be very compact. Likewise, if a region with many points is not very persistent, these points must be rather spread. Finally, the exact distribution of the function values of implicitly stored points is the branch's *stability*. It is the sum of the points' densities and conceptually describes the amount of energy required to erode the cluster. A region with many points close to the cluster's density maximum is thus both stable and expensive to erode, while a region with point densities primarily near the saddle density is less stable and cheap to erode. The dense regions, their nesting structure and all three properties per (sub-)region will be preserved in our final visualization (cf. Sect. 5).

Now that we have introduced all concepts involved, we can take a closer look at the algorithm's parameters. Technically speaking, both the chosen distance measure as well as the type of the applied kernel filter are configurable parts of the algorithm. However, in this work we always use the Euclidean distance together with Gaussian kernel to obtain a high-dimensional density function. Therefore, the first, but also the most crucial parameter of the algorithm is the kernel window with σ, also called the Gaussian filter radius. Basically every insight about the data derived from the topological analysis depends on this parameter. A filter radius chosen too big results in combining clusters and finding either only one single region or simply not all separated dense regions in the data. A filter radius too small separates everything, splits clusters, and can even assign each data point to its own cluster. Since the appropriate selection of this parameter is of paramount importance, we assist the user in finding the right value with interactive, visual means (cf. Sect. 5). The second parameter is the neighborhood graph. Choosing between the Euclidean minimum

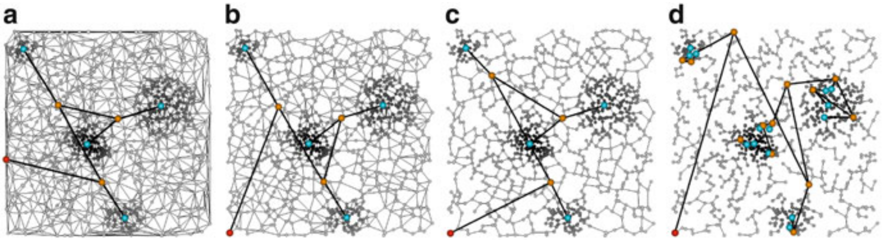


Fig. 5 Artificial 2-D data: Neighborhood graphs augmented with the join trees (*cyan* = maximum, *orange* = saddle, and *red* = minimum) showing the density function's (*dark* = dense) change in topology when using (**a**) the Delaunay graph, (**b**) the Gabriel graph, (**c**) the relative neighborhood graph, and (**d**) the Euclidean minimum spanning tree. Small tree edges were removed from the trees to perform topological simplification using a persistence threshold of 10 % of the maximum persistence

spanning tree, the relative neighborhood graph and the Gabriel graph is a trade-off between accuracy and runtime. Not only are the RNG and the GG worse in runtime complexity, they also produce significantly more edges that need to be upsampled afterwards (cf. Fig. 5). As a rule of thumb, with increasing dimensionality and data size, less complex neighborhood graphs should be used. While the GG is fine for around 30,000 points in some dozens of dimensions, for more input points and higher dimensionality, the RNG or even the EMST are recommended. The effect of using less complex neighborhood graphs is increasing *topological noise* in the guise of numerous saddle-maximum pairs of minor significance. This noise and small fluctuations in the function are countered with *topological simplification* (cf. Fig. 4b). This process means to peel off small leaf edges from the unaugmented join tree that represent insignificant features (Fig. 4c). Whenever a leaf edge is pruned, the other two involved edges of the saddle are combined, leaving a new leaf edge of higher significance. As significance measures we use the three cluster properties described above. That is, a region is considered insignificant if it does not feature enough persistence, size, or stability. The user defines thresholds for these properties to simplify the view on the data. Note that points of simplified (sub-)regions are assigned to their parent regions. In other words, simplification can be imagined as cutting off an insignificant (sub-)hill at its saddle level in the high-dimensional density height field and leaving plateaus behind.

5 Visualization of High-Dimensional Point Cloud Structure

To overcome the drawbacks of projections and axis-based techniques for the depiction of high-dimensional clustering structure, we strive for a final visualization that does not hide features and accurately presents their relationships in the original space. Technically, the density function's join tree already provides all details, so a drawing of the tree in the plane would apparently suffice. However, showing the hierarchy of possibly large trees, while indicating data items and three cluster properties per edge at the same time would require the tree-layout to adhere to several drawing conventions that probably result in cluttered overviews; especially for noisy data sets. Therefore, we need another visualization metaphor that provides quick and intuitive access to the information provided by the tree. This section introduces some topological visualizations that are based on a terrain metaphor to convey clustering information using hills in a landscape.

5.1 Topological Landscape Metaphor

The topological landscape metaphor as proposed by Weber et al. [36] is a landscape visualization that utilizes the user's naturally trained ability to read and understand the structure of a terrain. The metaphor was designed and introduced to illustrate intuitively the hierarchy and properties of a scalar function's contour tree in scientific visualization. In the landscape, maxima and minima of the function show up as hills and sinks in the terrain. The fundamental property of a topological landscape is that it has the same topology like its input contour tree. That is, their structures are equivalent in the sense that at every height, an intersection with the landscape cuts as many hills as the contour tree has edges containing this value. When we apply the landscape metaphor to the join tree of a point cloud's density function, we obtain a terrain that has as many hills as there are clusters in the point cloud. The nesting of clusters is represented by subhill relationships and a hill's extent reflects its corresponding cluster's properties. Since height values in the landscape reflect absolute densities, a dense region's persistence is accurately described by its hill's height. A cluster's size, i.e. the number of points, is represented by the volumetric extent of the corresponding hill's base area.

Figure 6a shows the topological landscape of the Reuters data set from Sect. 3. There is one hill for every density maximum and valleys reflect saddle densities between clusters; which are zero for well-separated clusters. Starting with the root branch of the join tree's branch decomposition, a branch is represented by a hill

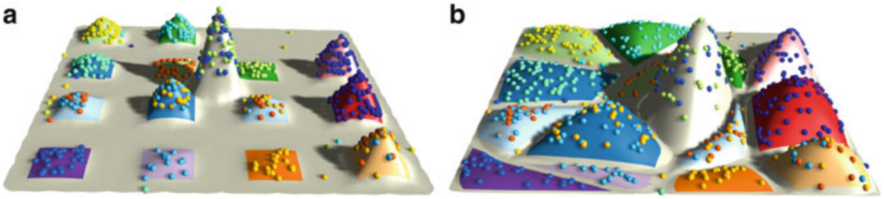


Fig. 6 Reuters data: (**a**) 3-D topological landscape showing a hill for each dense region. Height values reflect absolute densities. Little spheres represent data points (*colored by class*). (**b**) Volumetric distorted landscape with base areas of the hills proportional to cluster sizes

and then its child branches are sorted by decreasing saddle value and subhills are placed on a spiral around the center hill. This construction scheme is applied recursively for every child branch. The colors of the hills have no special meaning and are chosen randomly. Small spheres on the hills represent data points. They are colored by class (thus matching to Fig. 3) and they are placed at a random position on the contour at the height of their density. Their placement thus accurately describes the density distribution inside a cluster. Note that Euclidean distance between hills or spheres has no concrete meaning—only the nesting of hills and their saddle densities give information about the clustering. As can be seen in Fig. 6a, separated hills for the 'c'/'w'/'g' and the 'i'/'m' data points accurately disclose that these points actually accumulate to several clusters in the nine-dimensional space. Because the root branch belongs to the density maximum, the highest hill, which represents the 'i'/'m'-cluster, is always found in the center. A volumetric distorted version of the landscape, with base areas proportional to cluster sizes, is shown in Fig. 6b.

5.1.1 Atoll-Like Flattened Topological Landscape

Although the topological landscape already eliminates structural loss for high-dimensional data, the 3-D visualization still suffers from view-dependent occlusion problems. That is, depending on the viewing direction, hills can occlude each other and some data spheres are invisible because they reside on a hill's back side. To mitigate these issues, we can work with a flattened version of the original landscape. At first, we use hypsometric tinting to color the landscape by height values. To keep the strong expressiveness of the visual metaphor, we use a transfer function that maps naturally occurring colors to the landscape: going from blue (water) and yellow (beach) through green (grass) into brown (mountains) and finally to white (snowy mountain top). Afterwards, we determine isolines to permit a better comparison of density values and then we flatten the landscape by setting height values to zero. Because hills are not occluded anymore and because all data spheres are visible at the same time, we can augment the visualization with labels to provide additional information. To summarize the data inside a cluster, sophisticated labels like text or charts can be placed above each hill. For the data points, we implement the Excentric Labeling [7] approach. Moving a focus lens over the atoll, data spheres are labeled with text if they are enclosed by the focus area. To label clusters and points in Fig. 7, we used the Reuters data's class-names and the colors assigned to them.

5.2 *Topological Landscape Profile*

The original topological landscape metaphor [36] requires three dimensions because it expects a contour tree as input. However, in our analysis, we only work with the

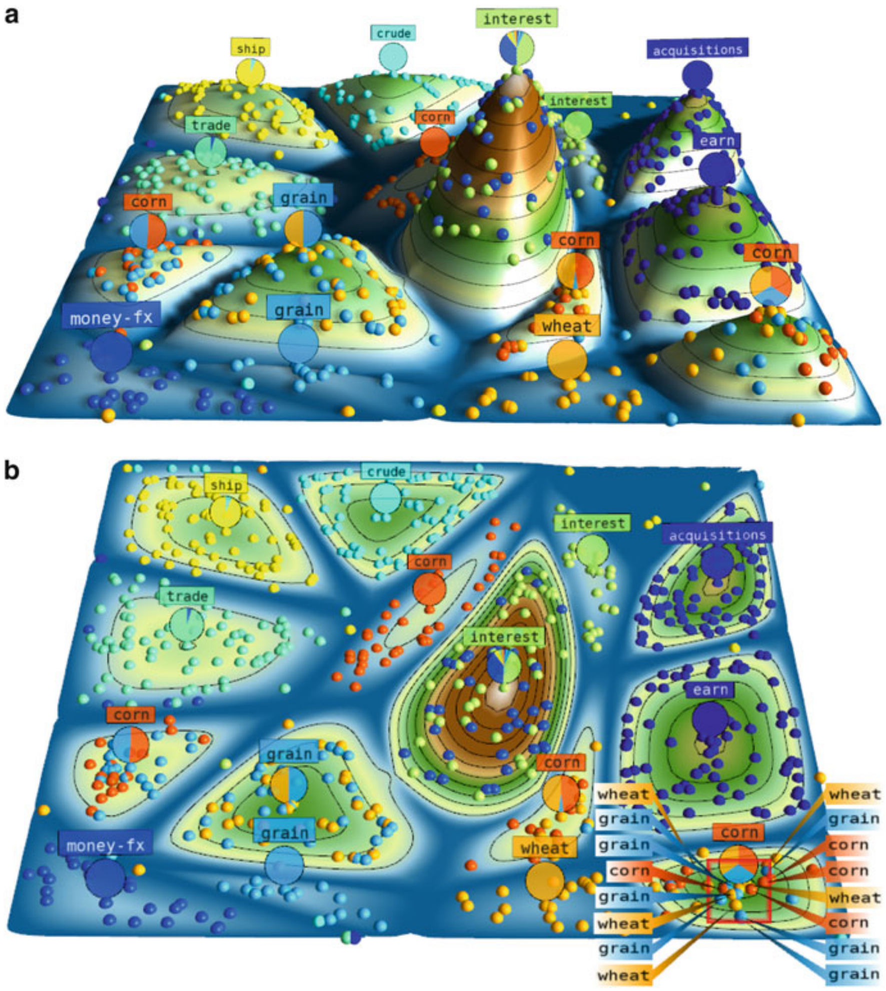


Fig. 7 Reuters data: (**a**) 3-D topological landscape with hypsometric tinting, *black isolines* and *pie charts* that illustrate the class-distribution and the maximum class for each hill. (**b**) Flattened atoll-like visualization with additional labeling to tag the data points

density function's join tree, which ignores the function's minima and is thus only a subset of the contour tree. It turns out that in contrast to the contour tree, a join tree can be visualized in two dimensions if we *unroll* the spiral layout from the 3-D landscape and put 2-D profiles of the hills next to each other. We obtain a *2-D topological landscape profile* as shown in Fig. 8. While a hill's height and width values still describe the dense region's respective persistence and size, the region's stability is now depicted by the hill's shape. That is, at each height, the width of a hill reflects the number of points that have at least this density, and therefore the shape accurately reflects the density distribution of the points. Histograms are used

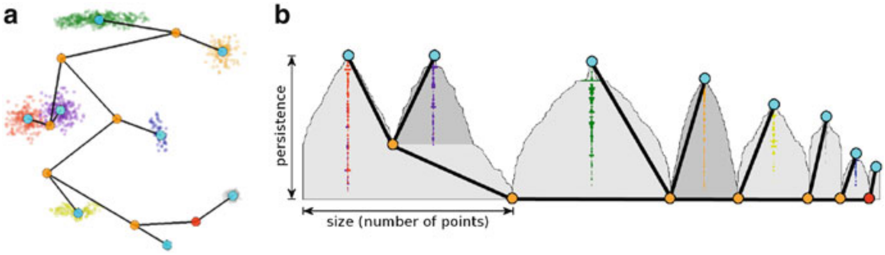


Fig. 8 Artificial data: (**a**) 2-D point cloud with clusters of varying size, shape, and compactness. The overlayed join tree indicates the critical points of the density function. (**b**) Topological landscape profile with the same topology of the height values like the input join tree. Extent and shape of the hills reflect join tree edge properties; histograms convey data point distributions per cluster

to augment the profile with the input data. The total length of all histograms on a hill is equal to the hill's size. In case of classified data, histograms are extended to stacked bar charts. Fractions of the histograms that represent individual data points can still be labeled with meta-information. Finally, the dual-color scheme of the profile indicates parent-child relations between nested hills.

The 2-D landscape profile has several advantages over the 3-D landscape: no view-dependent occlusion, no perspective distortion that complicates feature comparison, no invisible data points, less user interaction required to navigate through the scene, no strangely distorted base areas, easier and more accurate reading of width and height values to compare cluster properties, more compact depiction of data point distribution, a hill's shape to reflect stability, less complex geometry, faster construction scheme, and no expensive volumetric distortion. Drawbacks, on the other hand, are a little decrease in the intuitive expressiveness of profiles compared to natural hills in 3-D, and a little less efficient utilization of screen-space due to the left-to-right layout compared to the more compact spiral layout in 3-D.

5.3 Feature Selection and Local Data Analysis

The key advantage of the topology-based analysis over the attempt to preserve distances, and thus structure in the first place, is that the topology can be preserved without loss and thus features are also separated in the final visualization. As a consequence, the topological view on the data allows the user to select arbitrary structural entities like single clusters, sub-clusters, clusters or point sets of certain density, only the noise, everything but the noise, or points of single classes. Note that this is not easily possible in projections or axis-based techniques if features occlude each other or if noise covers everything. The ability to select individual features permits a linking of subsets to other views like projections or parallel coordinates plots. The idea behind this linking is a more thorough local analysis of selected features. Not only does the restriction to subsets improve the projection quality of, e.g., the shape of a single cluster or the spatial relation between a few clusters, also the overall visual complexity and clutter can be reduced if only a part of the data is visualized. If axis-based techniques are less cluttered, they efficiently disclose in which subspaces data points of separated regions actually differ. Even for non-classified data, the assignment of colors to single selection tremendously improves the local analysis if all points would otherwise be just monochrome. Figure 9 shows

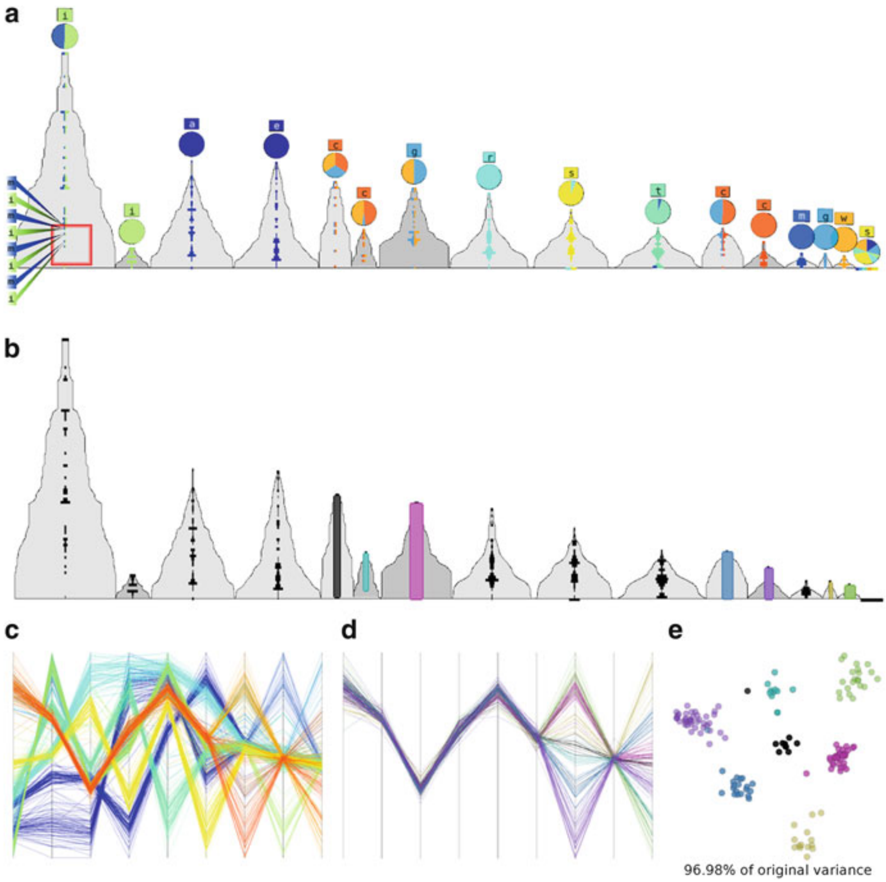


Fig. 9 Reuters data: (**a**) Topological landscape profile with labeling and *pie charts* above the hills to highlight class-distributions. (**b**) Even if there was no classification available, the structural view on the points would be the same. In this case, manual selections (*rectangles*) can be assigned with different colors to link them to other visualizations. Then the visual complexity of, e.g., (**c**) the PCP of the whole data set can be improved significantly if (**d**) non-selected points are suppressed and only the selected parts are displayed with their assigned colors. The PCP then easily reveals in which dimensions selected parts differ. (**e**) Also the quality of a PCA projection is increased if the optimization criterion is applied to only a subset of all data points

the Reuters data set as a landscape profile and demonstrates the advantage of linking only subsets and using colors per selection for unclassified data.

5.4 Parameter Widgets

We provide the user with two parameter widgets to set up the Gaussian filter radius and the simplification thresholds. Both controllers are shown in Fig. 10. The *filter radius suitability diagram* plots the ’suitability’ of the density function’s join tree against the filter radius σ used to create the function. More precisely, suitability is defined as the sum of all of the join tree’s edge stabilities normalized by σ. Note that stability is affected by persistence and size, but also considers the point distribution. A plot is obtained by calculating the suitability for different filter radii. The desired filter radius is then characterized by the local minimum of the function. The user finds the local minimum either by manually refining the plot or by an automatic divide-and-conquer strategy. For more details, we refer the interested reader to [25].

The *simplification controller* consists of three slider widgets, one for persistence, size and stability, that are augmented with little circles that correspond to the respective values for each branch of the join tree’s branch decomposition. Branches, and thus hills in the landscapes, with either of the values below the adjusted threshold are topologically simplified from the join tree. All sliders are linked, so if a branch is simplified in one slider, it also disappears in the other two. While topological noise is found at the bottom of the size- and the stability-slider, in the persistence diagram, noise resides near the diagonal. The simplification controller is also linked to the landscape, i.e. the landscape gets simplified in real-time if the user drags either of the sliders.

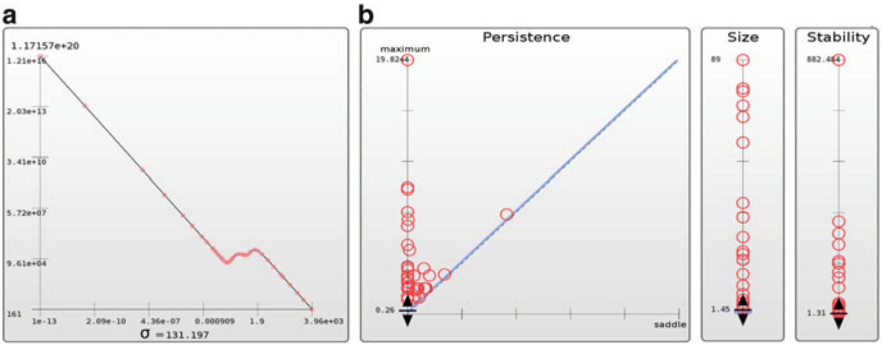


Fig. 10 Parameter widgets: (**a**) In the filter radius suitability diagram, the user interactively determines the optimal σ close to the local minimum. (**b**) The sliders in the simplification controller show the distribution of (sub-)clusters in terms of the three cluster properties. Dragging a slider triggers the simplification, followed by an update of the other sliders and the linked landscape

6 Conclusion

We reported on research results about the analysis and visualization of a high-dimensional point cloud's clustering structure—with an application to document collections (cf. Fig. 11). To visualize high-dimensional data without structural occlusion, we propose to neglect geometric properties, like distances, shape and position, and rather concentrate on clustering properties that can be extracted and preserved without loss in a 2-D/3-D illustration. The usage of topological concepts, here considering the topology of the point clouds density function, in combination with intuitive topology-based visualizations and the linking of subsets to geometry-based techniques constitutes an elegant alternative to those techniques that only try to reduce projection errors regarding some pre-defined optimization criteria and consequently rely on the data's benignity to produce useful illustrations of high-dimensional data. Of course, the extraction of topological information in arbitrary dimensions is not necessarily cheap. In fact, to obtain results within the range of seconds or at most minutes, data sizes and dimensionality are currently bound to a maximum of around 100,000 points in around 100 dimensions. Although bigger data sets could be processed by using more approximations, the curse of dimensionality eventually affects our analysis for very high-dimensional data because we also use a distance-based proximity measure. While this could be countered by the usage of metrics that are driven by natural language processing, we still opted for

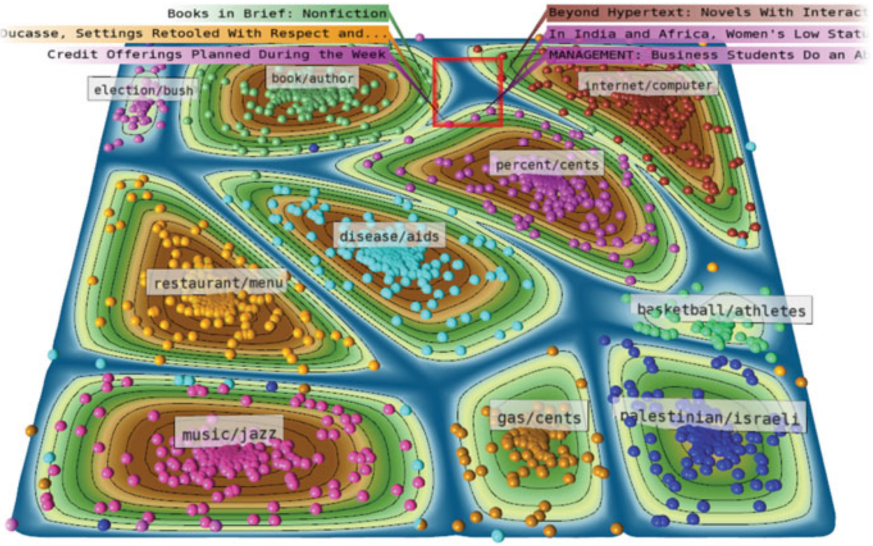

Books in Brief: Nonfiction
Ducasse, Settings Retooled With Respect and...
Credit Offerings Planned During the Week
Beyond Hypertext: Novels With Interac
In India and Africa, Women's Low Stat
MANAGEMENT: Business Students Do an A
election/bush
book/author
internet/computer
percent/cents
disease/aids
restaurant/menu
basketball/athletes
music/jazz
gas/cents
palestinian/israeli

Fig. 11 New York Times data: Islands represent topics in the collection of 1,896 articles corresponding to ten manually selected themes. Labels above each island indicate the most frequently shared words within the documents and labels for each data point show the article's title

an application-independent, multi-purpose approach that works on raw point data. Hence, for the application of topics in document collections, there is still potential for domain-specific optimizations regarding both data representation and the final visualization. We leave their exploration and investigation, the expansion to more clustering properties or even other structural features for future work.

References

1. Carr H, Snoeyink J, Axen U (2003) Computing contour trees in all dimensions. Comput Geom 24(2):75–94
2. Choo J, Bohn S, Park H (2009) Two-stage framework for visualization of clustered high dimensional data. In: IEEE VAST, IEEE, pp 67–74
3. Davidson GS, Hendrickson B, Johnson DK, Meyers CE, Wylie BN (1998) Knowledge mining with vxinsight: discovery through interaction. J Intell Inform Syst 11:259–285
4. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inform Sci 41(6):391–407
5. Edelsbrunner H, Letscher D, Zomorodian A (2002) Topological persistence and simplification. Dis Comput Geom 28(4):511–533
6. Elmqvist N, Dragicevic P, Fekete J-D (2008) Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation. IEEE Trans Vis Comput Graph 14(6):1539–1148
7. Fekete J-D, Plaisant C (1999) Excentric labeling: dynamic neighborhood labeling for data visualization. In: CHI '99: proceedings of the SIGCHI conference on human factors in computing systems
8. Fortune S (1997) Voronoi diagrams and Delaunay triangulations. In: Handbook of discrete and computational geometry. CRC Press, Boca Raton, pp 377–388
9. Gabriel RK, Sokal RR (1969) A new statistical approach to geographic variation analysis. Syst Zool 18(3):259–270
10. Hinneburg A, Aggarwal C, Keim DA (2000) What is the nearest neighbor in high dimensional spaces? In: Proceedings of the 26th international conference on very large data bases (VLDB'00). Morgan Kaufmann Publishers Inc., San Francisco, pp 506–515. http://dl.acm.org/citation.cfm?id=645926.671675
11. Holz F, Teresniak S (2010) Towards automatic detection and tracking of topic change. In: Gelbukh A (ed) Proceedings of CICLing 2010, Iai. LNCS, vol 6008. Springer, LNCS
12. Ingram S, Munzner T, Olano M (2009) Glimmer: multilevel mds on the gpu. IEEE Trans Vis Comput Graph 15:249–261
13. Inselberg A (2012) Parallel coordinates: visual multidimensional geometry and its applications. In: Fred ALN, Filipe J (eds) KDIR. SciTePress
14. Inselberg A, Dimsdale B (1990) Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: VIS '90: proceedings of the 1st conference on visualization '90, pp 361–378
15. Jaromczyk GT, Toussaint JW (1992) Relative neighborhood graphs and their relatives. Proc IEEE 80(9):1502–1517
16. John M, Chambers WS, Cleveland BK, Tukey PA (eds) (1983) Graphical methods for data analysis. The Wadsworth Statistics/Probability Series
17. Jolliffe IT (2002) Principal component analysis. Springer, New York
18. Jonathan KB, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful? In: International conference on database theory, pp 217–235
19. Kaski S, Honkela T, Lagus K, Kohonen T (1998) Websom-self-organizing maps of document collections. Neurocomputing 21(1):101–117
20. Kohonen T, Schroeder MR, Huang TS (2001) Self-organizing maps, 3rd edn. Springer, New York

21. Kruskal JB, Wish M (1978) Multidimensional scaling. SAGE Publications, Beverly Hills, London
22. Miller NE, Wong PC, Brewster M, Foote H (1998) Topic islands—a wavelet-based text visualization system. In: Proceedings of the conference on Visualization ’98 (VIS ’98). IEEE Computer Society Press, Los Alamitos, CA, pp 189–196
23. Oesterling P, Scheuermann G, Teresniak S, Heyer G, Koch S, Ertl T, Weber GH (2010) Two-stage framework for a topology-based projection and visualization of classified document collections. In: 2010 IEEE symposium on visual analytics science and technology (IEEE VAST), Utah, October 2010. IEEE Computer Society, pp 91–98
24. Oesterling P, Heine C, Janicke H, Scheuermann G, Heyer G (2011) Visualization of high-dimensional point clouds using their density distribution’s topology. IEEE Trans Vis Comput Graph 17(11):1547–1559
25. Oesterling P, Heine C, Weber GH, Scheuermann G (2013) Visualizing nd point clouds as topological landscape profiles to guide local data analysis. IEEE Trans Vis Comput Graph 19(3):514–526
26. Pascucci V, Mclaughlin KC, Scorzelli G (2005) Multi-resolution computation and presentation of contour trees, Lawrence Livermore National Laboratory. Technical report, in the proceedings of the IASTED conference on visualization, imaging, and image processing (VIIP)
27. Paulovich FV, Minghim R (2006) Text map explorer: a tool to create and explore document maps. In: 2013 17th international conference on information visualisation, pp 245–251
28. Paulovich FV, Oliveira MCF, Minghim R (2007) The projection explorer: a flexible tool for projection-based multidimensional visualization. In: Proceedings of the XX Brazilian symposium on computer graphics and image processing (SIBGRAPI ’07), Washington, DC. IEEE Computer Society, Los Alamitos, pp 27–36
29. Paulovich FV, Nonato LG, Minghim R, Levkowitz H (2008) Least square projection: a fast high-precision multidimensional projection technique and its application to document mapping. IEEE Trans Vis Comput Graph 14:564–575
30. Salton G, Buckley C (1987) Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY
31. Sammon JW (1969) A nonlinear mapping for data structure analysis. IEEE Trans Comput 18(5):401–409
32. Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47
33. Šilić A, Bašić BD (2010) Visualization of text streams: a survey. In: Setchi R, Jordanov I, Howlett RJ, Jain LC (eds) Knowledge-based and intelligent information and engineering systems. Lecture notes in computer science, vol 6277. Springer, Berlin, pp 31–43
34. Steinbach M, Ertöz L, Kumar V (2003) The challenges of clustering high-dimensional data. In: New vistas in statistical physics: applications in econophysics, bioinformatics, and pattern recognition
35. Teresniak S, Heyer G, Scheuermann G, Holz F (2009) Visualisierung von Bedeutungsverschiebungen in großen diachronen Dokumentkollektionen. Datenbank-Spektrum 31:33–39
36. Weber G, Bremer P-T, Pascucci V (2007) Topological landscapes: a terrain metaphor for scientific data. IEEE Trans Vis Comput Graph 13:1416–1423
37. Wise JA, Thomas JJ, Pennock K, Lantrip D, Pottier M, Schur A, Crow V (1995) Visualizing the non-visual: spatial analysis and interaction with information from text documents. In: Gershon ND, Eick SG (eds) INFOVIS. IEEE Computer Society, Los Alamitos, pp 51–58

Towards a Network Model of the Coreness of Texts: An Experiment in Classifying Latin Texts Using the TTLab Latin Tagger

Alexander Mehler, Tim vor der Brück, Rüdiger Gleim, and T. Geelhaar

Abstract The analysis of longitudinal corpora of historical texts requires the integrated development of tools for automatically preprocessing these texts and for building representation models of their genre- and register-related dynamics. In this chapter we present such a joint endeavor that ranges from resource formation via preprocessing to network-based text representation and classification. We start with presenting the so-called *TTLab Latin Tagger* (TLT) that preprocesses texts of classical and medieval Latin. Its lexical resource in the form of the *Frankfurt Latin Lexicon* (FLL) is also briefly introduced. As a first test case for showing the expressiveness of these resources, we perform a tripartite classification task of authorship attribution, genre detection and a combination thereof. To this end, we introduce a novel text representation model that explores the core structure (the so-called coreness) of lexical network representations of texts. Our experiment shows the expressiveness of this representation format and mediately of our Latin preprocessor.

1 Introduction

Unlike the *Digital Humanities* (DH), which mainly focus on the "creation, dissemination, and use" [1, p. 2] of resources of humanities data, the recent advent of *Computational Humanities* (CH) focuses on the explorative, mostly automatic analysis of this data [1]. By analogy to the notion of text mining [2] as a process of (semi-)automatic *abduction* [3], CH aims at computational models of interpretation processes that are still the domain of humanities scholars. A central consequence of this view is that CH requires models of the dual articulation [4] or even of the tripartite nature of signs [5]. That is, CH cannot reduce the analysis of signs to computer based models of their form.

A. Mehler (✉) • T. vor der Brück • R. Gleim • T. Geelhaar
Goethe-University Frankfurt, Frankfurt, Germany
e-mail: Mehler@em.uni-frankfurt.de; vorderBrueck@em.uni-frankfurt.de; Gleim@em.uni-frankfurt.de; Geelhaar@em.uni-frankfurt.de

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural
Language Processing, DOI 10.1007/978-3-319-12655-5_5

In any event, because of the manifold challenges of processing humanities data (which result from their sparseness and noisiness as well as from their temporal, genre- and register-related dynamics) an integrated formation *and* exploration of resources of this data is required. Obviously, such a joint endeavor does not allow for a labor division between humanities scholars and computer scientists. This can be exemplified by the recent success story of the computational analysis of longitudinal historical corpora of Greek and Latin Texts. In this area, the formation of digital resources like corpora [6], treebanks [7–9] and lexica [10] goes hand in hand with the build-up of corpus management-tools like eAQUA [11] or the eHumanities Desktop [12] and is continually accompanied by the development of text mining tools operating thereupon [13, 14].

Irrespective of these advancements, we are still far away from meeting the needs of humanists who ask for processing corpora with error rates less than 1%.[1] Thus, there is still the need of further developing tools and resources for historical languages. In this chapter, this is done with a focus on Latin.

Though there exist already taggers for Latin (cf. [15, 16]) as well as approaches to creating Latin lexica, we need a tagger that uses state-of-the-art methods of computational linguistics for tagging huge corpora of the size of the *Patrologia Latina* (PL) [17] under the control of knowledge about the morphosyntax of Latin. An integrated project is required that develops such a tripartite resource: a Latin tagger that combines statistical models with rule-based methods such that both are based on a user extensible lexicon which is easily extensible in order to cover a broad range of classical and medieval Latin. In this chapter, we describe efforts of building and applying such a resource. This is done by means of the *TTLab*[2] *Latin Tagger* (TLT) that operates on the *Frankfurt Latin Lexicon* (FLL), which is available via collex.hucompute.org and www.comphistsem.org. In order to range from resource formation to explorative data analysis, we apply this resource in the area of network-based text representation models [18–21]. To this end, we introduce a novel *Text Representation Model* (TRM) that explores the so-called core structure of text vocabularies. Our starting point is that the coreness of a text reflects organizational principles of its vocabulary that are both associated with principles of discourse comprehension and the discriminability of texts. From this point of view, it seems natural to apply our text representation model to text classification. This is exemplified by using the same corpus of medieval Latin texts as input of three classification tasks: authorship attribution, genre detection and a combination thereof. We show that our TRM is expressive regarding these tasks.

The chapter is organized as follows: Sect. 2 describes the TTLab Latin Tagger, its hybrid architecture based on combining *Conditional Random Fields* with linguistically informed rules about Latin morphology and also provides a first evaluation of it. Section 3 presents the FLL, the TTLab Latin Tagger is based upon. This is done by briefly describing how the FLL is continuously extended by exploring

[1] According to Anne Bohnenkamp-Renken, Goethe-House Frankfurt, personal communication.

[2] TTLab is an acronym that denotes the *Frankfurt Text-technology lab* (www.hucompute.org).

the Latin Wiktionary. The second half of the chapter concerns a first application of the TLT. To this end, Sect. 4 introduces a novel text representation model that explores the core structure of lexical text networks. In Sect. 5, this model is used to perform several classification tasks. This is done by comparing monastic sermons and letters with their early scholastic counterparts thereby focusing on the work of two classical authors: *Bernhard of Clairvaux* and *Petrus Abaelardus*. In this sense, the chapter ranges from resource formation, tool building via preprocessing historical documents to their network-based representation and classification. As a matter of fact, this is a labor-intensive procedure that most of the approaches to computational humanities still face when dealing with low-resourced historical languages. It is a merit of Gerhard Heyer who helped developing and shaping this novel research area as one of the first computational linguists.

2 Processing Latin Texts with the TTLab Latin Tagger

Current part of speech (PoS) taggers can be classified into rule-based and statistical taggers. Rule-based systems are developed by linguistic experts. A rule explores grammatical features of the mostly lexical context of the token to be tagged. There are more ambitious approaches that aim at parsing texts of the target language syntactically (e.g., [22, 23] for Latin). Using a parser, the PoS of a word is determined by parsing the sentence in which it occurs and by exploring the resulting parse tree [24]. If the input is ungrammatical, constructing a full parse tree may be out of reach so that tagging fails [25]. Generally speaking, a disadvantage of rule-based approaches is the effort they induce for every target language separately.

In order to overcome these and related problems, statistical approaches have been developed and are now predominant in linguistic preprocessing. Statistical preprocessors are mainly based on *Hidden Markov Models* [26], *maximum entropy models* [27], *structured SVMs* [28,29] or *Conditional Random Fields* (CRF) [30,31]. Statistical approaches face the risk of over-fitting as a result of too sparse or noisy training data. To overcome these and related problems, linguistic rules can be used in conjunction with statistical models [32]. In this section, we present such a *hybrid* part of speech tagger called *TTLab Latin Tagger* (TLT). It employs a statistical model together with linguistic rules for the automatic processing of texts of classical or medieval Latin. We know of no other system like this provided elsewhere.

In addition to manually annotated training corpora, the TLT uses the following linguistic resources:

- The *Frankfurt Latin Lexicon* (FLL) which we generated as a *collected* lexicon that covers a broad range of classical and medieval Latin.[3] The FLL, which is

[3] The FLL results from a cooperation of historians and computer scientists—see the project website for more information: www.comphistsem.org.

online available via the *ColLex.LA* website,[4] continues and extends the work of [12, 14, 33]. Currently, the FLL contains 249,379 lemmas together with their 11,212,223 wordforms each of which is mapped onto a range of morphological features. To the best of our knowledge, the FLL is currently the largest online lexicon of Latin. Its entries are primarily generated by automatically expanding its lemmas morphologically. These lemmas were mainly extracted and collected from various web-based resources. This includes the AGFL Grammar Work Lab [22], the Latin morphological analyzer LemLat [34], the Perseus Digital Library [15,16], William Whitaker's Words,[5] the Index Thomisticum [7,9], Ramminger's Neulateinische Wortliste,[6] the Latin Wiktionary,[7] Latin training data of the TreeTagger [35], the Najock Thesaurus[8] (as it is called here) and several other resources. Beyond that, the FLL is continuously manually checked, corrected and updated by historians and other researchers from the humanities.[9]

- Besides the FLL, the TLT exploits a set of Latin morphological rules for predicting the PoS and lemma of the word to be tagged (see Sect. 2.1). When being applied, the rules provide grammatical information about the context of the word.

The TLT is based on *Conditional Random Fields* (CRF) [30]. As a hybrid approach, the TLT tags words by means of additional modules. In the order of descending priority, this includes the following ones:

1. Linguistic rules: whenever one of our linguistic rules applies, it is preferred (see Sect. 2.1).
2. Numbers: a module is applied that recognizes Latin and Arabic numerals.
3. Lexicon: If the FLL lists exactly one PoS for the input token, then it is assigned without further processing.
4. CRF: if the FLL does not list the input token or if it is a homograph that is connected with different PoS, the CRF-model is employed (see Sect. 2.2). Note that for any word known by the FLL, only those PoS are considered that the FLL lists for the wordform.

2.1 Linguistic Rules

In order to overcome limitations of the CRF-model, we apply several morpho-syntactical rules. Each rule consists of a premise and a conclusion, which contains

[4] collex.hucompute.org.

[5] archives.nd.edu/whitaker/dictpage.htm.

[6] ramminger.userweb.mwn.de.

[7] la.wiktionary.org/wiki/Pagina_prima.

[8] Provided by Michael Trauth, Trier University. See also [36].

[9] collex.hucompute.org. is our interface for this human computation of a Latin resource.

the suggested PoS and optionally a lemma. The condition part of a rule is a regular expression which consists of feature structures to be matched with the lexicon entries mapped to the input tokens. In order to avoid time-consuming recursions, we always prefer the longest match of the regular expression. The following operators (of regular-expressions) are supported:

1. Sequence: the sequence operator requires that all of its components match in the input sequence.
2. Disjunction: the disjunction operator is evaluated to be true, if at least one of its arguments is true.
3. Plus(+): the plus operator requires that its argument matches at least once.
4. Star(*): the star operator requires that its argument matches zero or several times.
5. Question mark(?): this operator specifies optional arguments.
6. Negation: this operator implements a negation over matchings.
7. Feature structure: a feature structure contains lexical or grammatical features to be matched with the lexicon entries of the FLL mapped onto the input tokens. If a token of the input is mapped onto several lexicon entries, we require that the feature structure is unifiable with at least one of these entries in order to be applied.
8. Attribute negation: feature structure values can be negated. In this case the feature structure element must not be matchable with the associated attribute of the pattern.

The top-level operator of a rule is a sequence that starts with the token to be processed. It tries to match with the input sequence by processing it forward or backward.

Currently, we use 13 rules. Five of them are exemplified below (the character '#' marks the token to be tagged):

$$R_1 : \#[mood : \text{`}participle\text{'}][lemma : \text{`}sum\text{'}] \rightarrow pos = \text{`}VPP\text{'} \tag{1}$$

The input is tagged as *VPP* (participle perfect) if one of the associated moods is *participle* and followed by a form of *esse* (lemma is *sum*).

$$\begin{aligned} R_2 :&\#[word\_form : \text{`}ut\text{'}][not\, pos : \text{`}V\text{'}]^* \begin{bmatrix} pos & : \text{`}V\text{'} \\ mood & : \text{`}infinitive\text{'} \end{bmatrix}^* \\ &\begin{bmatrix} pos & : \text{`}V\text{'} \\ mood & : \text{`}subjunctive\text{'} \end{bmatrix} \rightarrow pos = \text{`}CON\text{'} \end{aligned} \tag{2}$$

If the word *ut* is followed by a verb with mood *subjunctive* (between both occurrences there may occur several non-verbs or verbs of mood *infinitive*) then its PoS is *conjunction*.

$$\begin{aligned} R_3 : \#[word\_form : \text{`}cum\text{'}][not\ case : \text{`}ablative\text{'}]^?[case : \text{`}ablative\text{'}] \\ \rightarrow pos = \text{`}AP\text{'} \end{aligned} \tag{3}$$

If the word *cum* is followed by a noun of case *ablative* (such that there is at most one word of a non-ablative case between them), its PoS is *preposition*.

$$\begin{gathered}R_4 :\#[word\_form : \text{‘}cum\text{’}][not\; pos : \text{‘}V\text{’}]^* \begin{bmatrix} pos & : \text{‘}V\text{’} \\ mood & : \text{‘}infinitive\text{’} \end{bmatrix}^? \\ \begin{bmatrix} pos & : \text{‘}V\text{’} \\ mood & : \text{‘}subjunctive\text{’} \end{bmatrix} \rightarrow pos = \text{‘}CON\text{’}\end{gathered} \tag{4}$$

If the word *cum* is followed directly (though not necessarily) by a verb of mood *subjunctive*, its PoS is *conjunction*.

$$R_5 : \#[word\_form : \text{‘}vel\text{’}][word\_form : \text{‘}ut\text{’}] \rightarrow pos = \text{‘}ADV\text{’} \tag{5}$$

If the input word *vel* is followed by *ut*, its part of speech is *adverb*.

Currently, we work on extending our set of morpho-syntactical rules in order to provide a controlled linguistic basis for reducing the errors of our statistical tagger. Such a hybrid approach seems to be indispensable in order to reach the extremely high level of tagging correctness required by humanities scholars.

2.2 Statistical PoS-Tagging with Conditional Random Fields

In addition to linguistic tagging rules, the TLT applies a CRF [30]. In contrast to the Maximum Entropy model, CRFs do not have the so-called *Label Bias Problem*. This means that in a Maximum Entropy model, the decision about assigning a tag to a word can be completely independent of the input word.

For training and tagging we use the CRFsuite [37] which implements first order CRFs using the limited memory BFGS (Broyden-Fletcher-Goldfarb-Shanno) training method. The following features are currently explored to determine the most likely PoS of the input:

1. Capitalization: a binary feature that is true if the first letter is upper-case. In Latin, capitalized words are either named entities or adjectives derived from toponym.
2. Word length: this feature refers to Zipf's law [38] according to which the length of a word is inversely proportional to its usage frequency. Frequent short words are usually function words, while rare long words are likely nouns.
3. Suffix n-grams: in inflecting languages, suffixes inform about the grammatical features of a token and, thus, about its PoS.
4. Letter-based Markov models: a letter-based word-intrinsic Markov Model is additionally applied to cover word-internal morphological information.
5. Lemma n-grams: all candidate lemmas in the FLL are identified for all words in a text window around the focal token. If this mapping is unique for all positions in the window, the resulting lemma sequence is used as a feature.

6. Part of speech n-grams: by analogy to lemma n-grams, we explore PoS n-grams that are uniquely assignable onto the text window around the focal token.
7. Numbers: Latin numbers are recognized by means of a regular expression, Arabic numbers by applying the clib-function *atoi*.

Note that while the rules of Sect. 2.1 result from an error analysis of the output of our tagger (to map those structures that are hardly mapped by a CRF), the features of the latter enumeration have been selected to map a wide range of word internal and external characteristics that are easily accessible by a CRF. We are currently working on extending both sets of rules and features to make our tagger more robust.

2.3 Evaluation

We evaluate the TLT in one experiment. The training and test sets are taken from the Capitula corpus.[10] This corpus was tagged manually[11] so that the resulting tags provide a gold standard. We conducted a ten-fold cross-validation with respect to PoS tagging and lemmatization. The evaluation results are shown in Table 1. Note that we apply two well-known methods of computing F-scores here for PoS-tagging: *micro-averaging* first builds a contingency over all events (tokens or wordforms) to be categorized and then calculates the F-score as the harmonic mean of the corresponding precision and recall. In contrast to this, *macro-averaging* calculates the F-score for each target class (in our case PoS) separately and then computes the arithmetic mean over all category-specific scores. Both kinds of F-scores are shown in Table 1.

We see that the micro-averaged F-score is remarkably high in the case of tagging the Capitula corpus, while it drops to 86 % in the case of calculating the macro-averaged F-score. Obviously, the distribution of tagging errors over the PoS is somehow skew. Making a brief error analysis, we detect that most errors concern

Table 1 Evaluation of the TLT based on the Capitula corpus and a ten-fold cross-validation

Measure	Value
Micro-averaging precision	0.935
Micro-averaging recall	0.935
Micro-averaging F-score	0.935
Macro-averaging precision	0.894
Macro-averaging recall	0.834
Macro-averaging F-score	0.863

[10] www.dmgh.de/de/fs1/object/display/bsb00000820_meta:titlePage.html?sortIndex=020:030:0001:010:00:00.

[11] By the members of Bernhard Jussen's lab at Goethe-University Frankfurt: Silke Schwandt, Tim Geelhaar, and colleagues.

adjectives that are erroneously tagged as nouns. A second class of errors concerns nouns that are wrongly tagged as adjectives. Adverbs that are tagged as adjectives or verbs that are mapped to nouns or past participles form two additional groups of errors. Further, by examining the tagged documents we see that the lemmatization is far from being perfect. From these observations, we conclude that the TLT and the FLL still require extensive extensions and improvements.

3 Extending the Frankfurt Latin Lexicon (FLL)

In this section, we briefly report ongoing efforts on extending the FLL as part of the TLT. Beyond manual corrections and updates, this is done by means of exploring a range of machine-readable resources. This section exemplifies how new lexical entries are automatically extracted from the Latin Wiktionary.

Victionarium,[12] the Latin Wiktionary, is an online lexicon provided by the Wikimedia Foundation. It adheres to the Wiki principle and, thus, is generated by volunteers. There are two ways in Victionarium to specify a paradigm for a lemma. Firstly, each conjugated or declined form can be specified separately. This approach is suited for irregularly inflected wordforms. The second method is to specify the base form or stem (several stems in the case of verbs) together with the inflection class. Using this information, the wordforms of the lemma are automatically generated. Note that a verb, its stem of indicative present, indicative perfect and the supina must be specified in order to generate the full paradigm. The conjugation class can take the values *first*, *second*, *third* or *fourth conjugation*. Special conjugation classes exist for irregular verbs and a lot of variants are handled in cases where, for example, passive verb forms or supina do not exist or where vowels are inserted (cf. [39]). In the case of nouns, five declension classes are considered together with Greek declension classes that are used to map Greek loan words (e.g. *argon*, *neon* or *xenon*).

In order to extract words from Wiktionary, we parse its database dump using a SAX parser. The lemma of a lexical entry is specified inside the *title*-tag, which is a child of the associated *page*-tag. Lexical information about the PoS of a word and its inflection class is provided by templates as part of its *text*-tag. The templates for the verbs *video* and *audio* look as follows:

(1) video: {{coniugatio-2|vid|vĭd|vid|vīd|vis|vīs}}

(2) audio: {{coniugatio-4|aud|aud|audiv|audīv|audit|audīt}}

The conjugation class is given by the first token: here *4nd conjugation* for the verb *audio* and *2th conjugation* for the verb *video*. The verb stems (with and without stress marks) are given subsequently. The second and the third token provide stems

[12] la.wiktionary.org/wiki/Pagina_prima

Table 2 Number of wordforms and lemmas for different parts of speech

Part of speech	Word forms	Lemmas
Verb	59,764	281
Noun	15,758	1,238
Adjective	13,882	276
Adverb	142	138
Named Entity	600	81
All	90,147	2,014
All*	40,878	1,992

All*: wordforms and lemmas belonging to different parts of speech are counted only once

for present indicative, the fourth and the fifth for perfect indicative, while the last two tokens contain stems for the supina.

In total, we extracted 90,147 wordforms of 2,014 lemmas from the Victionarium by exploring templates of the latter sort (see Table 2). If we count wordforms only once that belong to different parts of speech or grammatical forms, this number drops to 40,878 wordforms of 1,992 lemmas.

A central advantage of exploring the Victionarium for extending the FLL is that it is continuously updated by volunteers. However, for each web-based resource to be integrated into the FLL, a specialized module needs to be built in order to process its data. Since we aim at covering the whole range of classical and medieval Latin, we see no alternative to this approach.

4 From Tagging Latin Texts to Lexical Text Networks

In order to test the usefulness of the TLT, we perform a classification experiment that explores lemma-based lexical text networks. More specifically, for a set of medieval Latin texts (see Sect. 5), we induce vectors of graph invariants that capture structural information about the text's lexical organization. This done by looking for preferably small vector representations (ideally of only two dimensions). The aim is to perform a sort of *representational minimization*. That is, for a given task in text mining, we seek a text representation model that is as compact or small as possible while still producing high F-scores. Other than, for example, the *Bag-of-Words* (BoW) approach, which explores hundreds and thousands of dimensions, we look for models of a dozen or even less dimensions.[13] This approach is in line with Ockham's razor *with a focus on the space complexity of the text representation model.* We aim at minimizing this space complexity by reflecting structural while

[13] Models of text representation as small as the one introduced here are common in quantitative text linguistics—an early example is [40]. The difference is that while these models mostly consider well-established indices like TTR or the rate of hapax legomena, we are concerned with inventing a complete new set of quantitative text characteristics based on the same notion of text organization.

ignoring, unlike the BoW approach, content-related features. Note that BoW-based models require large corpus data and, thus, may run into difficulties when dealing with historical texts of low-resourced languages. This is another reason to look for compact text representation models as done here. Our approach of modeling document structures by means of graph invariants complements approaches that refer directly to graph models of (e.g., semantic) document structure to compute similarities of texts (e.g., by means of some graph kernel; cf. [41]). It would be an interesting alternative (not considered here) to integrate both approaches.

In order to realize downscaling of model size, we introduce a novel model of structurally characterizing lexical text networks that explores their decomposition into so-called k-cores [42, 43]. The reason to proceed in this way is as follows: firstly, remarkably few approaches explore the coreness of lexical networks—see, for example, [44] who explore cores of co-occurrence networks of graphemes—though there is a growing set of publications on lexical text network analysis [18–21]. So far, cores are mainly explored for reasons of community detection in social networks [45, 46]. Secondly, the coreness of a network provides a coherent framework for deriving structural invariants that address the same macroscopic dimension, that is, the way networks are decomposable into cores. By analyzing principles of this decomposition on the level of lemma-based co-occurrence networks we get a testbed for our Latin preprocessor. Note that the induction of lexical networks heavily hinges upon the quality of the underlying preprocessor[14]: If a wordform is wrongly lemmatized, a false vertex is added or updated in the network to be induced. Obviously, lemmatization has a huge impact on lexical network induction. Thus it makes sense spending much effort on creating resources like the TLT and the FLL in order to support the novel research branch of network-based text models.

4.1 Approaching Lexical Text Structures by Means of k-Cores

In this section, we present our text representation format based on graph invariants that describe the core structure of text vocabularies. The basic graph-theoretical notion explored for this task is that of a k-core. k-cores are used to study the macroscopic core-periphery structure of complex networks [47]. We use the formalism of [43, 47] and of [42] to define k-cores and related graph-theoretical concepts.

For an undirected graph $G = (V, E)$, a *k-core* or *core of order k* is a maximal subgraph $C_k = (W_k, E_k)$, $E_k = E|_{W_k}$, of G induced by $W_k \subseteq V$ such that $\forall w \in W_k : deg(w) \geq k$ [43]. While cores are nested ($0 \leq i < j \Rightarrow W_j \subseteq W_i$), they are not necessarily connected [43]. Since *deg* can denote in-degree, out-degree or both (in the case of undirected graphs or directed graphs with multiple arcs), we need to specify our perspective. In the present chapter, we focus on *undirected* graphs.

[14] A test of this hypothesis will be the object of a forthcoming paper.

When applied to a lemma-based co-occurrence network derived from a *single* text, one may state—by analogy to the redundancy hypothesis of [42, p.274]: *the higher the order of a core of a text, the more likely its lexical nodes are co-activated when reading it*. In this sense, the core-periphery structure of a *Lexical Text Network* (LTN) as mapped by its *core structure* or *coreness* may reflect the organization of the text's vocabulary. Membership to cores is then correlated with the probability to be activated and co-activated. If this is valid, texts should be classifiable (e.g., for their authorship, genre or topic) by analyzing certain aspects of the coreness of their LTN-based representations. This hypothesis about *the expressiveness of the coreness of text vocabularies* is the starting point of our classification experiment in Sect. 5. It states that texts can be distinguished by their core structure and that the respective differences are caused by some (though yet unknown) parameters of the authorship, genre or register [48] of a text. Though the chapter will shed some light on this hypothesis, its aim is rather preparatory in inventing and experimenting with a range of graph invariants for characterizing the coreness of a text.

Starting from the notion of a k-core, we get access to a range of substructures (as exemplified in Fig. 1) that help characterizing the topology of a (e.g., lexical) network:

1. The *shell index* (or *core number*) $\sigma(v)$ of a vertex $v \in V$ is the core number k such that $v \in W_k \wedge v \notin W_{k+1}$ [47].
2. $S_k = \{v \mid \sigma(v) = k\} = C_k \setminus C_{k+1}$ is the k-*shell* of G [47]. The maximum value of k for which $S_k \neq \emptyset$ is denoted by $k_{\max}$.
3. By $G_k = (S_k, E|_{S_k})$ we denote the maximal *subgraph of G induced by its k-shell* S_k.

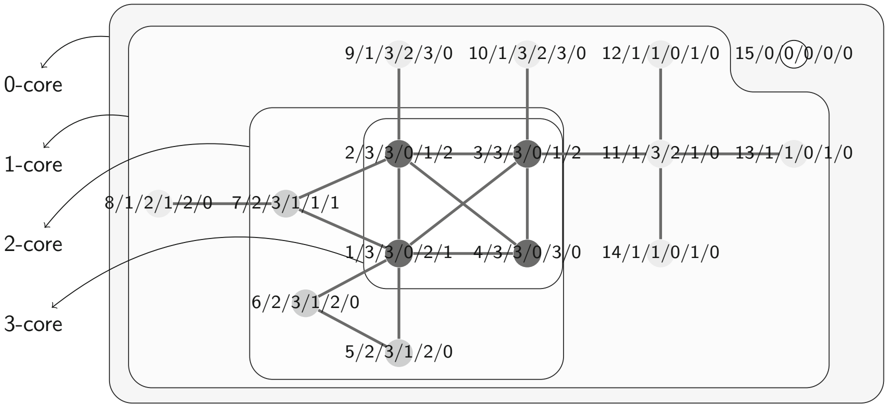


Fig. 1 A sample graph taken from [47] and extended in order to demonstrate graph-theoretical concepts introduced here to characterize the coreness of a graph. Each vertex is mapped onto a separate feature vector showing in sequential order: (1) the ID of the vertex, (2) its shell index, (3) upward range, (4) upward distance, (5) downward range and (6) downward distance

4. The *frontier* $F(W) = \{v \in V \setminus W \mid \exists w \in W : \{v, w\} \in E\}$ of a subset of vertices $W \subseteq V$ is the set of all vertices in $G - W$ that are adjacent with vertices in W [42].
5. The *k-frontier* $F_k = F(S_k)$ is the frontier of the k-shell S_k.

Cores, shells and frontiers provide different access points of characterizing the coreness of a network. This characterization will mainly be done with the help of graph entropies as introduced by [49, 50] together with the Kullback–Leibler divergence of the probability distributions involved. In a range of experiments, Dehmer and colleagues have shown that graph entropies are valuable sources of characterizing networks structurally (cf. [51]). In line with this approach, we define a set of probability distributions over the vertex sets of graphs that are finally input to graph entropy measurements. This is done with the help of three numbers:

6. $k_{\mathrm{Min}} = \min\{\arg\min_i \{|S_i| \mid i \in \{0, \dots, k_{\max}\}\}\}$ is the minimum of the core numbers of all shells of minimum order.
7. $k_{\mathrm{Med}} = \arg\operatorname{median}_i \{|S_i| \mid i \in \{0, \dots, k_{\max}\}\}$ is the core number of the shell of median order.
8. $k_{\mathrm{Max}} = \max\{\arg\max_i \{|S_i| \mid i \in \{0, \dots, k_{\max}\}\}\}$ is the maximum of the core numbers of all shells of maximum order.

These three notions allow for deriving quotients regarding the core formation in networks (see Table 3) with a focus on *vertices*. They consider the number of vertices in the shells of smallest, largest and median size in relation to the order of the network and, hence, provide location parameters of the distribution of shell sizes. In Table 3, we add two invariants (ID 4 and 5) considering the partition of *edges* into intra- and inter-shell edges. These location parameters give insight into whether the network has a denser shell-internal structure compared to its shell-external one.

9. The *upward range* $u(v) = \max\{\sigma(w) \mid \{v, w\} \in E\} \in \{0, \dots, k_{\max}\}$ of a vertex $v \in V$ is the highest shell index of its neighboring vertices. For isolated vertices, we assume that $u(v) = 0$.
10. $U_k = \{v \in V \mid u(v) = k\}$ is the set of all vertices with an upward range of k.
11. The *upward distance* $\delta_{\mathrm{up}}(v) = u(v) - \sigma(v) \in \{0, \dots, k_{\max} - 1\}$ of a vertex $v \in V$ is the core distance to its upward range.[15]
12. $\triangle_k = \{v \in V \mid \delta_{\mathrm{up}}(v) = k\}$ is the set of all vertices with upward distance k.
13. The *downward range* $d(v) = \min\{\sigma(w) \mid \{v, w\} \in E\} \in \{0, \dots, k_{\max}\}$ of a vertex $v \in V$ is the smallest shell index of its neighboring vertices. For isolated vertices, we assume that $d(v) = 0$.
14. $D_k = \{v \in V \mid d(v) = k\}$ is the set of all vertices with a downward range of k.
15. The *downward distance* $\delta_{\mathrm{down}}(v) = \max\{\sigma(v) - d(v), 0\} \in \{0, \dots, k_{\max} - 1\}$ of a vertex $v \in V$ is the core distance to its downward range.

[15] Obviously, for any v: $u(v) \geq \sigma(v)$.

Table 3 Feature Model for characterizing LTNs decomposed into nested k-cores. $H(\cdot)$ denotes Shannon's entropy, $D(\cdot)$ the Kullback-Leibler divergence

ID	Model	Short description
1.	$V_{k_{\text{Min}}} = \|S_{k_{\text{Min}}}\|/\|V\|$	Fraction of vertices in the shell of minimum order
2.	$V_{k_{\text{Med}}} = \|S_{k_{\text{Med}}}\|/\|V\|$	Fraction of vertices in the shell of median order
3.	$V_{k_{\text{Max}}} = \|S_{k_{\text{Max}}}\|/\|V\|$	Fraction of vertices in the shell of maximum order
4.	$E_{\text{intra}} = \|\{\{v, w\} \mid \sigma(v) = \sigma(w)\}\|/\|E\|$	Fraction of intra-shell edges
5.	$E_{\text{inter}} = \|\{\{v, w\} \mid \sigma(v) \neq \sigma(w)\}\|/\|E\|$	Fraction of inter-shell edges
6.	$H(C)$	Entropy of the relative core sequence
7.	$H(S)$	Entropy of the relative shell sequence
8.	$H(F)$	Entropy of the relative frontier sequence
9.	$D(S\|\|C)$	KL-divergence of shell and core sizes
10.	$D(C\|\|S)$	KL-divergence of core and shell sizes
11.	$D(S\|\|F)$	KL-divergence of shell and frontier sizes
12.	$D(F\|\|S)$	KL-divergence of frontier and shell sizes
13.	$H(U)$	Entropy of the sequence of upwards ranges
14.	$H(D)$	Entropy of the sequence of downwards ranges
15.	$D(U\|\|D)$	KL-divergence of up- and downward ranges
16.	$D(D\|\|U)$	KL-divergence of down- and upward ranges
17.	$H(\triangle)$	Entropy of the sequence of upwards distances
18.	$H(\bigtriangledown)$	Entropy of the sequence of downwards distances
19.	$D(\triangle\|\|\bigtriangledown)$	KL-divergence of up- and downward distances
20.	$D(\bigtriangledown\|\|\triangle)$	KL-divergence of down- and upward distances
21.	$\kappa(G)$	Compactness of G
22.	$\kappa(G - E_{\text{Max}})$	Compactness of G without edges in E_{Max}
23.	$\delta_\kappa = \kappa(G) - \kappa(G - E_{\text{Max}})$	Loss of compactness induced by $E \setminus E_{\text{Max}}$
24.	$\kappa(G_{\text{Max}})$	Compactness of the shell of maximum order
25.	$\pi(G)$	Closeness of G
26.	$\pi(G - E_{\text{Max}})$	Closeness of G without edges in E_{Max}
27.	$\delta_\pi = \pi(G) - \pi(G - E_{\text{Max}})$	Loss of closeness induced by $E \setminus E_{\text{Max}}$

16. $\bigtriangledown_k = \{v \in V \mid \delta_{\text{down}}(v) = k\}$ is the set of all vertices with downward distance k.
17. $C = (|W_0|/\sum_{i=0}^{k_{\max}} |W_i|, \ldots, |W_{k_{\max}}|/\sum_{i=0}^{k_{\max}} |W_i|)$ is the *(relative) core sequence*.
18. $S = (|S_0|/|V|, \ldots, |S_{k_{\max}}|/|V|)$ is the *(relative) shell sequence*.
19. $F = (0, |F_1|/\sum_{i=0}^{k_{\max}} |F_i|, \ldots, |F_{k_{\max}}|/\sum_{i=0}^{k_{\max}} |F_i|)$ is the *(relative) frontier sequence* (note that the frontier of S_0 is necessarily empty).
20. $U = (|U_0|/|V|, |U_1|/|V|, \ldots, |U_{k_{\max}}|/|V|)$ is the *sequence of (relative) upward ranges*. $U_0 = S_0$ is added, since its cardinality may vary for different networks.
21. $D = (|D_0|/|V|, |D_1|/|V|, \ldots, |D_{k_{\max}}|/|V|)$ is the *sequence of (relative) downward ranges*.

22. $\triangle = (| \triangle_0 |/|V|, \ldots, | \triangle_{k_{\max}-1} |/|V|)$ is the *sequence of (relative) upward distances*.
23. $\triangledown = (| \triangledown_0 |/|V|, \ldots, | \triangledown_{k_{\max}-1} |/|V|)$ is the *sequence of (relative) downward distances*.

The sequence of cores, shells and frontiers is made input to measuring graph entropies and relative entropies (using the Kullback–Leibler or KL-divergence)—see the graph invariants with IDs 6–12 in Table 3. By these measurements, we aim at getting insight into the distribution of the coreness of a network. The invariants under consideration can distinguish networks, whose shell sizes are evenly distributed, from those that follow skew size distributions. The relative entropy of two distributions computes the amount of information that one gets about the first distribution when knowing the second one. According to expectation, the KL-divergence of the distribution of shell and core sizes is low. However, a large k-shell (in relation to the order of a graph) does not need to be accompanied by a large k-core if most vertices belong to shells of order $i \ll k$. Thus, even small differences in the relative entropy of shell and core sizes may characterize authors, styles, registers or genres so that we include this invariant into our experiment. Likewise, by measuring the relative entropy of shell and frontier size per shell index, we aim at distinguishing networks, in which the size of a shell is proportional to the size of its frontier, from those where this relation is inversely proportional.

Rather than on the size of cores, shells or frontiers, the invariants with IDs 13–20 (see Table 3) focus on the interconnectedness of shells. This is done in terms of down- and upward ranges and the respective distribution of down- and upward distances—measured by the number of intervening cores—between the shell of the focal vertex and its most distant core adjacent to it. This is the starting point of deriving distributions of down- and upward ranges and distances that allow for distinguishing networks by the ranges of inter-shell links. Note that in order to compute the corresponding KL-divergences (ID 15, 16, 19, 20) of the sequence of upward and downward ranges, we add probability mass to the respective sequences (see above). This is done to circumvent situations in which because of $p \log \frac{p}{0} = \infty$ for $p > 0$ (cf. [52, p.19]) the value of the respective KL-divergence is infinite.

The invariants with IDs 21–27 finally regard the cohesion of a network and its cores. This is done with the help of the compactness measure of hypertext theory [53] and a modified geodesic distance that is transformed and normed into a closeness (or proximity) measure $\pi : V^2 \rightarrow [0, 1]$ by regarding every pair of vertices—whether connected or not. Note that instead of using the compactness measure of [53] directly, we use a variant that better scales in $[0, 1]$ for connected graphs. The idea of using these invariants is to study the impact of the shell of maximum order on the compactness and closeness of a network when being deleted. In terms of LTNs, this informs about the structural role of those lexical items of a text that are most likely (co-)activated when reading it. Our hypothesis is that invariants focusing on the role of lexical items may help distinguishing networks. The compactness of $S_{k_{\text{Max}}}$ is a measure of its connectedness: *the more disconnected the clusters of this shell, the more different lexical centers (in the sense of $S_{k_{\text{Max}}}$) are found in the underlying text.* This may hint at thematic broadness; it may also hint at a detailed, branching argumentation as exemplified by scholastic sermons in contrast to monastic ones.

Table 3 enumerates all the 27 graph invariants introduced or reused here (as in the case of the compactness measure) to characterize the coreness of an LTN numerically. Applied to LTNs, they represent each input text as a 27-dimensional feature vector that is input to *Quantitative Network Analysis* (QNA) [51,54,55].

5 Experimentation

In this section, we present an experiment based on the text representation model of Sect. 4. This is done in the framework of text categorization with a focus on three tasks: (1) *authorship attribution*, (2) *genre detection* and (3) *genre-sensitive authorship detection*. To this end, we start from the following classification hypotheses:

1. Authors differ in the way they organize their text vocabularies [56]. These differences are reflected by the core structure of the LTNs of their textual output.
2. Genres differ in the way authors organize the vocabularies of their text instances [48]. These differences are reflected by the core structure of the texts' LTNs.
3. Authorship and genre are reflected by different organizational principles of text vocabulary such that the authorship of a document is reflected by other properties of lexical organization than its genre.

Note that these hypotheses do *not* regard a combined (e.g., thematic *and* generic) classification. This is problematic whenever the corpus under consideration manifests several classification dimensions. In contrast to hypotheses (1–3), we may ask, for example, for simultaneously distinguishing between *monastic sermons* of one author in contrast to *scholastic sermons* of another author.[16] To this end, we additionally consider a joint classification task in which we simultaneously classify texts for their authorship *and* their genre. This is done starting from the following hypothesis:

4. Authorship- and genre-specific lexical features can be explored to simultaneously classify texts along both of these dimensions.

Since the present study deals with Medieval Latin texts in the context of the opposition of monasticism and (pre-)scholasticism and since this opposition correlates in our corpus with authorship (as explained below), we can finally reformulate our hypothesis as follows:

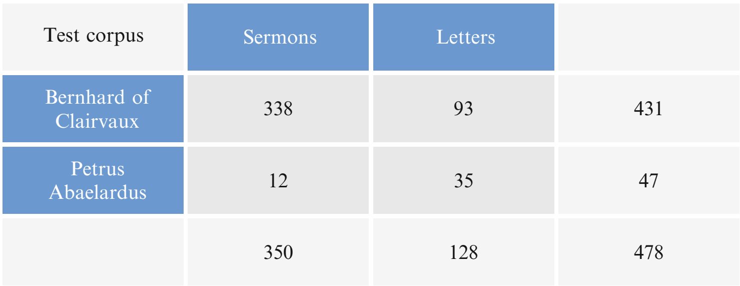
5. Monastic instances of given genres (e.g., letters or sermons) and a given author can be distinguished from their scholastic counterparts of another author in terms of their lexical organization.

[16] For linguistic indicators of the monastic-scholastic distinction see [57,58].

Hypotheses (5) draws upon the observation that unlike monastic sermons, for example, their scholastic counterparts have a much more elaborated discourse structure: while monastic sermons show a more narrative and commentarial form, scholastic texts elaborate delicate argumentations [57, 58]. The monastic sermon is basically structured as a succession of biblical quotations and commentaries (together with introducing/concluding remarks used by the sermonizer to address his audience) [57, 58]. In contrast to this, scholastic sermons are based on complex macro structures manifesting detailed argumentations. Further, they may contain incomplete biblical quotations together with elaborate enumerations of competing interpretations of biblical terms [57, 58]. Based on these observations, we expect very different lexical organizations in texts of both types: like scientific documents, scholastic texts are probably lexically richer while they interlink lexical constituents by means of various cohesion and coherence relations that tie together successive sentences. In contrast to this, we expect monastic texts to be lexically poorer (in order to be better understandable) and also more fragmentary in that successive sentences (of different pairs of citations and commentaries) are less cohesively tied to each other lexically. As a result, monastic sermons are expected to have a less nested coreness and also a higher compactness regarding its maximum core. In contrast to this tendency, we expect that scholastic sermons have a deeper, more nested core structure together with maximum cores that decompose into different, highly cohesive subgraphs reflecting the various standpoints and topics of the sermon. If this is valid, it should be possible to classify monastic and scholastic sermons based on the TRM of Sect. 4: compactness measures, for example, give access to the cohesiveness of the shell of maximum core number and of the remainder graph. Another example is that entropy measures rate less deeply nested core structures less entropic.

As a test corpus we utilize a set of sermons and letters from two medieval authors[17]: *Petrus Abaelardus* and *Bernhard of Clairvaux*. We start with a subset of texts from the Patrologia Latina (PL) [17]. Since several of these texts contain collections of sermons or letters, we perform a segmentation such that any single text in our corpus manifests exactly one letter or sermon of our target authors. As a result, we get a corpus of 590 texts. Since many of these texts are too short to allow for valid lexical networking, we decide to additionally filter out all those texts with less than 100 lemmas (or wordforms, respectively). In the case of lemma-based filtering, a subcorpus of 478 texts remains (see Fig. 2). Lemmatization and tagging has been done with the help of the TLT of Sect. 2. To the best of our knowledge, it provides the most fine-grained, most comprehensive preprocessing of Latin texts available so far. Starting with the output of the TLT, we induce a separate lexical network for each input text. Note that this procedure radically departs from approaches reported in the literature. The reason is that we do *not only* consider networks of wordforms but *also* of lemmas. This is justified by our aim of focusing on the lexical structure

[17] The underlying texts of this corpus have been selected by Silke Schwandt from the Patrologia Latina [17]. They are accessible via the eHumanities Desktop (hudesktop.hucompute.org).



Test corpus	Sermons	Letters	
Bernhard of Clairvaux	338	93	431
Petrus Abaelardus	12	35	47
	350	128	478

Fig. 2 The distribution of the test corpus over two authors and two genres

of texts according to which we need to deal with *signs* and not only with their forms.[18] Further, we include lemmas of every part of speech and therefore do not filter out any kind of stopwords. The reason is that there is no linguistic justification for considering, for example, only content words while disregarding, for example, function words. It has been shown that distributional characteristics of the latter reflect, for example, authorship or style (cf. [62]) so that it does not make sense to leave them out. Further, since we aim at mapping lexical characteristics of the more narrative structure of monastic texts in contrast to the more argumentative structure of scholastic texts, it would be misleading to filter out any lexical content.

Note finally that we exclude the content of any `head`- and `note`-element of the TEI-code of input documents as produced by the TTLab Latin Tagger. The reason is that these XML-elements contain metadata and notes of the editors of the PL that do not belong to the respective source text.

A first look on quantitative characteristics of our test corpus (see left-hand side of Fig. 3) shows that our expectation about the lexical richness of (early) scholastic texts is contradicted. When measuring the *Type-token Relation* (TTR) according to the method of [59], this is obvious. However, since this method of measuring the TTR converges for increasing text positions $x \to N$ to the classical TTR and, thus, is affected (negatively) by text length, we additionally compute the *measure of textual lexical diversity* (MTLD) [60]. Now, we get a slightly different perspective—more into the direction of our expectation (though not in line with it). Figure 4 shows the same ratios but now computed for nominal lemmas in relation to nominal tokens. Obviously, the impression that we get is still the same: compared to Bernhard of Clairvaux, Petrus Abaelardus tends to use a higher number of nominal tokens per noun or a smaller set of nouns. However, we get a different picture—more in the line of our expectation—if we relate the number of nouns to the number of all tokens—see Fig. 5, but only in the case of the MTLD. From this perspective, we observe a richer nominal style in the work of Petrus Abaelardus—either by the relative use

[18] Note that this goal also requires a semantic disambiguation and sense tagging [61]—*beyond PoS tagging*— which is not yet provided by the TLT.

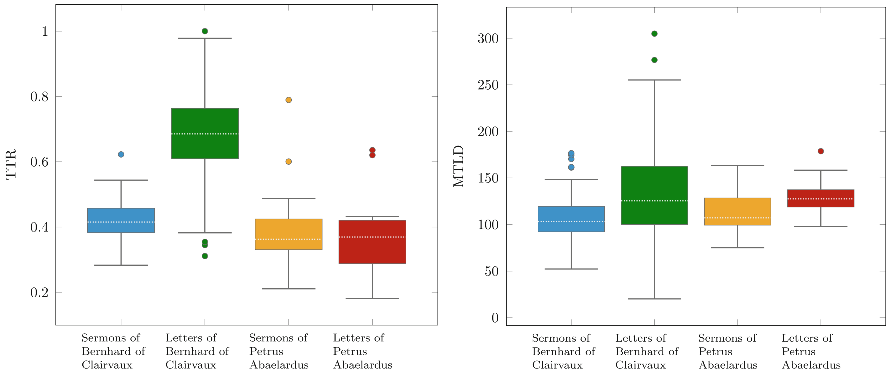


Fig. 3 Boxplots of the distributions of the *Type-token Relation* (TTR) in the target corpus distinguished per author and genre. Each boxplot represents the distribution of TTR-values for the corresponding subcorpus. *Left*: TTR is computed according to [59], that is, for each text position x the $TTR_x = \frac{t_x + T - \frac{xT}{N}}{N}$ is computed and finally averaged over all text positions—obviously, this converges to the classical TTR formula (N is the text length, T (in our case) the number of lemmas and t_x the number of lemmas up to position x). *Right*: the TTR-related index MTLD is computed according to [60] in order to better account for effects of text length. We used a threshold of 0.72 and a minimal sequence length of ten tokens

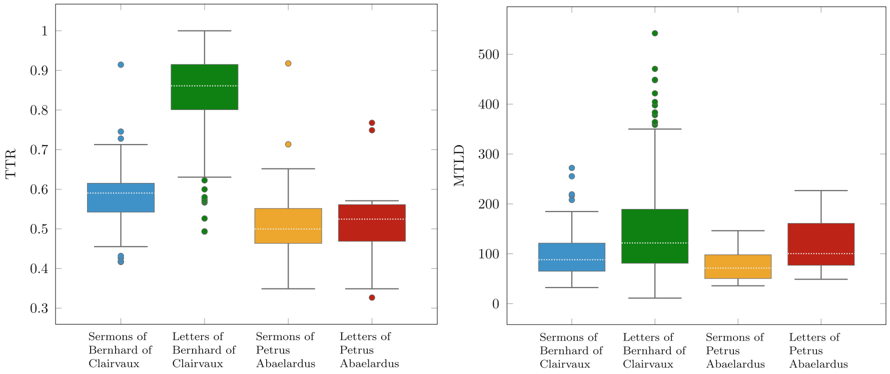


Fig. 4 *Left*: Boxplots for the averaged TTR according to [59] by considering nouns and nominal tokens only. *Right*: Boxplots for the measure of textual lexical diversity (MTLD). The TTR value used for calculating the MTLD index is the ratio of the number of types of part of speech *noun* and the number of tokens of part of speech *noun*. We used a threshold of 0.72 and a minimal sequence length of ten tokens

of a smaller number of tokens per noun or by the use of more nouns in relation to tokens of whatever type. Because of the mostly marginal differences shown in the boxplots, we resist in performing a significance test and pass over to network analysis.

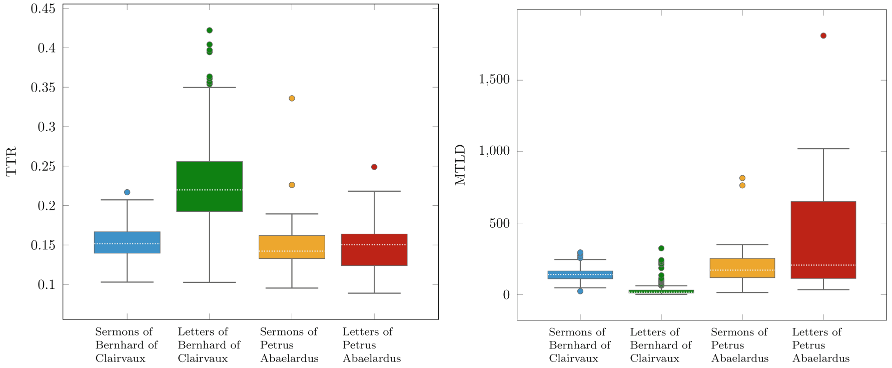


Fig. 5 *Left*: Boxplots for the averaged TTR according to [59] by considering nouns in relation to all tokens. *Right*: Boxplots for the measure of textual lexical diversity (MTLD) by considering nouns in relation to all tokens. We used a threshold of 0.72 and a minimal sequence length of ten tokens

In order to link lemmas in successive text windows, we explore so-called surface-structural co-occurrences [63] according to Miller's hypothesis about the limits of the short term memory [64]. Thus, we consider left- and right-sided text windows of size three around the focal text positions. Further, we allow for sentence-crossing co-occurrences—in other words, we assume that sentences overlap at their boundaries. The reason is that we view the sentence as the basic propositional unit of discourse comprehension [65] without assuming that the reader's memory of the lexical content of a sentence is lost when she is reaching the boundary of the sentence. For reasons of comparison, we also compute the variant according to which such sentence crossing co-occurrence windows are not allowed.

Starting from the input documents' LTNs, we utilize *Quantitative Network Analysis* (QNA) to learn our target classes (authorship, genre) *only by virtue of the structure of their instance LTNs*, while disregarding the labels of their vertices. That is, each document is represented by the 27-dimensional vector of coreness-related features (see Sect. 4) and made input to cluster analysis. This procedure, which combines hierarchical cluster analysis with subsequent partitioning, is informed about the number of target classes and, thus, performs a kind of semi-supervised machine learning on networks. We experiment with various methods of linkage in conjunction with the Euclidean distance to compute pairwise object distances and some alternatives. Classification results are reported in terms of F-scores (see Tables 4, 5, 6, and 7).

Starting with Hypothesis 1 about authorship attribution, we get the results of QNA as shown in Table 4. Obviously, when performing authorship attribution based on wordform-related LTNs, we get better results than in the case of their lemma-based counterparts. Moreover, the corresponding baseline scenarios are clearly outperformed: the one assuming an equipartition of the target classes and the one which is (unrealistically) informed about the cardinality of the target classes (see

Table 4 Ad hypothesis 1: QNA-based authorship attribution (AA) operating on the core structure of LTNs. $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$: with and without crossing sentence boundaries, lemma- or wordform-based

Input	Procedure	*F*-score	Size
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(weighted,euclidean)	0.919	12
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(complete,euclidean)	0.951	17
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.928	9
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.945	12
	Average over non-random approaches	0.936	12.5
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(weighted,euclidean)	0.86	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(weighted,seuclidean)	0.887	27
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(single,euclidean)	0.858	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(single,euclidean)	0.86	27
	Average over *all* non-random approaches	0.901	19.75
	Random baseline (known partition)	0.822	
	Random baseline (equi-partition)	0.612	
	Average over random approaches	0.717	

The last column indicates the number of features (out of 27—see Table 3) that were taken for the respective classification

Table 5 Ad hypothesis 2: QNA-based genre detection (GD) operating on the core structure of LTNs. $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$ as in Table 4

Input	Procedure	*F*-score	Size
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(Ward,euclidean)	0.922	10
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(Ward,euclidean)	0.922	9
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.911	10
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(Ward,euclidean)	0.918	8
	Average over non-random approaches	0.918	9.25
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(Ward,euclidean)	0.839	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(Ward,euclidean)	0.912	27
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(Ward,euclidean)	0.82	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(Ward,euclidean)	0.903	27
	Average over *all* non-random approaches	0.893	18.125
	Random baseline (known partition, RB1)	0.608	
	Random baseline (equi-partition, RB2)	0.545	
	Average over random approaches	0.577	

also the average F-score of the random approaches). From this point of view, we conclude that coreness is a potential source of features of authorship attribution. Note that the spatially least complex model is $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$ which needs only nine features to achieve an F-score of more than 92 %. We get a similar result—though to the prize of lower F-scores—when performing genre detection using the same feature model (see Table 5). However, other than in Table 4, lemma- and wordform-based approaches are now less distinguished. The same holds for sentence- and

Table 6 Ad hypothesis 3: comparing QNA-based authorship attribution (AA) and genre detection (GD) by exploring the core structure of LTNs. $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$ as in Table 4

Procedure	Approach	Procedure	*F*-score
$\mathrm{GD}_{\mathrm{opt}} \rightarrow \mathrm{AA}$	$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(single,euclidean)	0.854
	$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(weighted,euclidean)	0.888
	$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(single,euclidean)	0.854
	$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(single,euclidean)	0.86
Average			0.864
$\mathrm{AA}_{\mathrm{opt}} \rightarrow \mathrm{GD}$	$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(complete,euclidean)	0.858
	$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(Ward,euclidean)	0.906
	$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(complete,euclidean)	0.853
	$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.874
Average			0.873

Table 7 Ad hypothesis 4: QNA-based authorship and genre attribution operating on the core structure of LTNs. $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$, $\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$ as in Table 4

Input	Procedure	*F*-score	Size
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(weighted,euclidean)	0.797	8
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(weighted,euclidean)	0.876	11
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.797	13
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(weighted,euclidean)	0.864	17
	Average over non-random approaches	0.834	12.25
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{scr}}$	QNA(average,euclidean)	0.713	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{scr}}$	QNA(average,euclidean)	0.829	27
$\mathrm{LTN}^{\mathrm{lem}}_{\mathrm{nsc}}$	QNA(average,euclidean)	0.709	27
$\mathrm{LTN}^{\mathrm{wf}}_{\mathrm{nsc}}$	QNA(average,euclidean)	0.79	27
	Average over *all* non-random approaches	0.797	19.625
	Random baseline (known partition)	0.545	
	Random baseline (equi-partition)	0.347	
	Average over random approaches	0.446	

non-sentence-crossing approaches—at least this holds for those approaches which account for the maximum F-scores computed here. In any event, we note that genre detection by means of the coreness of text vocabularies is less complex in terms of the numbers of features being used. Moreover, we observe a difference of more than 30 % when comparing the non-random approaches with their random baselines—whether in terms of the single approaches or their average values. This is a hint at the expressiveness of coreness in terms of the present classification task. From this point of view we state that though we cannot verify Hypothesis (1) and (2), they are currently not falsified.

In Table 6, we consider Hypothesis (3). We use the best performing subset of features of the one task (e.g., authorship attribution) and apply it to the other one (i.e., genre detection). Our expectation is that if authorship attribution based on the coreness of text vocabulary is independent from genre detection based on the same feature space, this transfer should result in a significant loss of F-score. In Table 6, we see that this loss is only about 3–4 %. However, this holds only if, as done in Table 6, we look at the best performing classifications. That is, Table 6 reports upper bounds so that the "real" difference is certainly bigger. In any event, the results so far are not convincing in terms of Hypothesis (3). Therefore, we conclude that though the coreness of a text is expressive in terms of authorship and genre, the characterizations involved are not independent.

Finally, we consider Hypothesis (5), that is, the variant of Hypothesis (4) adapted to our test scenario. The corresponding results are reported in Table 7. It shows that all four classes are remarkably well separated: the average F-score of the baselines is outperformed by the best performer by about 40 %. Interestingly, we observe a clear difference between lemma- and wordform-based networks where the latter perform better with an increase of F-score of about 10 %. Though this does not tell very much about the independence of authorship attribution and genre detection in terms of the formation of lexical cores, it does definitely not falsify Hypothesis 5. From this point of view, one may further look on exploring core-related features for certain tasks of text classification—at least, our standpoint is supported according to which the organization of text vocabularies as modeled by the coreness of corresponding LTNs gives access to a novel text representation model. In any event, starting an explorative search into the direction of network-based models as exemplified here requires a very fine-grained, very sound and exact preprocessing—whether on the level of wordforms only (where one needs to consider variants, misspellings etc.) or even on the level of lemmatization. In this sense, the present chapter necessarily ranges—as mentioned in the introduction—from resource formation and preprocessing historical documents to their network-based representation and classification. We are convinced that any future effort in text classification will not so much depend on better machine learners, but rather on better preprocessors and sophisticated representation models based thereon.

A final note: each of the hypotheses (1–5) assumes that the divergence of the lexical organization of the texts under consideration correlates with divergent core structures of the LTNs derived from these texts. While the former divergence is seen to be empirical (since it concerns texts as empirical relatives), the latter divergence concerns numerical relatives and, thus, units of measurements that are here defined in terms of the TRM of Sect. 4. The simple this note, the significant its implications. The reason is that while a failure of experiments along the hypotheses (1–5) would clearly falsify the significance of our TRM, such a failure would however not falsify statements about the significance of the lexical text organization with respect to authorship attribution or genre detection. This also means that a successful experimentation cannot be interpreted in favor of the significance of our TRM, but only in favor of its expressiveness or efficiency (more narrowly: in favor of not being falsified). That is, high F-scores do not mean that the explored differences map the ones assumed by our hypotheses. To show this, we would need to show that our measurements are valid—to do this much more than machine learning is needed (cf. [66]). In a nutshell: our experiments do not touch the significance of the underlying linguistic hypotheses but show the expressiveness of our numerical TRM. From the point of view of computational humanities, this approach is in need of improvement —*we still face a long way to a* valid *numerical text representation model*.

6 Conclusion

We introduced the *TTLab Latin Tagger* (TLT) and the *Frankfurt Latin Lexicon* (FLL) as means to preprocess Latin texts automatically. The TLT is a hybrid tagger that combines a statistical model with morph-syntactical rules. It operates on the FLL as a lexical resource that is continually extended by exploring web-based resources of Latin. Both of these resources were used to exemplify the build-up of a novel, minimal text representation model that analyzes a rather unexplored area of lexical network formation, that is, the coreness of text vocabularies. In a tripartite classification experiment we demonstrated the expressiveness of our text representation model. However, the wide range of resource creation and explorative text analysis covered in this chapter also showed that still a lot needs to be done to establish the sort of methods developed here in the humanities. Future work will deal with testing our text representation model systematically in the area of authorship attribution and genre detection. Special emphasis will be given on testing the validity of this model and its integration with other approaches of numerical and symbolic text analysis.

Acknowledgements Financial support by the BMBF-project *Computational Historical Semantics* (www.comphistsem.org) as part of the research center on *Digital Humanities* is gratefully acknowledged. We also thank both anonymous reviewers for their valuable hints, Barbara Job and Silke Schwandt for their fruitful hints on distinguishing monastic and scholastic texts and, finally, Andy Lücking for assisting in producing TikZ graphics.

References

1. Heyer G (2014) Digital and computational humanities. www.dagstuhl.de/mat/Files/14/14301/14301.HeyerGerhard.ExtAbstract.pdf
2. Hearst MA (1999) Untangling text data mining. In: Proceedings of ACL'99: the 37th annual meeting of the association for computational linguistics, University of Maryland
3. Mehler A (2004) Textmining. In: Lobin H, Lemnitzer L, (eds) Texttechnologie. Perspektiven und Anwendungen, Stauffenburg, Tübingen, pp 329–352
4. de Saussure F (1916) Cours de linguistique générale. Payot, Lausanne/Paris
5. Peirce CS (1993) Semiotische Schriften 1906–1913, vol 3. Suhrkamp, Frankfurt am
6. Crane G, Wulfman C (2003) Towards a cultural heritage digital library. In: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries (JCDL '03), Washington. IEEE Computer Society, pp 75–86

7. Bamman D, Passarotti M, Busa R, Crane G (2008) The annotation guidelines of the latin dependency treebank and index thomisticus treebank. In: Proceedings of LREC 2008, Marrakech, Morocco, ELRA
8. Bamman D, Crane, G (2009) Structured knowledge for low-resource languages: The Latin and Ancient Greek dependency treebanks. In: Proceeding of the text mining services 2009, Leipzig. Springer, New York
9. Passarotti M (2010) Leaving behind the less-resourced status. The case of Latin through the experience of the Index Thomisticus Treebank. In: Proceedings of the 7th SaLTMiL workshop on the creation and use of basic lexical resources for less-resourced languages (LREC 2010), La Valletta, Malta, ELDA
10. Gleim R, Hoenen A, Diewald N, Mehler A, Ernst A (2011) Modeling, building and maintaining lexica for corpus linguistic studies by example of Late Latin. In: Corpus Linguistics 2011, Birmingham, 20–22 July 2011
11. Büchler M, Heyer G, Gründer S (2008) eAQUA–bringing modern text mining approaches to two thousand years old ancient texts. In: Proceedings of e-Humanities–An emerging discipline, workshop at the 4th IEEE international conference on e-Science
12. Jussen B, Mehler A, Ernst A (2007) A corpus management system for historical semantics. Sprache und Datenverarbeitung. Int J Lang Data Proc 31(1–2):81–89
13. Büchler M, Geßner A, Heyer G, Eckart T (2010) Detection of citations and text reuse on ancient Greek texts and its applications in the classical studies: eAQUA project. In: Proceedings of digital humanities 2010, London
14. Mehler A, Schwandt S, Gleim R, Ernst A (2012) Inducing linguistic networks from historical corpora: Towards a new method in historical semantics. In: Durrell M et al (eds) Proceedings of the Conference on new methods in historical corpora, April 29–30, 2011, Manchester. Corpus linguistics and Interdisciplinary perspectives on language (CLIP). Narr, Tübingen, pp 257–274
15. Crane, G (1996) Building a digital library: the perseus project as a case study in the humanities. In: Proceedings of the first ACM international conference on Digital libraries (DL '96), New York. ACM, USA, pp 3–10+++
16. Smith DA, Rydberg-Co JA, Crane GR (2000) The Perseus Project: A digital library for the humanities. Lit Linguistic Comput 15(1):15–25
17. Jordan MD (ed) (1995) Patrologia latina database. Chadwyck-Healey, Cambridge
18. Amancio DR, Antiqueira L, Pardo TAS, Costa LdF, Oliveira ON, Nunes MDGV (2008) Complex networks analysis of manual and machine translations. Int J Mod Phys C 19(4):583–598
19. Amancio DR, Jr, ONO, da Fontoura Costa L (2012) Identification of literary movements using complex networks to represent texts. New J Phys 14:043029
20. Liu J, Wang J, Wang C (2008) A text network representation model. In: FSKD '08: Proceedings of the 2008 fifth international conference on fuzzy systems and knowledge discovery, Washington. IEEE computer society, pp 150–154
21. Mehler A (2008) Large text networks as an object of corpus linguistic studies. In: Lüdeling A, Kytö M (eds) Corpus Linguistics. An international handbook of the science of language and society. De Gruyter, Berlin, pp 328–382
22. Koster CHA (2005) Constructing a parser for Latin. In: Gelbukh AF (ed) Proceedings of the 6th international conference on computational linguistics and intelligent text processing (CICLing 2005). LNCS, vol 3406. Springer, New York, pp 48–59
23. Passarotti M, Dell'Orletta F (2010) Improvements in parsing the index thomisticus treebank. Revision, combination and a feature model for medieval Latin. In: Proceedings of LREC 2010, Malta, ELDA
24. Voutilainen A (1995) A syntax-based part-of-speech analyzser. In: Proceedings of the 7th conference of the European chapter of the association for computational linguistics (EACL), Belfield, Ireland pp 157–164
25. Jurafsky D, Martin JH (2000) Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall, Upper Saddle River

26. Manning CD, Schütze H (1999) Foundations of statistical natural language processing. MIT Press, Cambridge
27. Ratnaparkhi A (1996) A maximum entropy model for part-of-speech tagging. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). Philadelphia, Pennsylvania
28. Tsochantaridis I, Joachims T, Hofmann T, Altun Y (2005) Large margin methods for structured and interdependent output variables. J Mach Learn Res **6**:1453–1484
29. Nguyen N, Guo Y (2007) Comparisons of sequence labeling algorithms and extensions. In: Proceedings of the 24th International conference on machine learning (ICML). ACM, New York
30. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning. St. Petersburg/Russia
31. Constant M, Sigogne A (2011) MWU-aware part-of-speech tagging with a CRF model and lexical resources. In: MWE '11 Proceedings of the workshop on multiword expressions: from parsing and generation to the real world. Stroudsburg, pp 49–56
32. Simionescu R (2011) Hybrid pos tagger. In: Proceedings of the workshop on language resources and tools with industrial applications, Cluj-Napoca
33. Mehler A, Gleim R, Waltinger U, Diewald N (2010) Time series of linguistic networks by example of the Patrologia Latina. In: Fähnrich KP, Franczyk B, (eds) Proceedings of INFORMATIK 2010: service science, September 27—October 01, 2010, Leipzig. Volume 2 of Lecture Notes in Informatics, GI, pp 609–616+++
34. Passarotti M (2000) Development and perspectives of the Latin morphological analyser LEMLAT (1). Linguistica Computazionale 3:397–414
35. Schmid H (1994) Probabilistic part-of-speech tagging using decision trees. In: Jones D, Somers H (eds) New methods in language processing studies in computational linguistics. UCL Press, London
36. Springmann U, Najock D, Morgenroth H, Schmid H, Gotscharek A, Fink, F (2014) OCR of historical printings of Latin texts: problems, prospects, progress. In: Antonacopoulos A, Schulz KU (eds) Digital access to textual cultural heritage 2014 (DATeCH 2014), Madrid. ACM, May 19–20, pp 71–75
37. Okazaki N (2007) CRFsuite: a fast implementation of conditional random fields (CRFs). http://www.chokkan.org/software/crfsuite/manual.html
38. Zipf GK (1972) Human behavior and the principle of least effort. An introduction to human ecology. Hafner Publishing, New York
39. Panhuis DG (2009) Latin grammar. University of Michigan Press, Ann Arbor
40. Liiv H, Tuldava J (1993) On classifying texts with the help of cluster analysis. In: Hřebíček L, Altmann G (eds) Quantitative text analysis. Wissenschaftlicher Verlag, Trier, pp 253–262
41. Schuhmacher M, Ponzetto SP (2014) Knowledge-based graph document modeling. In: Proceedings of the 7th ACM international conference on web search and data mining (WSDM '14), New York. ACM, pp 543–552
42. Seidman SB (1983) Network structure and minimum degree. Soc Networks 5:269–287
43. Batagelj V, Zavervsnik M (2003) An $O(m)$ algorithm for cores decomposition of networks. http://vlado.fmf.uni-lj.si/vlado/vladounp.html. arXiv:cs/0310049
44. Ashraf M, Sinha S (2012) Core-periphery organization of graphemes in written sequences: decreasing positional rigidity with increasing core order. In: Gelbukh A (ed) Computational linguistics and intelligent text processing. Lecture notes in computer science, vol 7181. Springer, New York, pp 142–153
45. Fortunato S (1983) Community detection in graphs. Phys Rep 486(3–5):75–174
46. Giatsidis C, Thilikos DM, Vazirgiannis M (2011) Evaluating cooperation in communities with the k-core structure. In: Proceedings of the 2011 international conference on advances in social networks analysis and mining (ASONAM '11), Washington. IEEE Computer Society, pp 87–93

47. Alvarez-Hamelin JI, Dall'Asta L, Barrat A, Vespignani A (2008) k-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases. Net Heterogeneous Media **3**(2):371–393
48. Halliday MAK, Hasan R (1989) Language, context, and text: aspects of language in a socialsemiotic perspective. Oxford University Press, Oxford
49. Dehmer M (2008) Information processing in complex networks: Graph entropy and information functionals. Appl Math Comput 201:82–94
50. Dehmer M, Mowshowitz A (2011) A history of graph entropy measures. Inform Sci 181(1):57–78
51. Mehler A (2011) A quantitative graph model of social ontologies by example of Wikipedia. In: Dehmer M, Emmert-Streib F, Mehler A (eds) Towards an information theory of complex networks: statistical methods and applications. Birkhäuser, Boston, pp 259–319
52. Cover TM, Thomas JA (2006) Elements of information theory. Wiley-Interscience, Hoboken
53. Botafogo RA, Rivlin E, Shneiderman B (1992) Structural analysis of hypertexts: identifying hierarchies and useful metrics. ACM Trans Infor Syst 10(2):142–180
54. Mehler A (2008) Structural similarities of complex networks: A computational model by example of wiki graphs. Appl Artif Intell 22(7,8):619–683
55. Mehler A, Pustylnikov O, Diewald N (2011) Geography of social ontologies: testing a variant of the Sapir-Whorf Hypothesis in the context of Wikipedia. Comput Speech Lang 25(3):716–740
56. Pieper U (1975) Differenzierung von Texten nach numerischen Kriterien. Folia Linguistica VII:61–113
57. Frank-Job B (1994) Die textgestalt als zeichen. Lateinische handschriftentradition und die verschriftlichung der romanischen sprachen, ScriptOralia, vol 67. Narr, Tübingen
58. Frank-Job B (2003) Diskurstraditionen im Verschriftlichungsprozeß der romanischen Sprachen. In: Aschenberg H, Wilhelm R (eds) Romanische sprachgeschichte und diskurstraditionen. Narr, Tübingen, pp 19–35
59. Köhler R, Galle M (1993) Dynamic aspects of text characteristics. In: Hřebíček L, Altmann G (eds) Quantitative text analysis. Wissenschaftlicher Verlag, Trier, pp 46–53
60. McCarthy PM, Jarvis S (2010) Mtld, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. Behav Res Methods 42(2):381–392
61. Schütze H (1998) Automatic word sense discrimination. Computat Linguistics 24(1):97–123
62. Stamatatos E (2011) Plagiarism detection based on structural information. In: Proceedings of the 20th ACM international conference on information and knowledge management (CIKM '11), New York. ACM, pp 1221–1230
63. Evert S (2008) Corpora and collocations. In: Lüdeling A, Kytö M (eds) Corpus linguistics. An international handbook of the science of language and society. Mouton de Gruyter, Berlin, pp 1212–1248
64. Miller GA (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychol Rev 63:81–97
65. van Dijk TA, Kintsch W (1983) Strategies of Discourse Comprehension. Academic Press, New York
66. Rieger B (1998) Warum fuzzy Linguistik? Überlegungen und Ansätze zu einer computerlinguistischen Neuorientierung. In: Krallmann D, Schmitz HW (eds) Perspektiven einer Kommunikationswissenschaft. Internationales gerold ungeheuer symposium, Essen 1995. Nodus, Münster pp 153–183

Part II
Text Mining Applications

A Structuralist Approach for Personal Knowledge Exploration Systems on Mobile Devices

Stefan Bordag, Christian Hänig, and Christian Beutenmüller

Abstract We describe the reasons and choices we made when designing an architecture for a multilingual Natural Language Processing (NLP) system for mobile devices. The most tangible limitations and problems are limited processing power of mobile devices, strong influence of idiolect (or generally personal language usage differentiation between individual users in their personal communication), effort required to port the NLP system to multiple languages, and finally the additional processing layers required when dealing with real-world data as opposed to *controlled* academic set-ups. Our solution is based on a strict differentiation between server-side preprocessing and client-side processing, as well as maximized usage of unsupervised techniques to avoid the problems posed by personal language usage variations. Hence it represents an adequate combination of solutions to provide robust NLP despite all these limitations.

1 Introduction

Natural Language Processing is a field that has evolved dramatically over the past fifty years. Some of the early core areas of study were syntax, morphology, and semantics of languages. The main goal is to provide methods to let computers do at least some of the work that us humans do with information embedded in unstructured data.[1] By now there is a wide range of use cases for which at least practically acceptable solutions exist, such as Machine Translation, Information

[1] Assuming the dichotomy between structured and unstructured data to be loosely defined as "information or pieces of data stored in explicit relations with each other (cf. relational data bases) to be structured data and information stored without explicit relations within unanalyzed texts to be unstructured". Another possible definition is that "any data base is structured if it is possible to use a simple and precise query system which guarantees to retrieve a particular piece of information if it exists". No current system is able to achieve that fully with textual information.

S. Bordag (✉) • C. Hänig • C. Beutenmüller
ExB Research & Development GmbH, Seeburgstr. 100, 04103 Leipzig, Germany
e-mail: bordag@exb.de; haenig@exb.de; beutenmueller@exb.de

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_6

Retrieval, Spell Checking, and Speech processing, amongst others. Most of these approaches are based on explicitly distinguishing individual languages. This makes it possible to build dedicated modules for each language such as for English versus a module for Spanish or French, etc. It is often assumed implicitly that a module built for English will be more or less uniformly valid for the language and all possible texts written in it. Yet it is obvious that there are distinguishable subdomains such as medicine versus technical documents within languages. Accordingly, much of the effort in NLP goes into solutions to the domain dependency problem to either reduce that dependency or find easier ways to port a solution from one domain to another.

In its entirety, any approach based on building language modules can be described as top-down, because at the core it will be about some particular language or language family. It splits the problem of building an NLP system top-down into solutions for particular languages of a language family and finally for domains inside a language. But what would a bottom-up approach be in this case? The most reasonably extreme form of that would be to treat the production and perception of language by a single person as distinguishable from that of any other person, which is very close to the one sense per discourse idea, but not as extreme.

A single person typically speaks one or more languages, is a specialist in one or more subdomains of each language she/he speaks, and also uses words, expressions, and even syntactic constructs in a highly idiosyncratic way. So much so, in fact, that it is even possible to detect original authorship of texts purely based on word frequency differences with a certain degree of confidence [33], distinguish individual persons based on their language production patterns even within the same location, profession, and topics [5], or conduct linguistic forensics pertaining to individual language usage [31]. The idiolect, the difference in language use amongst individual language users, may not seem very large. However, typical inter-annotator agreement figures show that these differences are far from negligible. For example, [21] show correlation scores ranging from 0.71 to 0.84 (Kappa) between human annotators while [11] show that even a task such as the detection of chemical Named Entities (which appears relatively easy) achieves 93 % F-score agreement between human annotators.

The relatively recent rapid adoption of new communication channels such as short messaging services (SMS) has brought about entirely new interaction limitations, which in turn spawned entirely new language domains, which additionally interact with existing domains. For example, lawyers and researchers are likely to exhibit different kind of language usage when they use SMS on smartphones or email on PCs. This leads to a further dispersion of personal language usage into larger numbers of different subdomains, cross-domains, and complex domain interaction phenomena. So much so that trying to build domain specialized NLP solutions for each of these possible interactions becomes more and more intractable. But is there any other way? Is it even possible to think of a solution that solves or circumvents these problems? In order to tackle these and other problems related to building NLP solutions for mobile devices, we formulate the following hypotheses and show that incorporating experimental results and observations arising from these hypotheses improves user-centric performance levels of our NLP solution significantly.

Language usage variation hypothesis We assume that language usage between different users differs so strongly that any additional mechanism of adapting to user-specific language usage will easily outperform any pre-built NLP systems that do not adapt.

Limited language usage variation hypothesis We assume that the language usage variation is stronger (weaker) in some areas of language. Hence it makes sense to preprocess some parts of the language models on a large corpus on a server and only compute the more variable parts directly on-device.

Principle of minimal language dependence We assume that a solution to an NLP problem which requires substantial professional input[2] will be inferior to a solution which provides slightly inferior results in an evaluation but requires significantly less or no professional input on real-world data in order to adapt it to a new language or domain.

Additional processing layers Real-world data contains many more structural levels compared to theoretical models of language usage which often focus on morphemes, words, sentences, paragraphs, and larger text units. We assume that any system that provides even basic support for dealing for additional levels (e.g. lists, tables, log files) will provide noticeably improved quality levels on real-world data.

Our solution is a combination of traditional NLP methods with structuralist methods as extensions of statistical algorithms and in some occasions simple word lists. This approach extends many of the more basic approaches described in [24]. We particularly want to show that such solutions can be built in a sufficiently efficient way so that they can even be employed on the weakest possible computing devices, that is, mobile devices. One reason to do this is that in the past, excessive computing power requirements of structuralist approaches have been used as a counterargument—as we think—erroneously.

1.1 The Structuralist Approach and Personal Data

In structuralism signs (e.g. words) get their meaning from their relationships and contrasts with other signs [12]. Related to this, an unsupervised approach is loosely defined as a solution to an NLP problem that relies on finding structure within the input text on its own, rather than assuming structure through pre-annotated training data or explicit rules and assignments [23]. Such algorithms are based on measuring the observable patterns of usage of, for example, words with other words. An unsupervised part-of-speech tagger, for example, clusters words according to

[2] Input by people educated in a relevant field of linguistics.

their syntactic usage as measured by occurrence with other nearby words, instead of assuming rigid categories such as nouns or verbs. This typically requires a large amount of raw data in order to produce good results, and there is a logarithmic dependency between data size and quality, as observed in Chap. 3 in [6]. Hence, the first step is to assess what kind of input data can be used as input and whether it is sufficient. Personal data can be found in and between the realms of structured and unstructured data. Examples for structured personal data include:

- address books across various device/application combinations such as email client, mobile phone, or notebook
- calendar applications
- equipment or book rental lists in custom data base designs (often, this is given just by a single Excel table)
- music and video files
- bookmarks of favorite websites and histories of visited websites

Beyond that, a large bulk of real information is stored in unstructured data (or semi-structured data):

- documents the user wrote or received (note that usually information about the creator of a particular document does not exist explicitly)
- emails the user wrote or received
- short messages the user wrote or received on a multitude of devices (mobile phone, notebook, PC, various websites such as Facebook).

The level of detail available in personal data from both data sources is already staggering. With enough personal data, a highly accurate map of a given user's daily life can be achieved. No current systems are able to come even close to enabling the user to work with all her/his own knowledge in a seamless, fully integrated way. The main obstacles that need to be overcome include, among other things, the following ones:

- The individual structured data bases are scattered among a great variety of completely different and incompatible non-standardized custom data bases.
- There exists no software that is able to reliably extract clean bits of knowledge from all the unstructured data that the user has. There are only large-scale projects such as NELL[3] [7], or Google knowledge graph[4] or IBM Watson[5] [15] which cannot be applied on the personal data and needs of a single user. Other solutions such as Apple's Siri, Google's Google Now are typically server based and do not possess capabilities yet to analyze the unstructured data of the user.

But what kinds of real-world use cases would such a software enable. Is it possible to solve some of these use cases with existing technology?

[3] rtw.ml.cmu.edu/rtw, retrieved on 24.03.2014.

[4] www.google.com/insidesearch/features/search/knowledge.html, retrieved on 24.03.2014.

[5] www-05.ibm.com/de/watson, retrieved on 24.03.2014.

1.2 Mobile Devices

Natural language processing on mobile devices poses a particularly hard combination of challenges, such as extremely limited computing power or very limited interaction with the device due to small screens and tiny virtual keyboards. Excluding spoken language from the scope of this contribution still leaves us with an immense variety of communication types such as SMS, emails, and web pages, amongst others. The amount of data available for each of these types will be small compared to what computational linguists are used to work with, but not too small to consider statistical approaches. It is reasonable to expect a single user to handle on the order of a thousand SMS, emails or visited websites per year. Additionally, each of these sources of different text type poses problems which, at first, seem to be peripheral from the point of view of traditional NLP.

For example, SMS messages feature and encourage strongly abbreviated language patterns. Emails contain huge amounts of seemingly non-linguistic components, such as signature text parts, quoted email parts, tables, ASCII art, emoticons etc. Websites typically devote very small areas for content proper and use most of the screen estate for ancillary data such as menus, advertisements, and (un)related content. When viewing emails or websites, the human eye quickly catches all relevant parts and discards irrelevant parts even quicker. However, simulating even this task as an algorithm proves to be surprisingly difficult [28] and is only solved using indirect tricks, such as comparing different websites from the same provider with each other and tracking the similar parts.

Finally, mobile devices are sold and used on a global scale now, so any NLP solution will also have to be useful for a wide variety of languages, from English through Finnish to Korean and Japanese. This means that even very basic traditional components such as tokenizers, sentence boundary detectors and part-of-speech taggers can become obstacles in some cases. It also means that the effort of adapting a NLP system to a new language must be as cost-efficient as possible in order to make it feasible to cover several dozen languages.

2 Our Solution

Considering all the problems and limitations described above, we have built and tested in a commercial product[6] a solution which offers a robust NLP capability for a variety of use cases, including information extraction, text similarity measurement, and various classification tasks. This solution is a hybrid system including rule-based parts as well as purely statistical parts. The system is split into two parts, one for pre-processing on the side of the server and a second for

[6] The product is an email client with enhanced NLP capabilities; see http://mailbe.at.

on-device computation. We employ rule-based systems for basic NLP tasks such as tokenization or sentence boundary detection as well as for tasks of information extraction system as, for example, time extraction. We further employ rule-based systems for detecting macro-structures such as detecting quoted text, bulleted lists and tables. We use self-learning classifiers for signature detection in emails, website scraping, email classification and other on-device learning components. We employ server-side bootstrapping algorithms for our generalized named entity recognition framework and client-side trained and compressed models. Finally, we use entirely unsupervised solutions for text similarity estimations, text summarization, inducing semantic networks and other components which are strongly influenced by personal language variation. Some of the learning components, in particular all classifiers, run exclusively on the mobile device. Some components, such as the pre-computation of a part-of-speech tagger model (clustering and training) run on the server and produce compressed language models for client-side deployment.

2.1 Pledge for Additional "Language" Layers

When developing NLP solutions, one of the most critical steps is to introduce an evaluation framework which measures success and error rates. Our main finding of the study reported here was that the primary source of errors is not partial intelligence in the NLP solution (even if it is just a baseline solution) but that most errors are introduced by applying the solution in the wrong way or on the wrong data. For example, even a very simple sentence boundary detection algorithm usually produces very few mistakes (typically around abbreviations). This is demonstrated, for example, in [27] which show that the difference between the best and worst system is only 1,5 % (error rate) on the Brown corpus. However, when applied to the raw text of an email, it is not uncommon to see that error rates jump up to nearly 100 %. This is because an email may contain a single sentence and a long log printout with lots of dots scattered around (none of which signify sentence boundaries), and no dots at the end of each log entry (which would constitute sentence boundaries). In an academic setting, it is easy to discard such examples as *invalid* or *unfitting data*. When it comes to real-world applications, it is impossible to use such an argument because real-world applications need to detect and make automatic decisions whether to use or ignore data of any kind. Taking into account all of these issues and challenges, the architecture of our NLP system for mobile devices can be seen in Fig. 1. The basic idea underlying this architecture is to design in a way that the traditional NLP modules (such as time and named entity recognition) are applied only after extensive pre-processing steps. These steps detect text types and blocks, and apply the analytic algorithms only on data where and when it makes sense. This approach reduces error rates to levels comparable to the ones reported in the literature and also keeps the error levels relatively constant.

Another aspect of our architecture is that it divides between server-side processing of language models and client-side processing for auxiliary modules. There

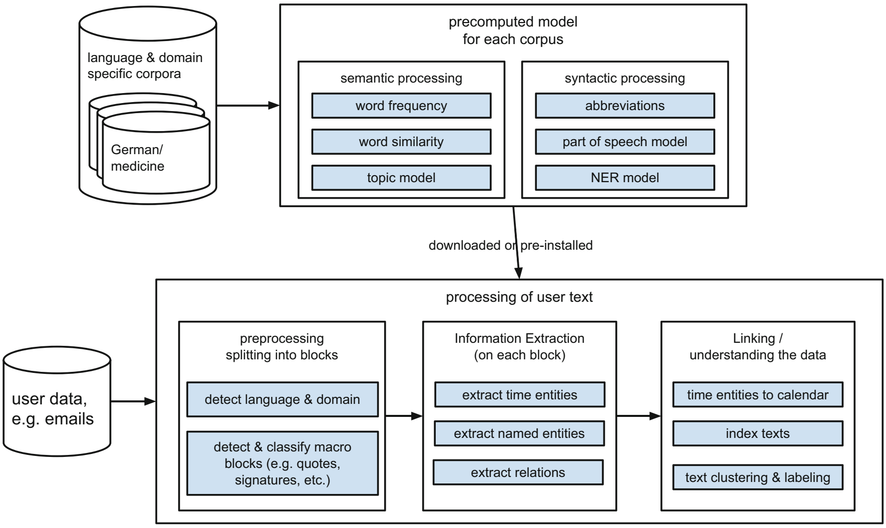

language & domain specific corpora
German/ medicine
precomputed model for each corpus
semantic processing
word frequency
word similarity
topic model
syntactic processing
abbreviations
part of speech model
NER model
downloaded or pre-installed
processing of user text
user data, e.g. emails
preprocessing splitting into blocks
detect language & domain
detect & classify macro blocks (e.g. quotes, signatures, etc.)
Information Extraction (on each block)
extract time entities
extract named entities
extract relations
Linking / understanding the data
time entities to calendar
index texts
text clustering & labeling

Fig. 1 An overview of the architecture of our NLP solution for mobile devices

are some algorithms that require a huge amount of input data in order to deliver acceptable results. Examples of such algorithms include part-of-speech tagging [29] and the differential analysis used for key word extraction [39]. On the other hand, there are algorithms that require relatively little data. However, they expect data which fits the problem very well. A good example of that is signature detection (based on observing similarities between emails from the same sender) or quote detection (where a helpful feature is whether some part of text has been seen previously) (cf. [8]).

In the following two sections we show how different parts of our architecture can be combined to achieve optimal performance levels. The next section describes the influence of the various parts and limitations of a mobile device on text similarity measurement followed by a detailed section on information extraction.

3 Text Similarity Measurement

Measuring semantic text similarities is a topic of NLP with a large variety of use cases. Finding semantically related content is an extension of the core use case of search in information retrieval. We can simply reformulate the typical information retrieval use case using arbitrary documents as a query, for example in the vector space model [34]. However, apart from information retrieval, many other applications that deal with textual data may benefit from text similarity algorithms and associated index structures. For example, in an email application, it can be used to show emails related to the one the user is currently reading or writing. It can also be used as a basis for text clustering, automatic filtering, automatic classification of emails into folders, and even recipient prediction or verification.

On the other hand, it is a computationally intensive problem because the worst case time complexity of exhaustive comparisons amongst texts is quadratic. One possible technique to reduce the number of comparisons is a pruned inverse index that associates a term with a limited set of documents. But any kind of pruning has the implied risk of missing relevant information by simply skipping it. This might lead to a reduction of the number of actually found similar items. A good pruning, however, could even increase the quality of the found items in terms of semantic similarity by inspecting only *relevant* terms. The central problem hereby is the computational representation of *relevancy*. Often, this is done by employing measures of statistical significance. The differential analysis formula presented in [40] (which is in turn based on the log-likelihood significance test [14]) is used in the following experiments to build a proprietary document index based on the frequency of a word in a document in comparison to its frequency in a pre-processed corpus as well as the size of the document.

Another aspect of measuring semantic similarities is that, even without any pruning, there is a risk of missing documents of the same topic(s) that use different words of close, similar or related meanings. For example, the two sentences "That tank fired a round" and "Armor fire detected" would be missed by an index that is based only on words. Additional knowledge (as provided, for example by a thesaurus) is needed to allow the text similarity system for matching words of related meaning. This knowledge can be provided by a lexical database like WordNet [32, 38]. The availability or size of such manually compiled lexical databases is rather limited for some languages. In some cases even their applicability beyond academic projects is limited. However, as discussed under the distributional hypothesis earlier [6], a corpus in a language and a domain is sufficient to automatically compute a knowledge base of distributional semantics that may serve the same purpose.

As a result, text similarity systems consist of a variety of components. Figure 1 shows the main building blocks of our NLP solution. Pre-processing starts with parsing of raw documents like textual emails, HTML, XML, or Office documents, and then continues with language detection, tokenization and text cleaning. The text cleaning part is highly dependent of the source and text type of the data. For generic web pages, for example, we employ a self-learning module that learns to skip advertisements and structural data like menus. In the email use case, processing and correct handling of quotations and signatures is central to the quality of the text similarity algorithm and other parts of the NLP processing chain. We have discovered that when our system does not recognize signatures and hence cannot exclude them from any processing, the performance of the text similarity system degrades severely for users who share many messages on different topics (e.g. co-workers or friends). Although in theory there is a *convention* regarding how to separate signatures from the main body (`-  \n`), many if not most users choose not to follow the convention. Signature detection has thus been defined as a classification task [8] with the conventional pattern as just one out of many possible features or as one part of a generalized email structure classifier as outlined above [30].

Typically, a complex information retrieval system also includes a lemmatization module which maps, for example, “fired” and “fire” on the same lemma. It may also contain the aforementioned thesaurus module which would allow for matching *tank* with *armor*. We assume that usage differences of personal language affect some areas of language more than others. For example, differences in pronunciation or choice of words tend to be stronger than differences in grammar or morphology. As a consequence, we state that some modules might profit only very little or not at all from a perfect fit to the data of the user. Other modules, however, would profit much more. In order to show the potential influence of personalizing NLP systems through self-learning techniques we now discuss several experiments.

3.1 Evaluation Method

Traditionally, evaluation of *Information Retrieval* (IR) systems has been centered around the so-called Cranfield Experiments using gold standards (e.g. TREC[7]). These gold standards offer various beneficial properties for the research community like comparability and repeatability. They are limited in scope and availability, however. Apart from that personalized information retrieval systems are hard to evaluate using standardized gold sets, since gold sets cannot easily capture the applied personalization. This might explain why the evaluation of personalized IR systems has focused on user studies [13]. While these user studies can clearly capture the personalization, they are limited in terms of repeatability and comparability. Gold standards require higher up-front costs but may be reused any number of times. User studies on the other hand lead to continuous costs for each experiment or improvement cycle. Simulated user studies or a living lab as described in [3, 26] might be a solution to combine the desired repeatability of experiments with the ability to capture personalization.

In order to quickly measure the impact of certain improvements, a specialized gold standard may provide a good estimate of the impact in the final product. Due to licensing restrictions and the previously mentioned scope and limitations of availability, we decided to create our own gold standard data sets. We distinguish gold standards by their language, source and content type. Each gold data set consists of a document collection with a list T of topics from a defined text source. Each topic contains a number of associated documents. We make a conscious simplification by assuming that a document is related to exactly one topic. Additionally we evaluated our system using the data from the SemEval 2012 and 2013 tasks on *Semantic Text Similarity* [1, 2].

[7] trec.nist.gov.

3.2 Experiments on News and Email Text Collections

For evaluation we add the entire collection to the index and then iterate over all documents d in the current document collection D. For each d we query a ranked top n list of similar documents d'. Each d' is then compared to the content of T (a list of topics, see above) of the original document d. From these counts of matching and non-matching documents we calculate the precision, recall and F_1-score (i.e., the harmonic mean of precision and recall). We use two different English gold standards which encompass:

- A collection of 26 topics in news texts where exactly 20 texts are given per topic
- A collection of 18 topics in emails from a single user with an average of 12 emails per topic.

The following experiments show the impact of domain dependence and advanced domain dependent filtering on our semantic text similarity index. To show the influence of domain dependence we compare pre-trained models with models trained directly on the evaluation data. Additionally we show the impact of signature detection to the email use case.

For the following experiments we used word frequencies and distributional semantic clusters pre-calculated on a 1 million lines newspaper corpus comparable to those available from the Wortschatz project [18]. In addition, we created a small corpus from the email gold data set with about 6,000 lines of text and computed distributional semantic clusters based on that as well.

E_0 is the baseline experiment evaluated on the news gold standards using our pruned index, distributional semantics, and word frequencies calculated from our newspaper corpus. E_1 evaluates the email gold standard using the same parameters as E_0. For E_2 we employed adapted models by using the distributional semantic clusters computed on the email corpus. For E_3, E_4 and E_5 we additionally executed a signature detection module to filter the emails. E_3 hereby uses the same default semantic clusters as employed in E_1 and a generic pre-trained version of our custom signature detection classifier. E_4 refines the signature detection classifier by training it directly on the emails. Finally E_5 is a combination of both the adapted semantic clusters as in E_2 and the trained signature detection classifier. For each of the experiments we show the aggregated precision, recall and F_1-score as computed by querying the index for the top 10 similar items to each given text. The results are summarized in Table 1.

With an F_1-scores of 72.43 % Experiment E_0 clearly outperforms Experiment E_1 (56.73 %). There are two main reasons for this. First of all, the real-world email data of E_1 is noisier than the manually cleaned news text. Among others, the email dataset contains large signatures, complicated quotes, bad spelling, ASCII art and even partial email headers in quotes, tables or stack traces. Secondly, we assume that the trained newspaper model is a better match for the news gold standard than for the emails. Further evidence for this assumption can be observed by using the computed distributional semantics of the email gold set as described above. By comparing the

Table 1 Experiments on source-dependent text cleaning and domain-dependent distributional semantics in text similarity measurement

E	Semantic clustering	Signature detection	Data set	Top N	Precision (%)	Recall (%)	F_1-score (%)
E_0	Default	n/a	News	10	72.51	72.36	72.43
E_1	Default	None	Email	10	55.94	57.54	56.73
E_2	Email	None	Email	10	59.03	58.01	58.51
E_3	Default	Pre-trained	Email	10	42.28	41.99	42.13
E_4	Default	Trained	Email	10	53.65	54.88	54.26
E_5	Email	Trained	Email	10	58.31	55.45	56.84

F_1-scores of E_1 versus E_2 and of E_4 versus E_5, we note a rather small but persistent difference of about 2 % (F_1). The small factor is partly due to the fact that we only fitted the distributional semantics. As described earlier, the distributional semantics module only works as an extension to the general pruned index which is similar to a thesaurus. All other parameters like the very important word frequencies are kept identical. Considering the size of the two corpora (one with 6,000 lines of email text and the other with 1 million lines of generic newspaper text), this effect is nonetheless surprising, especially since prior work [6] on the effect of corpus size and the quality of semantic relation extraction showed a direct logarithmic dependency of both. From these findings it seems that domain adaptation is at least as important as corpus size. Hence it can be predicted that, on larger personal email collections, the performance will increase for the personalized semantic model. At the same time, performance is likely to stay the same for the trained model or even degrades if the discourse of the personal communication moves away from general topics. While the distributional semantic model benefits from on-device adaptation our generic signature detection algorithm needs to be trained on specific emails to actually perform well. This can easily be seen by means of Experiment E_4 and E_5 which clearly outperform E_3.

In summary, we have shown that for a task like signature detection per-user adaptation is a requirement in order to provide acceptable performance. Furthermore, even for algorithms like semantic clustering, we have initial evidence showing that domain adaptation and on-device training can be more important than corpus size and design.

In regard to our discussion so far it seems strange that E_4 and E_5 do not outperform E_1 and E_2 as clearly as E_3. This effect is, however, due to the relatively small size of our email gold standard. As noted earlier, there is a strong correlation between senders and topics in the gold standard. Since many topics in the email set contain only a discussion of a few selected people, signatures turn out to be good features for association. On a more complete email inbox a user will most probably have discussions on different topics with the same persons over the course of time. This is clearly an example of how challenging it is to scope gold standards correctly to reflect the final use cases.

3.3 (Unofficial) Semantic Text Similarity Experiments

In an effort to provide evaluation data on standard datasets we also ran our system against the shared *Semantic Text Similarity* task (SemEval workshops of 2012 and 2013 [1, 2]). The STS evaluations consist of sentence pairs associated with scores between 0 and 5 to indicate their observed semantic relatedness. Here 0 is not related and 5 is identical on a semantic level. The evaluation pairs were gathered from different sources such as video subtitles, statistical machine translation, news headlines or glossaries (cf. [1, 2]). In 2012 35 teams submitted a total of 88 system runs, in 2013 34 teams participated in the similar CORE sub-task submitting 89 system runs in total.

Before considering the results, it is necessary to stress the following facts. Neither did we optimize our system to perform well on sentence-level comparisons (it is optimized to compare entire emails with each other), nor do the STS evaluations take the amount of CPU cycles to compute the similarity, RAM usage or resources disk space usage into account when comparing the quality of the algorithms. Our system, for example, has to have response times for computing the 10 most similar documents in a collection of 1,000 documents of less then several hundreds of milliseconds (exact numbers depending on specific phone models). The linguistic processing also has to never exceed more than a couple of megabytes of RAM usage (irrespective of other factors, e.g., document size), because any android process using more than 35 megabytes will be terminated by android without warning and the application has to perform many other operations such as user interface processing or data base handling at the same time. Additionally, download sizes of applications still typically range within a few dozen megabytes, with anything bigger than that considered to be too large. Hence, we were forced to optimize our entire language models to be around one to three megabytes per language.

In 2012 most algorithms with decent results relied on a mixture of different string- or word-based similarity algorithms combined with external knowledge sources (e.g. WordNet), or complex linguistic tools (e.g. parsers, PoS-taggers, machine learning and sometimes even machine translation (an overview is provided in [1, p. 392])). UKP, the best performing system in 2012, which also served as a baseline for 2013 and still performed very well, uses a combination of many different string and word based features, explicit knowledge from WordNet and Wikipedia and a distributional thesaurus leading to more than 300 different vectors [4]. We do not have access to exact numbers, but it is likely a computationally rather complex approach to the problem.

Our results were matched to the gold standard using the WEKA toolkit [22] to calculate a simple supervised linear regression for each dataset. To work around one issue of our system when dealing with very short sentences consisting solely of stopwords, we defined a return value of 5 in cases where the lowercased words between both sentences all matched. The results are summarized in Table 2. Our unofficial results show us at rank 57 of 88 in the 2012 challenge. A run on the freely available parts of the 2013 gold sets using regression parameters obtained on 2012

Table 2 Pearson correlation on STS 2012 and 2013 datasets including the rank

Dataset	Year	Correlation	Rank
MSRpar	2012	0.37072	77
MSRvid	2012	0.69851	57
SMTeuroparl	2012	0.49825	17
OnWN	2012	0.57392	59
SMTnews	2012	0.42229	7
Weighted mean	2012	0,52431	57
Headlines	2013	0.62939	52
OnWN	2013	0.63346	26
FNWN	2013	0.14607	78

data shows us again at a mid range position. Due to the distribution restrictions on one 2013 dataset, we cannot provide an overall mean.

Our system appears to have a particular weakness regarding the headline paraphrases of the MSRpar dataset of 2012 and the FrameNet-WordNet glossary pairs in the FNWN dataset, which we will look into in future improvement rounds. Presently, our system also does not contain several modules yet, because they are still under development or because they are excluded for reasons of efficiency, such as unsupervised lemmatization, our time extraction module, or the extraction module for measurement units.

4 Information Extraction

The extraction of information out of textual data is a complex and challenging task in NLP. The field of information extraction has evolved over the last few decades. Starting with early conferences focusing on message understanding [19, 20], information extraction was firstly defined as the task of extracting entities, events, and relations. These entity types range from names of locations, persons and organizations, temporal expressions and monetary values; the defined relation types cover logical relations between those entities (e.g. *Located-In(Org, Loc)*).

Today, research efforts in this area are closer to a deep understanding of language [10, 37]. This includes detection and extraction of manifold entities which are not necessarily restricted to named entities (e.g. entities from the biomedical domain [25] or arbitrary expressions for practical purposes such as shopping list items). For many use cases extraction of simple logical relations is only a minor step towards more complex structures. We consider detecting and understanding flight confirmations as one representative example. This task includes extraction of numerous entities (e.g. airports, departure / arrival time and flight durations) and logical relations (e.g. departure airport / time, arrival airport / time), sanity constraints (e.g. departure time must be earlier than arrival time, time zone(s)), and additional information (e.g. baggage restrictions, flight booking references).

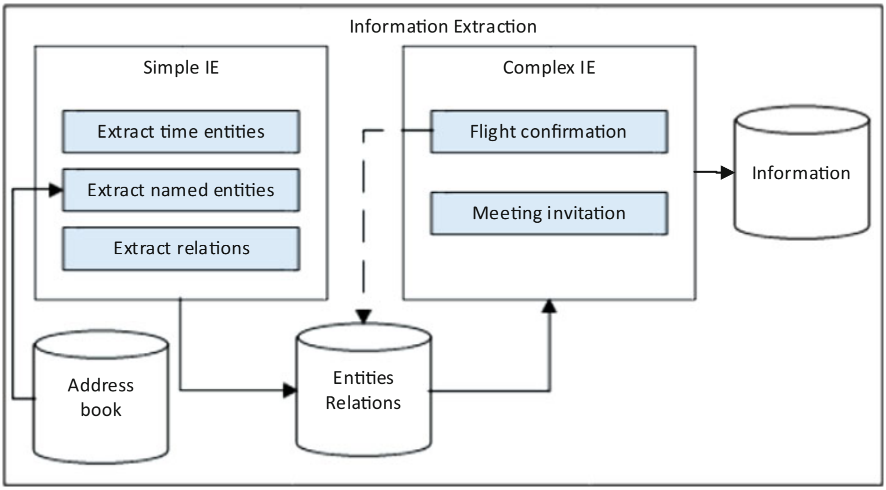


Fig. 2 An overview of the information extraction system

In order to master information extraction for these and similar complex tasks, we created a hybrid information extraction system (see Fig. 2) combining the strength of established statistical approaches for *Named Entity Recognition* (NER) with rule-based approaches to ensure special properties of the information to be extracted without neglecting characteristics of personal data and restrictions of applying this system on mobile devices. Furthermore, the information extraction system should be able to pass extracted information arbitrarily from one layer to another (e.g. from the layer of complex information extraction back to the layer of simple information extraction). This leads to an iterated information extraction process capable of using features which were not there during a first run through the text (e.g. after the text type extractor found out that the current text is a flight confirmation then the normal NER module could use this information to increase the possibility of three letter abbreviations to be airport codes).

4.1 (Named) Entity Recognition

We divide the extraction of (named) entities into two approaches: the rule-based and the statistical one. We developed flexible and robust rule-based solutions for detecting temporal expressions (e.g. 04/12/2014, yesterday in the evening, today) and measurements (e.g. $100, 12 km).

Statistical approaches are used for detecting person names, places and company names. Our system is based on the well-established Maximum Entropy classifier architecture [9] rather than on *Conditional Random Fields* (CRFs) due to lower resource usage. It is trained using many of the features that have achieved competitive results such as lexical features, PoS-tags, affix information, and word structure information [35].

In contrast to common NER systems that operate on PCs, we have to deal with a number of limitations:

- Mobile devices have slow CPUs and limited amount of memory which poses serious limitations on the model size and, thus, on number of features that can be processed.
- Supervised PoS-taggers do not perform well on noisy data [36] and require large models.
- User-generated data (e.g. SMS, emails, Twitter messages) are very noisy: they contain a high number of (personal) abbreviations, no reliable case information, and highly personalized personal language patterns.

We developed a number of strategies to circumvent these major challenges. We incorporated a fast, unsupervised PoS-tagging producing a small model size, and which does not require huge amounts of memory (our model is based on [29]). Furthermore, unsupervised PoS-tagging adds a semantic differentiation to the PoS which provides additional information to the NER model (e.g. there is no class containing all normal nouns, there are multiple semantic classes containing normal nouns separating words like weekdays, cities, countries, first names, last names etc.). Additional semantic classes for all words within a sentence provide more information about the content of a sentence to the classifier which is more robust against idiosyncratic language structures and case information (which is sometimes missing and sometimes completely broken due to automatic corrections made by T9 (text on 9 keys)). Most importantly, our unsupervised PoS-tagging naturally adapts itself to any domain by virtue of being completely unsupervised. Hence, domain adaptation for our system is reduced to the (still sometimes challenging) task of collecting a sufficient amount of text samples of the domain in question. For example, when we trained a PoS-tagging model on a biomedical domain corpus, it produced POS-tags with highly useful distinctions between genes as one “POS-tag” vs. diseases and medical ingredients as another cluster instead of lumping everything together into one huge “noun” cluster.

In addition to these restrictive adaptations, our NER system benefits from the usage of personal resources that are available on-device.

4.2 Integration of Personal Resources

Although there are massive restrictions regarding running NER systems on mobile devices, is it possible to use their personal resource storage to improve the information extraction system. The NER classifier can be backed up by various approaches as exemplified subsequently.

4.2.1 Address Book

What kind of information is available on all mobile phones? The obvious answer is: an address book. It contains the names and telephone numbers of the most frequent communication partners of a person. It may also contain addresses and user names of this person in social networks or communication platforms like Facebook, Twitter and Skype. Especially the latter ones are typically very hard to detect with traditional name extraction methods. Merely adding all the contained names and places to the personal name list of the NER system easily increases the perceived performance of the entire system.

Nevertheless, the disambiguation of names is still a necessary step since people do not always use full names when writing messages, for example when the receiver knows which *Christian* or *Mr. X* the sender refers to due to context or previous messages. Even the address book might contain multiple persons sharing the same first or last name (typical due to family name sharing). Since this disambiguation step is not important during the NER step, it is postponed to a semantic disambiguation module.

Another important benefit is the automatic linking to alternate names of the people stored in the address book. Most pre-trained NER modules are unable to detect nicknames used in social networks or communication apps like Skype or Twitter unless they are trained on respective pre-annotated corpora with adapted feature sets that do not exist for multiple languages. They are even less able to resolve nicknames to real names without a complex system which collates and keeps track of information across different sources. Using the information of an address book that contains real-world names linked with various nicknames of various social networks enables our NER module to use matches of these nicknames as on-device training instances.

4.2.2 Exploiting the Personal Corpus

The personal corpus contains many documents of personal communication. Although it is not possible to train a complex classifier directly on-device, it is possible to exploit the personal corpus in several ways. The main challenge of a NER system is to classify unseen entities. It also has to rely on the context within neighboring tokens, the sentence or even the document. This context can be extended to the complete personal corpus, too. In many cases, the direct context does not contain enough information to classify an entity into its correct class. Most approaches take context tokens occurring within a window of two tokens around the target token into account. Such contexts seldom contain sufficient information to decide the target word's type (e.g. "[quiet around] X [for the]"). Looking at the complete sentence unveils more valuable context and the decision becomes easier: "After months of hype, it's been pretty quiet around X for the last few weeks." But still, X could be a person, company, place or even a fancy new candy.

In news texts the document would most likely contain X in other positions with more context information than in this sentence and it should be possible to classify the token(s) based on the additional information. Short messages, emails, and tweets are very short because in personal communication explanations of common knowledge is left out as discourse participants are relatively familiar with each other. Hence, further references to entities might not exist in the document.

This is where the personal corpus can be used to find more evidence for entities that the user might be interested in. The personal corpus contains all communication data and thus, the entities which are important / interesting for the user. Our assumption is that crawling previously extracted entities can yield a high chance to find more contexts for the token(s) in question that makes it easier to extract proper name type (or conversely to prove that no name is present in a given text).

4.2.3 Combining Precomputed NER Models with Personal Models

Precomputed models work well when they are applied on data that is similar to the training corpus used. When applied to personal communication the personal language of the user becomes a non-negligible factor. Complete on-device NER model creation is not possible due to hardware restrictions and the lack of annotated data. Furthermore, the model would need a very long time to collect enough instances to make confident decisions. Thus, we propose a combined model consisting of a pre-computed model and a less complex personal (e.g. a Naïve Bayes classifier [16]) model which is iteratively trained directly on-device.

The user should get the possibility to revise extractions made by the NER system. A convenient GUI should be able to provide inconspicuous feedback loops for extracted names (e.g. the possibility to cross out false positives, verify true positives and add false negatives upon result list presentation). In addition, the personal model can be trained using this user feedback additionally to boost instances like address book matches that are not recognized by the general model and confident classification results made by the general model in order to provide more training instances.

Both models are members of an ensemble [17] and are weighted differently. Regular evaluation runs that validate the personal model against the general model provide a reliability score for the personal model. This score is used to weight the personal model and to account for its steadily improving classification performance. The combined model then applies both classifiers and combines the resulting scores according to the respective weights of the classifiers.

In order to demonstrate the crucial information gain obtained by including the address book as a list of known names we sampled some statistics from an authentic email box. This email box contains 913 emails in which the trained NER model

Table 3 Experiments on incorporating user feedback during iterative training of a NER module

	Precision(%)	Recall(%)	F-Score(%)
Without user feedback	77.67	56.99	65.74
With user feedback	**89.68**	**74.40**	**81.33**

tagged about 16.8 k tokens as person names.[8] More than 8.6 k of them are tokens covered by matching against the address book (which only contains 65 different person name tokens). Although the address book is not very comprehensive, this experiment shows that the most common names in personal communication are covered by address books. Furthermore, it provides a considerable amount of positive training instances, even in fragmentary contexts like signatures or in SMS that lack proper case patterns. We thus expect further work to prove our hypothesis that even lightweight classifiers which are trained on-device can improve the performance of pre-trained models significantly.

We also performed an experiment to evaluate the impact of user feedback on iterative NER module training. Iterative training uses a precomputed model and applies it to the user's data. An intuitive GUI provides the possibility of manual corrections (e.g. deny false positive matches, annotate missing matches or correct the boundaries of detected entities) to the user. This feedback is used to improve the training data of the iteratively trained classifier which afterwards achieves superior result to the precomputed one. In our experiment on English data (mixture of user-generated data and web texts with 1,139 annotated person names) the user feedback resulted in an increase of more than 15 % (F1) (see Table 3), although the user was told to only focus on the most frequent mistakes. This also means that each user may improve the performance of the NER module until the quality fulfills the user's expectations.

5 Conclusions

We have built a robust and efficient NLP system for mobile devices. We have shown, partially demonstrated, and briefly explained the most important design decisions which make any mobile-centric NLP system different from typical NLP systems that have access to virtually unlimited CPU and RAM resources, and which deal with clean data.

Even though our on-going task is to continually develop our system, there are already several clear conclusions that can be drawn as follows:

[8] This equals an average of more than 18 person name tokens per mail. Besides many false positives (classified incorrectly due to broken contexts in log file texts, quoted mails and signature parts), most of them can be verified as correct.

Personal language variation Our work shows that the influence of personal language variation on the performance of an NLP system is indeed very strong and goes very deep through many layers of language processing. The influence appears to be about as strong as that of corpus size. This also implies that optimal performance can only be achieved if sufficient amounts of everyday communication content from a particular user can be input into the system. Doing so can optimize both the analysis of personal language usage as well as corpus size.

Personal data Personal data constitutes an extremely rich and varied domain in terms of the data types present and the amount of data available. Large amounts of personal data can easily be used for automatic, on-device refinement of various NLP subsystems.

On-device processing Since personal data is very sensitive in terms of data protection and privacy issues, the best way to build trust with the users is to not send their data anywhere but to analyze it directly on the device. Thus, the results of the analyses belong to the user and remain on her/his device. We observed that many users displayed a certain amount of uneasiness when they discovered the high quality of some (from their perspective) more advanced analyses such as the *compelling* automatic semantic recipient suggestion (or alert in case the user clearly added the wrong recipient) based on the contents of the email they are writing. These emotions did not develop further into an outright rejection because the users were always fully aware of the fact that their personal data would not be transmitted anywhere.

At least baseline It is common knowledge that many NLP solutions require context for their decisions. However, in real-world applications many contexts are completely misleading. For example, attempting to run a PoS-tagger on the textual lines found in a flight ticket confirmation will result in very low hit rates because in this case the algorithm degenerates to plain vocabulary look-up performance levels at best. Hence, a robust real-world NLP solution will provide at least baseline solutions for all levels of processing. Even just detecting a flight ticket as non-valid input into the PoS-tagger and NER system will prevent many false detections.

Robustness and language scale-out Real-world applications have to deal with very diverse and noisy data. Besides employing advanced adaptive preprocessing techniques, we focus our efforts on robust unsupervised algorithms such as an unsupervised PoS-tagger or automatically computed semantic clusters. Although these algorithms are known to produce output of lower quality when being applied under artificial conditions of scientific evaluations, we emphasize robustness, performance and model size over the last percent of theoretical performance. If the employed algorithm is good enough in terms of user acceptance, we simply use it rather than that we focus solely on theoretical numbers. Another benefit of using unsupervised algorithms is the possibility to scale up the number of supported languages and domains with minimal manual effort. This is a central aspect for a rather small company addressing an international market.

We found that the increased processing power of the mobile devices and the possibilities of compressing NLP subsystems converged sufficiently to allow ubiquitous NLP systems to be pre-installed on even mid-range mobile devices without hurting their core performance in any noticeable ways. Nevertheless, the processing power of current mobile devices is still not sufficient to run all possible or necessary core analyses directly on-device. Moreover, some analyses also require initial background corpus processing in order to produce good results. The most obvious example in this regard is the background corpus-based word frequency computation for keyword extraction where the deviation of personal language usage from default language usage patterns is estimated. Another example is the PoS-tagger clustering and training which definitely requires a corpus larger than what would could be compiled from limited personal language usage samples, and which also requires considerable CPU and space resources that are not yet available on current mobile devices. We have found a good balance between server-side and client-side processing which is facilitated by the fact that all server-side processing can be done as a pre-analysis that results in compressed models eventually shipped to the user.

References

1. Agirre E, Cer D, Diab M, Gonzalez-Agirre A. Semeval-2012 task 6: A pilot on semantic textual similarity. In:*SEM 2012: The first joint conference on lexical and computational semantics – vol 1: Proceedings of the main conference and the shared task, and vol 2: Proceedings of the 6th International workshop on semantic evaluation (SemEval 2012), pp 385–393, Montréal, Canada, 7–8 June 2012
2. Agirre E, Cer D, Diab M, Gonzalez-Agirre A, Guo W (2013) *Sem 2013 shared task: Semantic textual similarity. In: Second joint conference on lexical and computational semantics (*SEM), vol 1 Proceedings of the main conference and the shared task: semantic textual similarity, pp 32–43, Atlanta, Georgia, June 2013
3. Azzopardi L, Balog K (2011) Towards a living lab for information retrieval research and development: a proposal for a living lab for product search tasks. In: Proceedings of the 2nd international conference on multilingual and multimodal information access evaluation, CLEF'11. Springer, Berlin, pp 26–37
4. Bär D, Biemann C, Gurevych I, Zesch T (2012) UKP: Computing semantic textual similarity by combining multiple content similarity measures. In: *SEM 2012: The first joint conference on lexical and computational semantics – vol 1: Proceedings of the main conference and the shared task, and vol 2: Proceedings of the 6th international workshop on semantic evaluation (SemEval 2012), pages 435–440, Montréal, Canada, 7–8 June 2012
5. Barlow M (2013) Individual differences and usage-based grammar. Int J Corpus Linguist 18(4):443–478
6. Bordag S (2007) Elements of knowledge-free and unsupervised lexical acquisition. Phd, University of Leipzig, Leipzig
7. Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka ER, Mitchell TM (2010) Toward an architecture for never-ending language learning. In: Proceedings of the conference on artificial intelligence (AAAI), pp 1306–1313, AAAI Press
8. Carvalho VR, Cohen WW (2004) Learning to extract signature and reply lines from email. In: Proceedings of the conference on email and anti-spam

9. Chieu HL, Ng HT (2002) Named entity recognition: a maximum entropy approach using global information. In: Proceedings of the 19th international conference on Computational linguistics, pp 1–7, Morristown, NJ
10. Collobert R, Weston J (2008) A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on machine learning, Helsinki, Finland
11. Corbett P, Batchelor C, Teufel S (2007) Annotation of chemical named entities. In: Proceedings of the annual meeting of the ACL, pp 57–64, ACL
12. De Saussure F (1916) Cours de linguistique générale. Payot, Lausanne/Paris
13. Dumais S, Cutrell E, Cadiz JJ, Jancke G, Sarin R, Robbins DC (2003) Stuff I've seen: a system for personal information retrieval and re-use. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval, SIGIR '03, pp 72–79, New York
14. Dunning T (1993) Accurate methods for the statistics of surprise and coincidence. Comput Linguist 19(1):61–74
15. Ferrucci D, Brown E, Chu-Carroll J, Fan J, Gondek D, Kalyanpur A, Adam L, J William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, Chris Welty (2010) The ai behind watson - the technical article. AI Mag 31
16. Fleischman Michael, Hovy E (2002) Fine grained classification of named entities. In: Proceedings of the 19th international conference on Computational linguistics, pp 1–7, Morristown, NJ
17. Florian R, Ittycheriah A, Jing H, Zhang T (2003) Named entity recognition through classifier combination. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, vol 4, pp 168–171, Edmonton
18. Goldhahn D, Eckart T, Quasthoff U (2012) Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In: Proceedings of the 8th international conference on language resources and evaluation (LREC'12), pp 759–765
19. Grishman R (1995) The NYU system for MUC-6 or where's the syntax? In: MUC6 '95: Proceedings of the 6th conference on Message understanding, pp 167–175, Morristown, NJ
20. Grishman R, Sundheim B (1995) Design of the MUC-6 evaluation. In: MUC6 '95: Proceedings of the 6th conference on message understanding, pp 1–11, Morristown, NJ
21. Grouin C, Rosset S, Zweigenbaum P, Fort K, Galibert O, Quintard L (2011) Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In: Proceedings of the 5th linguistic annotation workshop, LAW V '11, pp 92–100, Stroudsburg, PA, 2011
22. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: An update. SIGKDD Explor Newsl 11(1):10–18
23. Heyer G, Bordag S (2007) A structuralist framework for quantitative linguistics. In: Alexander Mehler and Reinhard Köhler, editors, Aspects of Automatic Text Analysis / Series: Studies in Fuzziness and Soft Computing. Springer, Berlin, New York
24. Heyer G, Quasthoff U, Wittig T (2008) Text mining: Wissensrohstoff text – konzepte, algorithmen, ergebnisse. W3L-Verlag, Herdecke
25. Kim JD, Pyysalo S, Ohta T, Bossy R, Nguyen N, Tsujii J (2011) Overview of BioNLP shared task 2011. In: Proceedings of the BioNLP shared task 2011 workshop, pp 1–6. ACL, 2011
26. Kim J (2012) Retrieval and evaluation techniques for personal information. PhD thesis, Graduate School of the University of Massachusetts, 2012
27. Kiss T, Strunk J (2006) Unsupervised multilingual sentence boundary detection. Comput Linguist 32(4):485–525
28. Kushmerick N (2000) Wrapper verification. WWW 3(2):79–94
29. Lamar M, Maron Y, Johnson M, Bienenstock E (2010) SVD and clustering for unsupervised pos tagging. In: Proceedings of the ACL 2010 conference short papers. Uppsala, pp 215–219
30. Lampert A, Dale R, Paris C (2009) Segmenting email message text into zones. In: Proceedings of the 2009 conference on empirical methods in natural language processing: vol 2 - vol 2, EMNLP '09. Stroudsburg, PA, pp 919–928

31. McMenamin GR (2002) Forensic linguistics: Advances in forensic stylistics. CRC Press, London
32. Richardson R, Smeaton AF, Murphy J (1994) Using WordNet as a knowledge base for measuring semantic similarity between words. In: Technical Report, Proceedings of AICS conference, 1994
33. Rudman J (1997) The state of authorship attribution studies: Some problems and solutions. Comput Hum 31(4):351–365
34. Salton G (1989) Automatic text processing: The transformation, analysis, and retrieval of information by computer. Addison Wesley, Reading
35. Tjong Kim Sang EF, Meulder FDe (2003) Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Daelemans W, Osborne M (eds), Proceedings of CoNLL-2003, volume pages, pages 142–147
36. Schierle M (2011) Language engineering for information extraction. Phd, University of Leipzig, Leipzig
37. Schuetze H, Scheible C (2013) Two svds produce more focal deep learning representations. CoRR, abs/1301.3, 2013
38. Varelas G, Voutsakis E, Euripides, Petrakis EG, Milios EE, Raftopoulou P (2005) Semantic similarity methods in WordNet and their application to information retrieval on the web. In: 7 th ACM international workshop on web information and data management (WIDM 2005), pp 10–16, ACM Press, 2005
39. Witschel Hf (2004) Terminologie-extraktion – möglichkeiten der kombination statistischer und musterbasierter verfahren. Ergon Verlag, Würzburg
40. Witschel Hf (2007) Multi-level association graphs - a new graph-based model for information retrieval. In: Proceedings of the HLT-NAACL-07 Workshop on Textgraphs-07, New York, 2007

Natural Language Processing Supporting Interoperability in Healthcare

Frank Oemig and Bernd Blobel

Abstract Improving personal health (care) for pervasive and ubiquitous health services requires the involvement of principals with different skills, education, social, cultural, ethical and legal background. They have to cooperate and communicate the necessary information in an interoperable way, so that the information can be used on all sides. The resulting necessary interoperability among human beings and of course systems requires the management and communication of knowledge. This knowledge management should be based on appropriate and hopefully shared ontologies. Natural languages are an efficient and powerful means in representing meaning, knowledge and skills. They balance between special sentence structures and generative flexibility, allowing for unambiguous representation of real world concepts used in communication. This paper provides an overview of the current state of the art functionality in NLP with regard to its application in health information systems interoperability. Therefore this paper deals less with an in-depth analysis of the methodologies currently developed in NLP and rather motivates for using NLP in real-life use cases.

1 Introduction

After a detailed introduction into the challenges, some methodologies and underlying associated models [1], we envision in this paper the application of NLP methods to enable interoperability in the healthcare domain. This will be demonstrated with the exchange of patient-related medical and clinical data. Health system interoperability is the most complex challenge in the field of (medical) informatics, because a huge set of specific information systems developed from different vendors must be integrated in an environment with many different specialties, each of which takes itself as the most relevant/important one. Consequently, the

F. Oemig (✉)
Agfa HealthCare GmbH, Konrad-Zuse-Platz 1-3, 53227 Bonn, Germany
e-mail: frank.oemig@agfa.com

B. Blobel
Medical Faculty, University of Regensburg, Regensburg, Germany

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_7

integration introduces a big challenge on the participating stakeholders. (Reversely, if the responsible stakeholder is not participating—caused by whatever reason—or following its own strong interests, introducing such an interoperability specification becomes almost impossible as can be observed within the German healthcare market and its "Selbstverwaltung".)[1]

For improving quality, safety, social care, efficiency of care processes under the well-known conditions in healthcare [2], health systems in developed and developing countries are challenged to undertake fundamental paradigm changes from both organizational and methodological perspectives [3, 4]. At organizational dimension, this challenge addresses the move from an organization-centric through a process-centric to a person-centric way of managing health. Regarding the methodological perspective, health care is slowly turning from its traditional disease-centric phenomenological approach through evidence-based medicine enabled by stratification of population for specific, clinically relevant conditions to personalized care considering health status, conditions and contexts of the individual. Common to both paradigm changes is that they require increasingly multi-disciplinary, domain-crossing interoperability between a growing number of involved principals such as persons, organizations, devices, applications, or even components. Changing objectives, structure and behavior of the business case in question requires new approaches to the analysis of requirements as well as to the design, implementation and maintenance of possible solutions. In this paper, necessary basics, principles and methodologies, but also resulting techniques and technological means for enabling interoperability among the aforementioned principals will be introduced and discussed.

2 Methods

There are different definitions of interoperability used among organizations addressing the interoperability challenge [5, 6]. (1) Merriam Webster is stating that "Interoperability is the ability of a system (as a weapons system) to use the parts or equipment of another system [7]". (2) The Institute of Electrical and Electronics Engineers (IEEE) originally formulated in its Standard Computer Dictionary that "Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged [8]". Meanwhile, IEEE has extended its view, applying the interoperability definition also to organizational, social and political systems.

Health Level Seven (HL7) [9] is an international organization working on data exchange communication standards in healthcare. It was founded in the USA in

[1] In Germany, improvements in the healthcare domain are rarely enforced by jurisdictional requirements, but by the strong commitments given by the involved stakeholders. However, these have conflicting interests preventing any real progress.

1987 with the intent to create a communication standard for healthcare related information. Up to now a family of different standards has been established ranging from simple data exchange and document management through visual application integration up to context management and knowledge representation.

HL7 International, self-attributed as *THE* Health Care Interoperability Standards Developing Organization (SDO), refers to those two definitions ($1 + 2$). While the second definition focuses on information exchange supported by information technology, the paradigm changes mentioned in the introduction require the inclusion of other domains and especially the real world of end users.

2.1 The Interoperability Challenge

For achieving interoperability, several pre-requisites have to be met such as common interest between the players involved, common architectural frameworks and reference models, an agreed set of coordinated reference terminologies/ontologies and a harmonized development process [10]. In the following, we will highlight the different issues.

A person-centric and personalized approach to health and social care integrates many different regulated and non-regulated stakeholders that are bound to diverse legal, cultural and social background. Even further, these stakeholders are also impacted by different education, knowledge and experiences that are represented using their languages with explicit terminologies based on domain-specific ontologies. Following the information cycle of communication and cooperation [11] among health professionals (HP), respectively the health care establishment (HCE) on the one hand and the patient on the other hand, must be defined, first concentrating on the business objectives and the business case. In that context, inputs and environmental conditions, but also the process outcome must be observed. The observations (data) must be interpreted to represent the underlying concept (the related information) and to derive appropriate actions that are cooperatively performed by the different involved actors according to the aforementioned business objectives and outcome (Fig. 1, left). This cycle of communication and cooperation between interoperating actors is recursively performed transforming simple observations into data (Fig. 1, middle). It is repeated to achieve a business objective or to maintain it. In our business case, this is curing patients and keeping them healthy (Fig. 1, right). The information cycle approach is based on Shannon and Weaver [12], Brillouin [13] and Wiener [14] considering areas and levels of human activities. At this point it is worth mentioning that communication and cooperation is not only performed among health professionals (HPs) and patients, but also between

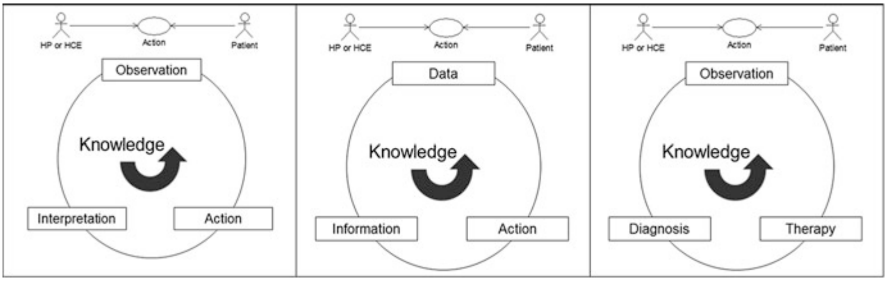


Fig. 1 Information cycle (after van Bemmel and Musen [11], changed)

HPs and HPs who may be associated with different institutions from the same or different sectors.[2]

As the conduction of appropriate observations, but even more the interpretation of data to information and the derivation of appropriate actions is a matter of knowledge and skills, the interoperability process between the actors is a combination of communication and cooperation based on shared knowledge. To make it clear, the information cycle is performed on each actor's site, i.e. upon receipt of data the information cycle is executed with communicating the outcome triggering the information cycle on the other side again. Therefore, each participating actor must have the same knowledge in order to perform the same actions or to draw the same conclusions. In other words, different knowledge will lead to diverging conclusions although the underlying data is the same. The process especially for communicating the information may be supported by Information and Communication Technology (ICT) by providing highly sophisticated communication standards. The HL7 interoperability definition is not sufficient here, because it only considers the support of the communication processes, which originally lead to the set of HL7 Communication Standards[3] known as "HL7 version 2.x" or "HL7 Version 3". The required knowledge for this kind of communication can either be shared in advance or must be provided at runtime. In modern health settings, sharing knowledge with all participating actors in advance is practically almost impossible due to the dynamics, flexibility, complexity and heterogeneity of the business case. Consequently, knowledge must be provided just in time and incorporated into the communication.

[2] In the healthcare domain, the ambulatory (outpatients with general practitioners) and stationary sector (inpatients with hospitals) have different requirements impacting the communication between those stakeholders.

[3] HL7 started its work on enabling and establishing intra-hospital communication, i.e. supporting the data exchange among systems within the same hospital. For this purpose the world-wide most accepted communication standard is "HL7 version 2.x". A few years after HL7's inception work, "HL7 Version 3" has started to support inter-hospital and cross-sectoral communication.

Table 1 Interoperability levels from both the information and the organizational perspective [4]

Information perspective		Organizational perspective
Interoperability level	Instances	Interoperability level
Technical interoperability	Technical plug & play, signal- & protocol compatibility	Light–weight interactions
Structural interoperability	Simple EDI, envelopes	Information sharing
Syntactic interoperability	Messages and clinical documents with agreed upon vocabulary	
Semantic interoperability	Advanced messaging with common information models and terminologies	Coordination
Organizations/service interoperability	Common business process	Collaboration, cooperation

If knowledge and skills for interpreting the data and taking the right action in cooperative processes is available, sharing of data alone, i.e. data exchange performed by the ICT system is sufficient for achieving comprehensive interoperability. If an actor has knowledge and skills to provide his contribution in the business process but not the knowledge about the concepts interconnected with its process steps, exchange of data in combination with its associated semantics is required. If the aforementioned knowledge and skill for actions to be taken are not available, the service must be provided as well. The different levels of sharing knowledge and skills for cooperative business processes result in different levels of interoperability to be provided by the communication and collaboration platform as shown in Table 1.

Even advanced projects of IEEE or national programs on interoperability such as the HIE initiative of the U.S. Presidents have been focused on the ICT interoperability challenges. Figure 2 illustrates an extended interoperability challenge covering the real world business case and its ICT support.

In Fig. 2, rectangles represent the ICT-related part of the interoperability challenge, i.e. the business case's informational representation in the ICT domain, while the ellipses represent the ICT-independent part, i.e. the business process described from different domains' perspectives on the real world system. *IF* describes the interface and *DR* the data representation of applications (*APP*), *BC* represents the real world business case. The numbers represent the following interoperability

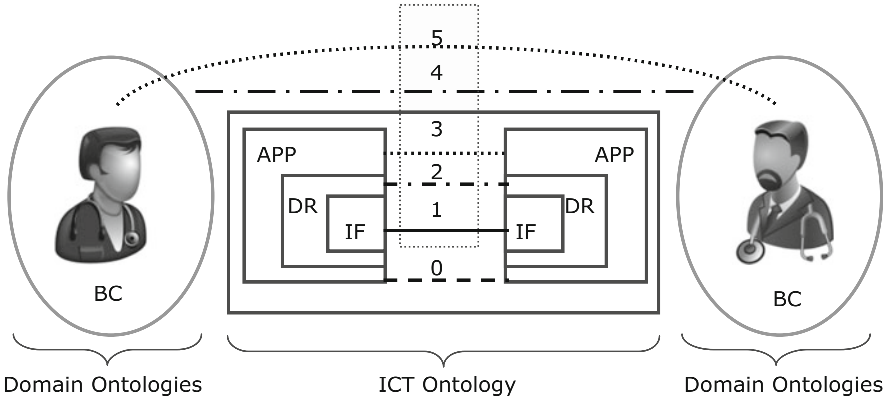


Fig. 2 ICT-independent and ICT-related interoperability challenges

levels: 0—technical; 1—structural and syntactic; 2—semantic; 3—service. Level 1 and 2 are application agnostic, as expressed, e.g., for the related HL7 protocols.

Following the object-oriented paradigm, level 2 and 3 are interconnected as an object combines attributes and operations on them. The first 4 levels are ICT dependent and represented using ICT ontologies. For integrating the reason of our endeavor, the real world, ICT supported business case (paper-based, handshaking, physical business processes) must be considered, representing knowledge using domain-specific terminologies and methodologies based on domain-specific ontologies (level 4). Finally, the human factor of education, skills, experiences, social and psychological aspects, etc., but also common sense knowledge must be considered (level 5). When two human actors perform ICT facilitated communication and cooperation for achieving a common business objective, the relevant information is processed through all the levels of the layered system down the sender side and up the receiver side. This communication stack is exemplified by the so-called ISO/OSI-Model [15], which is the foundation for the whole internet.

2.2 The Language Challenge

As introduced so far, interoperability is always bound to the availability of knowledge that must be shared either a priori or on the fly during the business process. Therefore, modern health systems require the formalization of knowledge, the development of formalized and accepted ontologies, the decentralization of knowledge and decision support, and the development of ontology harmonization tools and mechanisms.

The representation and communication of something is a matter of language. Humans are using natural languages for this purpose, while machines depend on simpler formal languages explicitly expressing the knowledge [4].

A language is a set of words (signifiers) composed of letters out of an alphabet and being defined by formation rules over that alphabet, called a grammar. Semantics considers the relation between those signifiers and their denotation. Terms of natural languages have semantics, i.e. meaning and rules are expressed implicitly in the terms and their relations. Contrary, formal languages do not have semantics *per se*. For bearing semantics, they have to be enriched with constructs such as operators and rules. The system of logics belongs to the formal language family.

Symbols, operators, and interpretation theory give sequences of symbols meaning within Knowledge Representation (KR). A key parameter in choosing or creating a KR is its expressivity. The more expressive a KR is, the easier and more compact is it to express a fact or element of knowledge within the semantics and grammar of that KR. However, more expressive languages are likely to require more complex logic and algorithms to construct equivalent inferences. A highly expressive KR is also less likely to be complete and consistent. Less expressive KRs may be both complete and consistent [4, 16].

This characteristic results in the complexity problem of formal language and reasoning systems with the lack of computability, at the same time losing the consistency of the language system. Natural languages are not only efficient in representing meaning, shared knowledge, skills, and experiences. They provide an optimum between restriction to special structure and generative power enabling the rich and nevertheless sufficiently unambiguous representation of real world concepts, enabled and supported of course by common sense knowledge. This is one of the reasons for investing in natural language processing and not only relying on the formal representation of medical facts. Figure 3 provides an overview on KR languages or ontology types, evolving from informal KR languages up

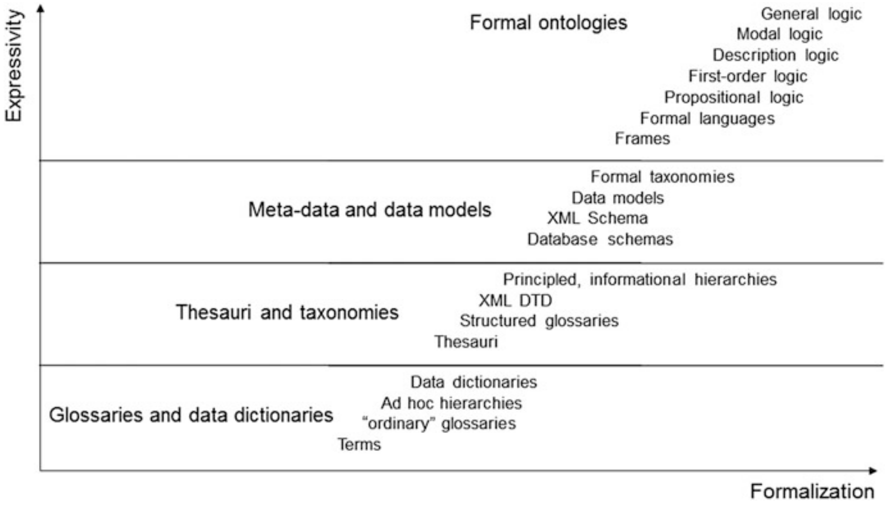


Fig. 3 Expressivity and formalization of ontology types, after [17], changed

to formal ones. Regarding the aforementioned formalization and expressivity of more abstract, explicit ontology types, formal languages could be defined without restriction as it could be done with an unrestricted Turing Machine. But this is less suited for the intended purpose of expressing natural language concepts because it allows expressing everything without being complete and consistent. In the other extreme, a formal language generated with a highly restricted Markov Process is not useful, as it can only express quite simple concepts. Finally, we should not forget other languages such a graphical ones (Unified Modeling Language—UML) or even such exotics such as body and sign language, all of them meeting the aforementioned principles.

To get closer to the vision of comprehensive interoperability, the ontological representations used by different domain experts for representing entities in reality must be harmonized. For that purpose, the ontological representation must be provided at a level of formalization and expressivity which guarantees common understanding, i.e. expresses meaning and rules as explicit as needed depending on education, skills, and experiences of the actors involved (Fig. 3). In other words, ontology harmonization requires a top-level ontology with higher generalization or according to Fig. 2 an ontology type of at least one level up in expressivity and formalization.

More details on knowledge representation and management in the context of interoperability can be found, e.g., in [4].

2.3 The Natural Language Processing Challenge

Traditionally, health professionals (prefer to) collect their information from the perspective of its uniqueness and specialty as narrative text, despite of the administrative and legal enforcement of classifying and coding that information for statistical assessment or interoperability challenges as discussed above. So, clinical narratives are a rich source of knowledge. Natural Language Processing (NLP) methods and algorithms are used to extract that knowledge by, e.g., Named Entity Recognition (NER) and Information Extraction (IE) [18]. Wikipedia defines named entities as "atomic elements in text" belonging to "predefined categories such as the names of persons, organizations, locations, date, time, quantities such as age, weight, speed, distance, monetary values, percentages, etc." [19]. IE addresses the extraction of names, entities, relations and events from semi-structured or unstructured text to store them in a structured way like databases or use them for, e.g., document indexing, structured search, question answering, summarization, automatic translation, opinion mining/sentiment extraction, text data mining over extracted relationships. The general principle followed when automatically processing free text is indexing that entity, semantically interpreting, and storing/searching/retrieving it. In the biomedical domain, Critical Assessment of Information Extraction Systems in Biology (BioCreAtIvE) [19] enabled studies on automatic identification of biomedical entities, e.g., genes and proteins, in text drawn from various databases such as PubMed, FlyBase, genome databases, etc. Another challenge to evaluate systems for information retrieval and related technologies in the genomics domain—also held under auspices of NIST (National Institute for Standardization and Technology)—is the TREC Genomics track. A series of workshops and conferences for evaluating NLP solutions, not explicitly dedicated to the biomedical field, is the Message Understanding Conference (MUC) series, organized by NIST as well [18].

3 Towards NLP in Healthcare

If we take physicians, no matter whether they are working in the ambulatory or stationary sector, they do not like the obliged care-related bureaucracy. Their primary business is caring and curing patients. Maintaining a health record for each patient is normally only done for jurisdictional reasons. Anything beyond is hardly accepted and only performed on external pressure. (A very good means frequently applied is a financial stimulus.) However, according to further specializations of each individual, data exchange in the context of the necessary communication and cooperation is inevitable. So, from an interoperability perspective a document exchange with formatted text or even structured information is performed. The most desired form of interaction is by speech because of its simplicity, while direct calls are rarely performed. A speech-to-text conversion on the sender's side and a text-to-speech conversion on the receiver's side are desired to enhance comfort and convenience. Hence, a human–human interaction as designated in Fig. 4 can be imagined [20].

At this point, having this kind of electronic indirect communication does not contradict to the exchange of structured information in the middle. On the contrary, they are mutually supportive in the whole process of information exchange.

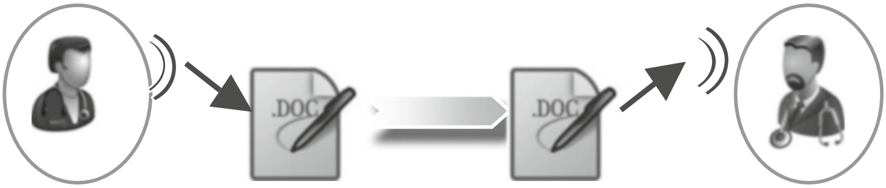


Fig. 4 Human–human interaction

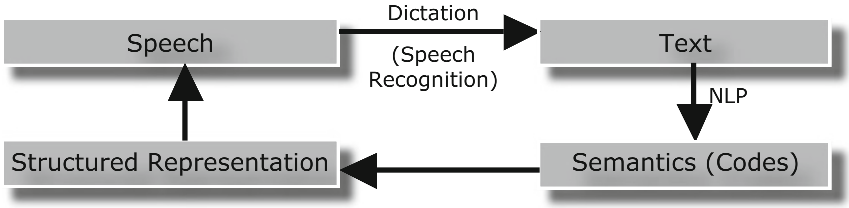


Fig. 5 Round trip

3.1 Round Trip

To enable communication, a round trip as depicted in Fig. 5 becomes necessary. It includes the speech to text recognition and the generation of speech out of text or even information models, but this should not be considered in depth in this paper. A broader space should be covered by semantics, i.e. how the text is related to content and represented in form of formal information models [21–23].

As such, an ontology providing a deeper knowledge about the underlying domain becomes necessary. However, only modern and sophisticated information systems make use of ontologies at all. In most cases, those ontologies are created in a proprietary fashion from scratch. Any alignment with publicly available reference ontologies like the Basic Formal Ontologies (BFO) [24] would slow down the development process and enforce long-lasting harmonization efforts, both of which are not appreciated.

3.2 Phases of NLP

Figure 6 elaborates a little deeper on the necessary NLP steps from sound waves to the ultimate reasoning based on the patient's context. This figure is a combination of the Paninian's approach as described by Kak [25] and further information provided by Rindflesch [26]. It demonstrates the importance of the utilization of appropriate lexical information that is provided in form of classifications, terminologies and ontologies [27]. A specific catalog for the medical domain thereof, which is providing a very detailed list of entries, is the so-called Alpha-Identifier for Diagnosis [28], because it contains all possible inflections of diagnosis terms. It covers almost all terms used by German physicians, and it is closely linked to the German modification of ICD 10[4] [29]. The weaknesses of classifications applying, e.g., ICD 10 should be overcome by using very detailed ontologies, which not only provide a

[4] ICD is the International Classification of Diseases as defined by the World Health Organization (WHO) and is used to code information about diagnoses.

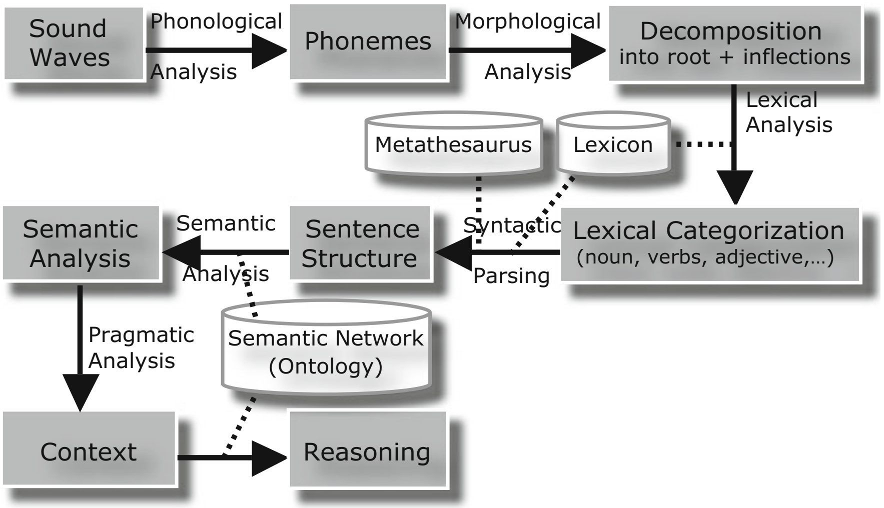


Fig. 6 NLP phases [2, 3]

very comprehensive list of concepts but also the relationships among them. For this reason, the best currently available catalog Snomed CT (Systemized Nomenclature for Medicine—Clinical Terms) [30] should be used. It contains approx. 400,000 clinical concepts with more than 1.5 million relations.

Since it also contains relations, Snomed CT cannot only be used as a lexicon to support a syntactic analysis, but also as a semantic network to verify the information conveyed.

3.3 From Speech to Text Fields

A normal dictation process is designed to convey continuous information for a specific topic like a discharge letter in healthcare starting with a short introduction (name of the person/patient, topic and other simple meta-data) followed by the real data like diagnosis, anamnesis, observations, procedures and recommendations. But sound waves are not only used to transport this kind of continuous information, they can also be reduced to short phrases that represent commands or only pieces of information. This is comparable to punctuation commands (like “comma”, “period”) in a standard dictation process.

An appropriate NL process should cover this requirement and benefit from additional information being provided within the application, which helps to control the processing and appropriate distribution of the information. For this purpose, the

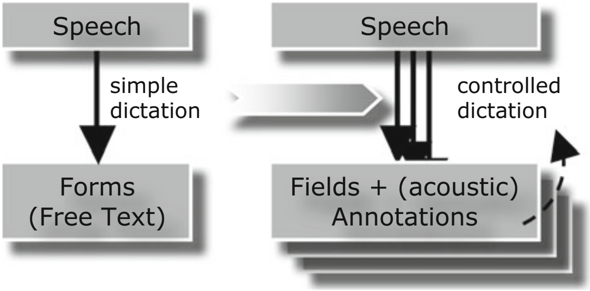


Fig. 7 Overview of dictation

fields should be annotated with “acoustic information”—in analogy to “semantic annotations” which should provide additional details about the semantic concepts being kept within a specific field. In this regard, we have to migrate from a simple command oriented approach to a more sophisticated one as is shown in Fig. 7.

This annotation helps to identify the fields where the successive information should be sent to. This is especially applicable for those applications that highly deal with structured data to prepare a special report. This approach is opposed to those applications that make use of standard text processing tools to generate and issue a report.

3.4 From Text to Codes

Besides identifying the correct field, another challenge is the correct translation of text into codes. An ontology as a knowledge database may help to control and verify this process.

As mentioned before, the biggest, most sound and solid ontology is Snomed CT [30]. Such an ontology will also support the identification of the associated field, if an appropriate semantic annotation is provided which is marked as “controlled NLP” in Fig. 8.

At this point, it must be kept in mind that in healthcare applications are preferred which provide a huge flexibility by allowing each physician to create his/her own set of forms. However, such opportunity is increasing the burden to provide additional (semantic) information for each field in form of annotations. Most physicians take the comfort of this WYSIWYG[5] form approach for granted without realizing and understanding the resulting problems the self-made forms will provide to the interfaces. Without such an annotation, each field contains a value, but it does not

[5] WYSIWYG: “What You See Is What You Get”.

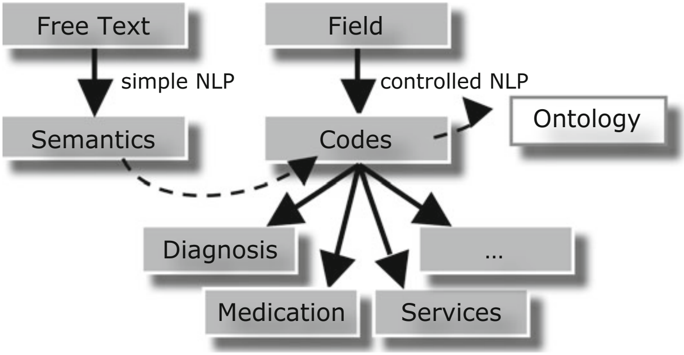


Fig. 8 Overview about NLP

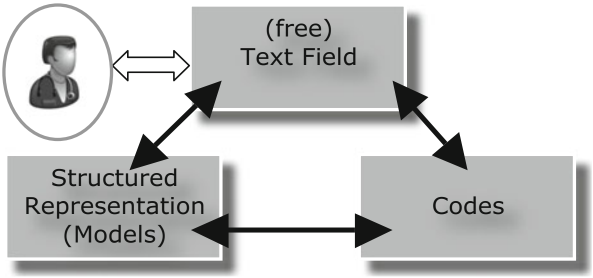


Fig. 9 Triangle

provide any usable information—at least not for the underlying interfaces. So, how does an interface know that "80" in a specific field denotes the body weight and not the systolic blood pressure? Hence, the application's flexibility for its users on the one hand is diminishing the chance for real interoperability on the other hand.

Currently, physicians prefer a NL based information exchange, while in the near future a structured representation in form of models being annotated with appropriate codes will become necessary (Fig. 9). For user acceptance, this process has to be automated as much and as early as possible. But on the other hand, an improved automatism will support the real clinician's work by releasing them from purely administrative tasks like coding.

3.5 From Codes to Structured Data

When trying to export data, a totally different aspect comes into play: for all data the appropriate authorization must be checked, i.e., whether the patient's consent grants the permission to do so. Hence, an export filter has to examine the security annotation for authorization based on policies; otherwise it must be

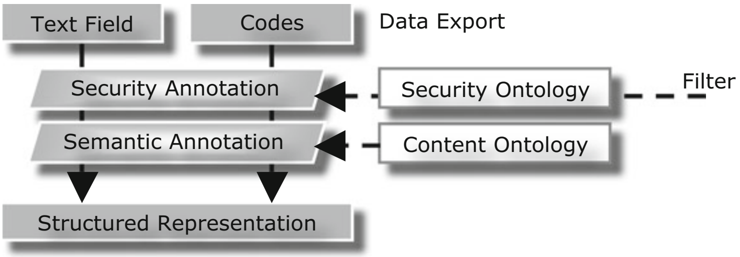


Fig. 10 Ontology controlled export of data

blocked. A suitable standard is the HL7 Security Labeling Services [31]. Currently provided in form of vocabulary tables, the method must be enhanced to an ontology-based one as well, so that the security information operates as a filter and allows for a comprehensive processing (Fig. 10, upper part).

The creation of the data structure must be controlled by an ontology considering the aforementioned semantic annotation of the fields as presented in Fig. 10.

In order to process data correctly it is not sufficient to be informed about a single data item ("blood pressure"), it must also be exemplified what the information consists of. In our example it must be specified that the overall blood pressure consists of two components, denoting the systolic and diastolic blood pressure. All these relationships must be defined in form of information models. In HL7 several tools were developed supporting the creation of models. These are known as the Model-Driven Health Tools (MDHT) [32]. The visual notation is based on UML, but enhanced with minor improvements like coloring to increase the readability for physicians. Figure 11 shows a snippet from the Clinical Document Architecture (CDA), Release 2 [33].

HL7 has established an overall methodology to specify and describe information models for different domains: HL7 Development Framework [37]. This framework is an extension to UML and allows for detailed specifications of information models. Those models are abstracted from "unessential" details that are normally expressed by additional attributes. Good examples are data types like person names and addresses. The components of a name or address are hidden on this level.

For a developer those details are injected when transformed using Implementable Technology Specifications into so-called Implementation Guides describing the concrete contents of messages or documents for an information interchange. An example for an Implementation Guide is the German "VHitG-Arztbrief" [38] also known as discharge letter.

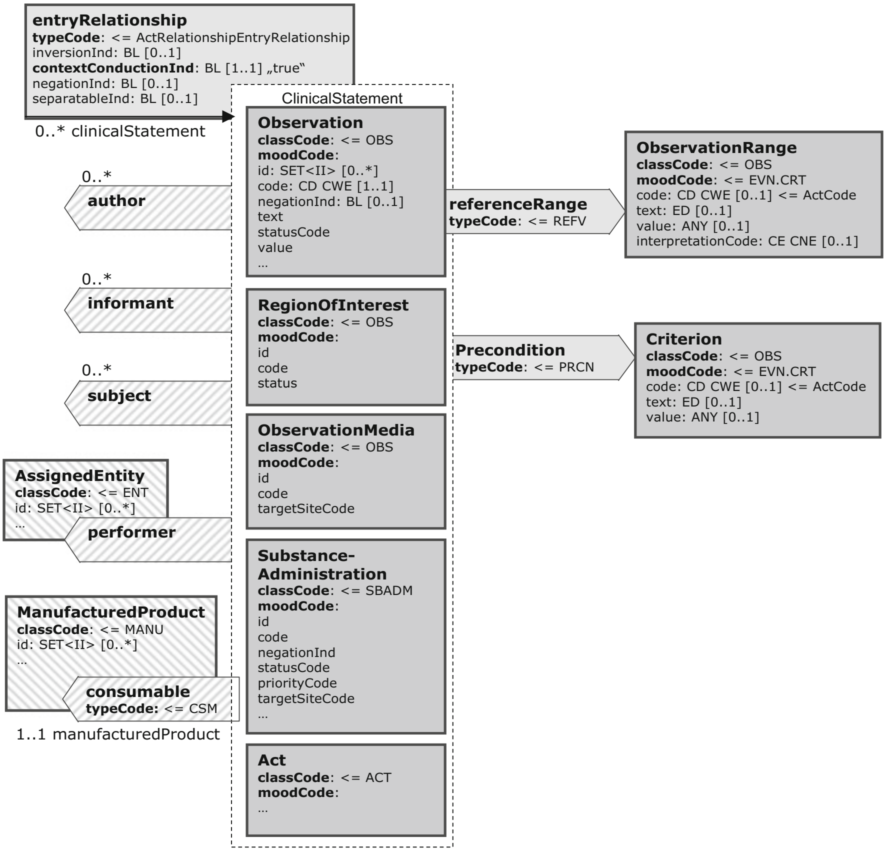


Fig. 11 Snippet of HL7 Version 3 domain information model for structure documents [33–36]

3.6 Importing Data

Reversely, importing data works the same way (Fig. 12). The ontology controls, which information is taken and where, respectively in which field, it is placed.

Comparable to the export, imported data must be associated with the appropriate security information bound to roles and persons ensuring the patient's consent regarding security and privacy, but this is left out in the aforementioned figure.

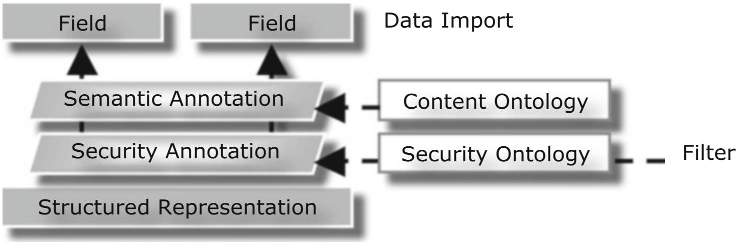


Fig. 12 Ontology controlled import of data

3.7 *Data Exchange*

HL7 International has defined a data format called Clinical Document Architecture (CDA) [33] to enable data exchange in healthcare at different level of structuring. (A snippet is shown in Fig. 11.) The primary focus of this interoperability specification is a gentle migration from simple letters (like a discharge letter) into a semantically enriched form which allows for an automatic extraction of data. This approach takes into account that a simple exchange of text can easily be implemented by the different vendors, because no complex data structures or information models must be considered, and an overarching information exchange is done between humans facilitating simple text. Structured information is then embedded into sections as indicated by “Entry” under “Option 4” in Fig. 13. (Figure 11 provides a rough idea of how the different structures may look like. The interested reader is directed to the original specification here.)

Enriching the content of those documents can be done for, or by, those systems that are capable of handling this structured information. However, it requires an enhancement to the architecture of the applications, which is not always possible, especially not for old and outdated legacy systems.

Meystre et al. [39] presents a totally different approach. She combines two different standards to take advantage of NLP techniques: Those two standards are the Clinical Document Architecture (CDA) and Graph Annotation Format (GrAF) [40]. The different options thereof are shown in Fig. 13, and are caused by XML itself, because two different XML schemas for those two standards must be combined without compromising each other:

Option	Description
1	Inserting the annotation into each individual section of the text
2	Insert the whole annotation as a new part to the document
3	Maintain the annotation as a separate document but combine the XML schemas
4	Keep both documents separate

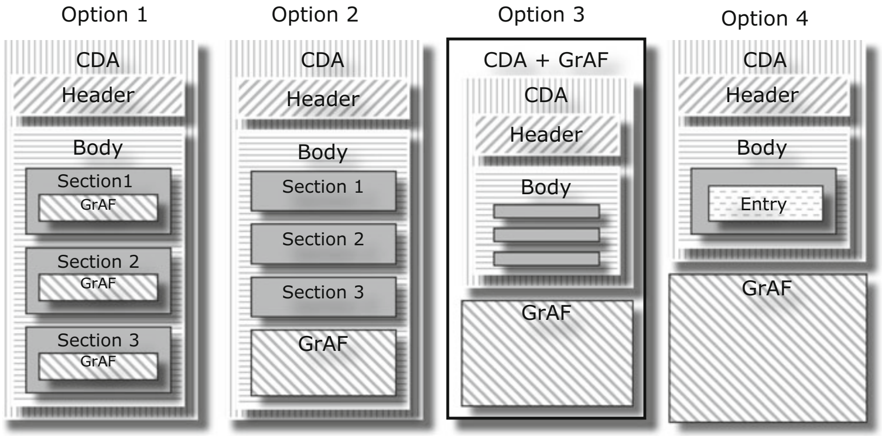


Fig. 13 Annotated data after Meystre et al. [39] (changed)

Another problem beside schema validation is the subsequent manipulation of the document content. Therefore, the addition of annotations must be done before the document is signed; otherwise the digital signature is rendered invalid.

Consequently, she argues in favor of option 3 by providing a top-level schema integrating the original ones. In principle, this approach is similar to, and also works with, PDF to ensure a correct visualization of the content. The corresponding specification is called ZUGFeRD [41] and tries to encapsulate an XML-based instance of the semantic content like CDA [42]. Hence, we have to integrate three different XML schemas in a similar fashion.

3.8 Semantic Translations

As many different actors such as persons, organizations, devices, applications, components, and objects are involved in future-proof health and social services delivery, the underlying knowledge must be formally represented based on the different domains' as well as ICT ontologies. However, there is another challenge at least as crucial as the former one: the inclusion of patients or citizens before becoming patients, and their relatives, caretakers at different level of knowledge and skills as well as domain experts. Based on their education, knowledge, skills, experiences, and social as well as environmental context, those different stakeholders reflect a business case and the system to run that process differently. Thereby, they explicitly use their "own, non-agreed terminologies" based on their implicit "own, non-agreed ontologies", i.e. only the terminological part of the ontologies is used verbosely. Such behavior creates the challenge of semantic translations between languages and horizons. Here, the next level of NLP comes into play.

4 Discussion

The paper draws a bow from paradigm changes in health systems causing new interoperability challenges between the manifold actors involved through the need and ways of sharing knowledge and skills in heterogeneous environments up to properly representing resources and automatically harmonizing them using advanced NLP means. In that context, the deployment of ontologies and their management is inevitable. ICT-specific interoperability solutions must be completed by also considering human users and their business context, but also their education and proficiencies, culture, expectations, etc.

Currently, within the HL7 community the development of the third release of CDA has started. The first drafts are available. It is intended to overcome some identified weaknesses especially in the field of representing structured data because the current expressivity is limited. Therefore, the current work would be a good opportunity to allow for combining CDA with GrAF, officially endorsing the use of NLP technology in the field of information exchange in healthcare, but at least to raise the awareness for such a need.

Also, in advanced communication environments, data exchange is based on controlled vocabularies. This must be even further enhanced to ontologies so that an automatic process can take advantage out of it without manual adjustments.

The requirements for health professionals will change in the future. They will be released from stupid manual coding of data by hand, thereby increasing the time they can spend for treating patients. An important prerequisite thereof will be the use of NLP techniques in healthcare.

Acknowledgements The authors are indebted to their colleagues from SDOs such as IHTSDO and HL7 for open-minded discussions and their kind cooperation and valuable support.

References

1. Blaschke C, Hirschman L, Yeh A, Valencia A (2003) Critical assessment of information extraction systems in biology. Comp Funct Genomics 4:674–677. Published online in Wiley InterScience (www.interscience.wiley.com). doi:10.1002/cfg.337
2. Garets G (2001) Gartner's vision for healthcare: the next 10 years. Presentation at the HL7 Plenary and Working Group Meeting in Orlando, FL
3. Hall JA, Blobel B (2013) Paradigm changes in health lead to paradigm changes in pathology. Stud Health Technol Inform 179:38–50
4. Blobel B (2013) Knowledge representation and management enabling intelligent interoperability – principles and standards. In: Blobel B, Hasman A, Zvárová J (eds) Data and knowledge for medical decision support. Series studies in health technology and informatics, vol 186. IOS, Amsterdam, pp 3–21
5. Blobel B, Gonzalez C, Oemig F, Lopez DM, Nykänen P, Ruotsalainen P (2010) The role of architecture and ontology for interoperability. In: Blobel B, Hvannberg EP, Gunnarsdóttir V (eds) Seamless care – safe care: the challenges of interoperability and patient safety in health care. Series studies in health technology and informatics, vol 155. IOS, Amsterdam, pp 33–39

6. Blobel B (2011) Ontologies, knowledge representation, artificial intelligence – hype or prerequisite for international pHealth Interoperability? In: Stoicu-Tivadar L, Blobel B, Marčun T, Orel A (eds) e-Health across borders without boundaries. E-salus trans confinia sine finibus. Series studies in health technology and informatics, vol 165. IOS, Amsterdam, pp 11–20
7. Merriam Webster Dictionary. http://www.merriam-webster.com/dictionary/interoperability. Last accessed 1 Apr 2014
8. Institute of Electrical and Electronics Engineers (IEEE) (1990) IEEE standard computer dictionary: a compilation of IEEE standard computer glossaries. New York
9. Health Level Seven (HL7). www.hl7.org
10. Blobel B (2010) Architectural approach to eHealth for enabling paradigm changes in health. Methods Inf Med 49(2):123–134
11. Van Bemmel J, Musen MA (1997) Handbook of medical informatics. Springer, Heidelberg
12. Shannon C, Weaver W (1959) The mathematical theory of communication. University of Illinois Press, Champaign
13. Brillouin L (1962) Science and information theory. Academic, Waltham
14. Wiener N (1948) Cybernetics. MIT Technology Press, Boston
15. DIN ISO 7498: Informationsverarbeitung Kommunikation Offener Systeme, Basis-Referenzmodell, DIN EN ISO 7498-1 ISO/OSI-Modell, Beuth Verlag, 1982
16. Chomsky N (1959) On certain formal properties of grammars. Inf Control 2:137–167
17. Rebstock M, Fengel J, Paulheim H (2008) Ontologies-based business integration. Springer, Berlin
18. NIST: Named Entity Recognition. www.nist.gov
19. Wikipedia. www.wikipedia.de
20. Medica 2013 (Bröckerhoff, Main, Geßner, Heidenreich, Mohr, Oemig): Einfacher Datenaustausch für Ärzte II – Wozu benötigt man überhaupt Interoperabilität und wie kann sie hergestellt werden?, discussion round 20.11.2013, https://www.youtube.com/watch?v=xMxXu-TH77U&feature=view_all. Last accessed 3 Apr 2014
21. Oemig F, Blobel B (2007) Semantic interoperability adheres to proper models and code systems: an examination of different approaches or score systems. In: Blobel B, Pharow P, Zvarova J, Lopez DM (eds) CeHR conference proceedings 2007: eHealth: combining health telematics telemedicine, biomedical engineering and bioinformatics to the edge, S. 97ff. ISBN: 978-3-89838-089-8 (Aka), ISBN: 978-1-58603-834-2 (IOS), pp 97–104
22. Detailed Clinical Models. http://wiki.hl7.org/index.php?title=Detailed_Clinical_Models, http://www.detailedclinicalmodels.nl/. Last accessed 3 Apr 2014
23. openEHR Archetypes. http://www.openehr.org/. Last accessed 3 Apr 2014
24. BFO: The Basic Formal Ontology. www.ifomis.org/bfo. Last accessed 30 June 2014
25. Kak S (1987) The Paninian approach to natural language processing. Int J Approx Reason 1:117–130
26. Rindflesch TC, Fiszman M (2003) The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. J Biomed Inform. doi:10.1016/j.jbi.2003.11.003
27. Oemig F, Blobel B (2010) Harmonizing the semantics of technical terms by the generic component model. In: 10th international special topic conference of the European federation for medical informatics in Reykjavík Iceland, 2–4 June 2010, pp 115–121. IOS. http://www.sky.is/efmi-stc-2010-.html. ISBN: 978-1-60750-562-5
28. DIMDI: Alpha-ID – Identifkationsnummer für Diagnosen. http://www.dimdi.de/static/de/klassi/alpha-id/. Last accessed 3 Apr 2014
29. DIMDI: ICD-10-GM. http://www.dimdi.de/static/de/klassi/icd-10-gm/index.htm. Last accessed 3 Apr 2014
30. Snomed CT: Systemized NOmenclature for MEDicine – Clinical Terms. www.ihtsdo.org. Last accessed 3 Apr 2014
31. HL7 Version 3 Standard: Privacy, Access and Security Services; Security Labeling Service, Release 1 (SLS). http://www.hl7.org/implement/standards/product_brief.cfm?product_id=360. Last accessed 3 Apr 2014

32. MDHT: Model-Driven Health Tools. https://www.projects.openhealthtools.org/sf/projects/mdht/, http://sourceforge.net/projects/oht-modeling/files/Releases/Runtime/. Last accessed 30 June 2014
33. HL7 Clinical Document Architecture. http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7, http://www.hl7.org/search/viewSearchResult.cfm?search_id=505296&search_result_url=%2Fdocumentcenter%2Fprivate%2Fstandards%2Fcda%2Fr2%2FCDA%5FR2%5FNormativeWebEdition2010%2Ezip. Last access 30 June 2014
34. HL7 Version 3. www.hl7.org. Last accessed 9 Apr 2014
35. Heitmann K. CDA und HL7 Version 3 – Nachrichten und Dokumente. http://sciphox.hl7.de/atwork/protokolle/sciphoxHL7v3o.pdf. Last accessed 15 July 2014
36. CorepointHealth. HL7 CDA Rel.2 D-MIM. http://www.corepointhealth.com/sites/default/files/CDA-R-MIM-Model.gif. Last accessed 15 July 2014
37. HL7 Development Framework, HL7 Inc. HL7. 2006. http://www.hl7.org/library/committees/mnm%5Chdf_workproduct/HDF%20Methodology%20Specification%2Ezip
38. VHitG-Arztbrief. http://www.bvitg.de/arztbrief.html. Last accessed 30 June 2014
39. Meystre SM, Lee S, Jung CY, Chevrier RD (2011) Common data model for natural language processing based on two existing standard information models: CDA+GrAF. J Biomed Inform. doi:10.1016/j.jbi.2011.11.018
40. Ide NM, Suderman K (2007) GrAF: a graph-based format for linguistic annotations. In: Proceedings of the first linguistic annotation workshop, Prague, pp 1–8
41. ZUGFeRD einheitliches Format für elektronische Rechnungen: http://www.ferd-net.de/front_content.php?idcat=231&lang=3. Last accessed 9 Apr 2014
42. CDA und PDF/A: http://wiki.hl7.de/index.php/IG:CDA_und_PDF/A3. Last accessed 9 Apr 2014

Deception Detection Within and Across Cultures

Veronica Perez-Rosas, Cristian Bologa, Mihai Burzo, and Rada Mihalcea

Abstract In this paper, we address the task of cross-cultural deception detection. Using crowdsourcing, we collect four deception datasets, two in English (one originating from United States and one from India), one from Romanian speakers, and one in Spanish obtained from speakers from Mexico, covering three predetermined topics. We also collect two additional datasets, one for English from United States and one for Romanian, where the topic is not pre-specified. We run comparative experiments to evaluate the accuracies of deception classifiers built for each culture, and also to analyze classification differences within and across cultures. Our results show that we can leverage cross-cultural information, either through translation or equivalent semantic categories, and build deception classifiers with a performance ranging between 60–70 %.

1 Introduction

The identification of deceptive behavior is a task that has gained increasing interest from researchers in computational linguistics. This is mainly motivated by the rapid growth of deception in written sources, and in particular in Web content, including product reviews, online dating profiles, and social networks posts [10].

To date, most of the work presented on deception detection has focused on the identification of deceit clues within a specific language, where English is the most

V. Perez-Rosas (✉)
University of North Texas, Denton, TX, USA
e-mail: veronicaperezrosas@my.unt.edu

C. Bologa
Universitate Babes-Bolyai, Cluj-Napoca, Romania
e-mail: cristian.bologa@econ.ubbcluj.ro

M. Burzo
University of Michigan-Flint, Flint, MI, USA
e-mail: mburzo@umich.edu

R. Mihalcea
University of Michigan, Ann Arbor, MI, USA
e-mail: mihalcea@umich.edu

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_8

commonly studied language. However, a large portion of the written communication (e.g., e-mail, chats, forums, blogs, social networks) occurs not only between speakers of English, but also between speakers from other cultural backgrounds, which poses important questions regarding the applicability of existing deception tools. Issues such as language, beliefs, and moral values may influence the way people deceive, and therefore may have implications on the construction of tools for deception detection.

In this paper, we explore within- and across-culture deception detection for four different cultures, namely United States, India, Romania, and Mexico. Through several experiments, we compare the performance of classifiers that are built separately for each culture, and classifiers that are applied across cultures, by using unigrams and word categories that can act as a cross-lingual bridge. Our results show that we can achieve accuracies in the range of 60–70 %, and that we can leverage resources available in one language to build deception tools for another language.

1.1 Related Work

Research to date on automatic deceit detection has explored a wide range of applications such as the identification of spam in e-mail communication, the detection of deceitful opinions in review websites, and the identification of deceptive behavior in computer-mediated communication including chats, blogs, forums and online dating sites [10, 11, 15, 16, 19].

Techniques used for deception detection frequently include word-based stylometric analysis. Linguistic clues such as n-grams, count of used words and sentences, word diversity, and self-references are also commonly used to identify deception markers. An important resource that has been used to represent semantic information for the deception task is the Linguistic Inquiry and Word Count (LIWC) dictionary [12]. LIWC provides words grouped into semantic categories relevant to psychological processes, which have been used successfully to perform linguistic profiling of true tellers and liars [9, 14, 20]. In addition to this, features derived from syntactic Context Free Grammar parse trees, and part of speech have also been found to aid the deceit detection [3, 17].

While most of the studies have focused on English, there is a growing interest in studying deception for other languages. For instance, Fornaciari and Poesio [5] identified deception in Italian by analyzing court cases. The authors explored several strategies for identifying deceptive clues, such as utterance length, LIWC features, lemmas and part of speech patterns. Almela et al. [1] studied the deception detection in Spanish text by using SVM classifiers and linguistic categories, obtained from the Spanish version of the LIWC dictionary. A study on Chinese deception is presented in [18], where the authors built a deceptive dataset using Internet news and performed machine learning experiments using a bag-of-words representation to train a classifier able to discriminate between deceptive and truthful cases.

It is also worth mentioning the work conducted to analyze cross-cultural differences. Lewis and George [6] presented a study of deception in social networks sites and face-to-face communication, where authors compare deceptive behavior of Korean and American participants, with a subsequent study also considering the differences between Spanish and American participants [7].

At difference from us, both studies analyze cultural differences using a statistical approach, where data was collected by interviewing participants and principal component analysis was applied to identify cultural aspects related with deception such as liars topic's choice, and gender differences. In this study we rely on machine learning techniques to build deception classifiers from written statements provided by true tellers and deceivers.

In general, related research findings suggest a strong relation between deception and cultural aspects, which are worth exploring with automatic methods.

2 Datasets

We collect four datasets for four different cultures: United States (English-US), India (English-India), Romania, and Mexico (Spanish-Mexico). Following [8], we collect short deceptive and truthful essays for three topics: opinions on Abortion, opinions on Death Penalty, and feelings about a Best Friend.

To collect both truthful and deceptive statements for the Abortion and Death Penalty topics we first instructed the participants to think they were participating in a debate, where they were asked to provide their truthful opinion about the topic. Secondly, we asked them to imagine a debate where they had to provide an opposite view from what they truly believed, thus generating false statements about the topic being discussed. In both cases, we asked them to provide plausible details and to be as convincing as possible. For the Best Friend topic, we collected the deceptive and truthful essays by first asking participants to provide a description of their best friend, and second asking them to describe someone they disliked as though he/she were their best friend.

In order to collect the English-US and English-India datasets, we used Amazon Mechanical Turk with a location restriction, so that all the contributors are from the country of interest (US and India). We collected 100 deceptive and 100 truthful statements for each of the three topics. To avoid spam, each contribution was manually verified by one of the authors of this paper.

For Spanish-Mexico, while we initially attempted to collect data also using Mechanical Turk, we were not able to receive enough contributions. We therefore created a separate web interface to collect data, and recruited participants through contacts of the paper's authors. The overall process was significantly more time consuming than for the other two cultures, and resulted in fewer contributions as shown in Table 1.

For the Romanian dataset we also used a separate web interface and participants were recruited through contacts of one of the paper's authors. Since participants were allowed to end their participation at any time, the final process resulted in a different number of contributions per each topic as shown in Table 1.

For all four cultures, the participants first provided their truthful responses, followed by the deceptive ones. Also, all contributors provided their responses for different topics in the same topic order: Abortion, Best Friend, and Death Penalty.

Table 2 shows sample statements from each dataset. Also, word count distributions for the four datasets are shown in Table 3. Interestingly, for all four cultures,

Table 1 Dataset distributions for four deception datasets

	English-US		English-IN		Romanian		Spanish	
Topic	D	T	D	T	D	T	D	T
Abortion	100	100	100	100	139	139	39	39
Best Friend	100	100	100	100	151	151	42	42
Death Penalty	100	100	100	100	145	145	94	94

Table 2 Sample statements from four deception datasets

Topic	Deceptive	Truthful
English-US		
Abortion	Abortion should not be an acceptable practice, ever. Precluding the life of an unborn child is dominating and nullifying their inalienable right to live...	Abortion should be a legal option for pregnant mothers. Of course, it needs to be very early in the pregnancy and the mother must give significant...
Best Friend	“John” Is a great person. John always puts himself before others. John never says derogatory remarks to people.	My best friend, we will call him “Bob” is a truly exceptional person. I can talk to Bob about anything and everything.
Death Penalty	Life is sacred. Who are we to end a life? People, even criminals, deserve to live. They deserve a second chance.	Sometimes, there are those who commit crimes so heinous that there is only one appropriate punishment.
English-India		
Abortion	I think abortion is needed. It should be done, if the life of the mother is in risk. It should also be done in other necessary circumstances. Abortion should...	In my opinion, abortion is very cruel. It is another form of murder. We have no right to end the life of an innocent child. So, abortion should be banned.
Best Friend	He is one of the best people I have met in my life. He has never troubled be in any way. At work, he never competes with me. I “hope” we remain friends...	He is my best friend in my life. He helped me in all my downs in my life as guiding and gives suggestions. He can understand me as anyone can and
Death Penalty	I disagree the act death penalty. No one has the rights to take the life of a human except God. Instead of death penalty...	Yes, of course I support death penalty. Only fear from death would prevent these crimes. In this modern era crime...

(continued)

Table 2 (continued)

Topic	Deceptive	Truthful
Spanish-Mexico (Translated)		
Abortion	Abortion is a legal thing.it needs to be appreciated in all the way. People should be encouraged to do an abortion.	Abortion is very cruel thing for all humans in the earth. Abortion is a big sin before God.
Best Friend	My best friend is very nice. I love spending time with her. We have always get along very well and we like each...	My best friend always listen to me. We have a lot of things in common. We always find time to talk to each other.
Death Penalty	Death penalty should be applied in all countries without mercy. Criminals should pay for what they have done	I think we should not decide about the life of another human being. The only one who can make such decision is...
Romanian (Translated)		
Abortion	I do not agree with abortion under any circumstances (or in exceptional cases, any request) because it is not moral...	Abortion can help women to avoid giving birth a child that could affect their life's. If a woman decides she does...
Best Friend	This person give me a sense of confidence, always coming up with new ideas that I like. Always supports...	My best friend knows me very well. He knows when I'm upset and something goes wrong. We got along...
Death Penalty	The death penalty is very brutal and should not take place in a civilized world. Although they are murderers...	I think the death penalty is the correct one because criminals do not think about the lives of others when they...

Table 3 Word count distribution between deceptive (D) and truthful (T) statements and average number of words per statement for four deception datasets

	English-US		English-IN		Romanian		Spanish	
Topic	D	T	D	T	D	T	D	T
Abortion	52	72	64	76	68	91	76	106
Best Friend	51	64	67	75	65	89	60	87
Death Penalty	56	68	74	85	70	92	63	97
Average	53	68	69	78	68	90	66	97

the average number of words for the deceptive statements is significantly smaller than for the truthful statements, which may be explained by the added difficulty of the deceptive process, and is in line with previous observations about the cues of deception [2].

3 Experiments

Through our experiments, we seek answers to the following questions. First, what is the performance for deception classifiers built for different cultures? Second, can we use information drawn from one culture to build a deception classifier for another culture? Finally, what are the psycholinguistic classes most strongly associated with deception/truth, and are there commonalities or differences among languages?

In all our experiments, we formulate the deception detection task in a machine-learning framework, where we use an SVM classifier to discriminate between deceptive and truthful statements.[1]

3.1 What is the Performance for Deception Classifiers Built for Different Cultures?

We represent the deceptive and truthful statements using two different sets of features. First we use unigrams obtained from the statements corresponding to each topic and each culture. To select the unigrams, we use a threshold of 10, where all the unigrams with a frequency less than 10 are dropped. We choose this threshold due their best performance in the reported experiments. Also, since previous research suggested that stopwords can contain linguistic clues for deception, no stopword removal is performed.

Experiments are performed using a ten-fold cross validation evaluation on each dataset. Using the same unigram features, we also perform cross-topic classification, so that we can better understand the topic dependence. For this, we train the SVM classifier on training data consisting of a merge of two topics (e.g., Abortion + Best Friend) and test on the third topic (e.g., Death Penalty). The results for both within- and cross-topic are shown in the last two columns of Table 4.

Second, we use the LIWC lexicon to extract features corresponding to several word classes. LIWC was developed as a resource for psycholinguistic analysis [12]. The 2001 version of LIWC includes about 2,200 words and word stems grouped into about 70 classes relevant to psychological processes (e.g., emotion, cognition), which in turn are grouped into four broad categories[2] namely: linguistic processes, psychological processes, relativity, and personal concerns. We also used a Spanish version of the LIWC lexicon [13] as well as a Romanian version [4]. A feature is generated for each of the 70 word classes by counting the total frequency of the words belonging to that class. The resulting features are then grouped into four different sets containing the LIWC classes subset corresponding to each of the four

[1] We use the SVM classifier implemented in the Weka toolkit, with its default settings.

[2] http://www.liwc.net/descriptiontable1.php.

Table 4 Within-culture classification, using LIWC word classes and unigrams

	LIWC					Unigrams	
Topic	Linguistic (%)	Psychological (%)	Relativity (%)	Personal (%)	All (%)	Within-topic (%)	Cross-topic (%)
English-US							
Abortion	72.50	68.75	44.37	67.50	73.03	63.75	80.36
Best Friend	75.98	68.62	58.33	54.41	73.03	74.50	60.78
Death Penalty	60.36	54.50	49.54	50.45	58.10	58.10	77.23
Average	69.61	63.96	50.75	57.45	69.05	65.45	72.79
English-India							
Abortion	56.00	48.50	46.50	48.50	56.00	46.00	50.00
Best Friend	68.18	68.62	54.55	53.18	71.36	60.45	57.23
Death Penalty	56.00	52.84	57.50	53.50	63.50	57.50	54.00
Average	60.06	59.19	52.84	51.72	63.62	54.65	53.74
Spanish-Mexico							
Abortion	73.17	67.07	48.78	51.22	62.20	52.46	57.69
Best Friend	72.04	74.19	67.20	54.30	75.27	66.66	50.53
Death Penalty	73.17	67.07	48.78	51.22	62.20	54.87	63.41
Average	72.79	69.45	54.92	52.25	67.89	57.99	57.21
Romanian							
Abortion	61.87	64.02	64.02	62.58	63.30	65.10	58.99
Best Friend	70.19	68.21	68.21	68.54	67.54	68.80	54.30
Death Penalty	64.13	66.55	66.55	64.48	65.51	63.79	57.27
Average	65.39	66.26	66.26	65.20	65.45	65.89	56.85

For LIWC, results are shown for within-topic experiments, with ten-fold cross validation. For unigrams, both within-topic (ten-fold cross validation on the same topic) and cross-topic (training on two topics and testing on the third topic) results are reported broad categories. We perform separate evaluations using each of the feature sets derived from broad LIWC categories, as well as using all the categories together. The accuracy classification results obtained with the SVM classifier are shown in Table 4.

Overall, the results show that it is possible to discriminate between deceptive and truthful cases using machine learning classifiers, with a performance superior to a random baseline which for all datasets is 50 % given an even class distribution. Considering the unigram results, among the four cultures, the deception discrimination works best for the English-US dataset, and this is also the dataset that benefits most from the larger amount of training data brought by the cross-topic experiments. In general, the cross-topic evaluations suggest that there is no high topic dependence in this task, and that using deception data from different topics can lead to results that are comparable to the within-topic data. An exception to this trend is the Romanian dataset, where the cross-topic experiments lead to significantly lower results than the within-topic evaluations, which may be partly explained by the high lexicalization of Romanian. Interestingly, among the three topics considered, the Best Friend topic has consistently the highest within-topic performance, which may be explained by the more personal nature of the topic, which can lead to clues that are useful for the detection of deception (e.g., references to the self or personal relationships).

Regarding the LIWC classifiers, the results show that the use of the LIWC classes can lead to performance that is generally better than the one obtained with the unigram classifiers. The explicit categorization of words into psycholinguistic classes seems to be particularly useful for the languages where the words by themselves did not lead to very good classification accuracies. Among the four broad LIWC categories, the linguistic category appears to lead to the best performance as compared to the other categories. It is notable that in Spanish, the linguistic category by itself provides results that are better than when all the LIWC classes are used, which may be due to the fact that Spanish has more explicit lexicalization for clues that may be relevant to deception (e.g., verb tenses, formality).

Concerning the specific accuracy for the deception class, we analyzed detailed accuracies per class, obtained by the best classifier from Table 4, which is the one built using only the Linguistic category from LIWC. Table 5 shows the precision, recall, and F-measure metrics obtained for the deceptive and truthful classes obtained by the classifier for each culture. From this table we can observe that for Spanish as well as for both English cultures, the identification of deceptive instances is slightly easier than the identification of truthful statements. For Romanian instead, the truthful instances are more accurately predicted than the deceptive ones. We further analyzed differences in word usage among true tellers and liars in each culture in Sect. 3.3.

Table 5 Classification accuracy per class for Linguistic category classifier

Topic	Precision	Recall	F-measure	Class
English-US				
Abortion	0.73	0.71	0.72	Deceptive
	0.72	0.73	0.72	Truthful
Best Friend	0.74	0.79	0.76	Deceptive
	0.77	0.72	0.75	Truthful
Death Penalty	0.60	0.58	0.59	Deceptive
	0.60	0.62	0.61	Truthful
English-India				
Abortion	0.55	0.59	0.57	Deceptive
	0.56	0.53	0.54	Truthful
Best Friend	0.68	0.68	0.68	Deceptive
	0.68	0.68	0.68	Truthful
Death Penalty	0.55	0.58	0.56	Deceptive
	0.56	0.54	0.55	Truthful
Spanish				
Abortion	0.73	0.73	0.73	Deceptive
	0.73	0.73	0.73	Truthful
Best Friend	0.69	0.77	0.73	Deceptive
	0.75	0.67	0.70	Truthful
Death Penalty	0.73	0.73	0.73	Deceptive
	0.73	0.73	0.73	Truthful
Romanian				
Abortion	0.66	0.55	0.60	Deceptive
	0.61	0.71	0.66	Truthful
Best Friend	0.66	0.61	0.63	Deceptive
	0.64	0.68	0.66	Truthful
Death Penalty	0.65	0.70	0.67	Deceptive
	0.67	0.62	0.65	Truthful

3.2 Can We Use Information Drawn from One Culture to Build a Deception Classifier in Another Culture?

In the next set of experiments, we explore the detection of deception using training data originating from a different culture. As with the within-culture experiments, we use unigrams and LIWC features. For consistency across the experiments, given that the size of the Spanish and the Romanian datasets is different compared to the two English datasets, we always train on the English-US dataset.

To enable the unigram based experiments, we translate the two English datasets into either Spanish or Romanian by using the Bing API for automatic translation.[3] As before, we extract and keep only the unigrams with frequency greater or equal to 10. The results obtained in these cross-cultural experiments are shown in the last column of Table 6.

In a second set of experiments, we use the LIWC word classes as a bridge between languages. First, each deceptive or truthful statement is represented using features based on the LIWC word classes grouped into four broad categories: linguistic process, physiological process, relativity, and personal concerns. Next, since the same word classes are used in all three LIWC lexicons, this LIWC-based representation is independent of language, and therefore can be used to perform cross-cultural experiments. Table 6 shows the results obtained with each of the four broad LIWC categories, as well as with all the LIWC word classes.

Note that we also attempted to combine unigrams and LIWC features. However, in most cases, no improvements were noticed with respect to the use of unigrams or LIWC features alone.

These cross-cultural evaluations lead to several findings. First, we can use data from a culture to build deception classifiers for another culture, with performance figures better than the random baseline, but weaker than the results obtained with within-culture data. An important finding is that LIWC can be effectively used as a bridge for cross-cultural classification, with results that are comparable to the use of unigrams, which suggests that such specialized lexicons can be used for cross-cultural or cross-lingual classification. Moreover, using only the linguistic category from LIWC brings additional improvements, with absolute improvements of 2–4 % over the use of unigrams. This is an encouraging result, as it implies that a semantic bridge such as LIWC can be effectively used to classify deception data in other languages, instead of using the more costly and time consuming unigram method based on translations.

3.3 What are the Psycholinguistic Classes Most Strongly Associated with Deception/Truth?

The final question we address is concerned with the LIWC classes that are dominant in deceptive and truthful text for different cultures. We use the method presented in [8], which consists of a metric that measures the saliency of LIWC classes in deceptive versus truthful data. Following their strategy, we first create a corpus of deceptive and truthful text using a mix of all the topics in each culture. We then calculate the dominance for each LIWC class, and rank the classes in reversed order of their dominance score. Table 7 shows the most salient classes for each culture, along with sample words.

[3] http://www.bing.com/dev/en-us/dev-center.

Table 6 Cross-cultural experiments using LIWC categories and unigrams

Topic	Linguistic (%)	Psychological (%)	Relativity (%)	Personal (%)	All LIWC (%)	Unigrams (%)
Training: English-US Test: English-India						
Abortion	58.00	51.00	48.50	51.50	52.25	57.89
Best Friend	66.36	47.27	48.64	50.45	59.54	51.00
Death Penalty	54.50	50.50	50.00	48.50	53.5	59.00
Average	59.62	49.59	49.05	50.15	55.10	55.96
Training: English-US Test: Spanish-Mexico						
Abortion	70.51	46.15	50.00	52.56	53.85	61.53
Best Friend	69.35	52.69	51.08	46.77	67.74	65.03
Death Penalty	54.88	54.88	53.66	50.00	62.19	59.75
Average	64.92	51.24	51.58	49.78	61.26	62.10
Training: English-US Test: Romanian						
Abortion	61.15	55.04	56.47	48.2	57.19	56.47
Best Friend	64.56	50.66	63.90	51.55	52.98	66.22
Death Penalty	61.72	48.96	64.13	47.93	58.27	60.34

Table 7 Top ranked LIWC classes for each culture, along with sample words

Class	Score	Sample words	Class	Score	Sample words
English-US					
Deceptive			Truthful		
Metaph	1.77	Die, died, hell, sin, lord	Friends	0.46	Buddies, friend
Other	1.46	He, her, herself, him	We	0.55	Our, ourselves, us, we,
You	1.41	Thou, you	Self	0.55	myself, our, ourselves, us
Humans	1.22	Baby, human, person	Optimism	0.65	accept, hope, top, best
Othref	1.18	He, her, herself, him	I	0.66	I, me, my, myself,
Negemo	1.18	Afraid, agony, awful, bad	Insight	0.68	Accept, believe, understand
English-India					
Deceptive			Truthful		
Negate	1.49	Cannot, neither, no, none	Friends	0.46	Buddies, companion, friend, pal
Physical	1.46	Heart, ill, love, loved,	We	0.55	Our, ourselves, us, we
Future	1.42	Be, may, might, will	Self	0.55	I, me, mine, my, myself
Negemo	1.37	Afraid, agony, alone, bad,	Optimism	0.65	Accept, accepts, best, bold,
Other	1.17	He, she, himself, herself	I	0.66	I, me, mine, my
Humans	1.08	Adult, baby, children, human	Past	0.78	Happened, helped, liked, listened
Spanish-Mexico					
Deceptive			Truthful		
Certain	1.47	Fiel(loyal), jamás (never)	School	0.32	Consejo(advice), estudiar(study)
Humans	1.28	Bebé(baby), persona(person)	Past	0.32	Compartimos(share), vivimos(lived)
You	1.26	Eres(are),estas(be), su(his/her)	Friends	0.37	Amigo/amiga(friend), amistad(friendship)
Negate	1.25	Jamás(never), tampoco(neither)	We	0.58	Estamos(are),somos(be), tenemos(have)
Other	1.22	Es(is), esta(are), otro(other)	Self	0.65	Conmigo(me), tengo(have), soy(am)
Othref	1.11	Eres(are),tiene(have), tuvo(had)	Optimism	0.66	Aceptar(accept), alegre(cheerfully)

Romanian					
Deceptive			Truthful		
Money	2.31	Bani(money), pret(price)	We	0.65	Ne(us,ourselves), noi(we), noastra(our)
Posfeel	1.95	Fericita(happy), zambetul(smile)	Religion	0.72	Cer(heaven), dumnezeu (god), suflet(soul)
Other	1.42	Ei/ele(they), insusi(oneself)	Family	0.73	Tata(dad),mamica(mother), familie(family)
Pronoun	1.34	Ei/le(they), ii(him), va(yourself)	Time	0.77	Oricand(always), momentul(time)
Optimism	1.29	Increderea(confidence), usoara(easy)	Past	0.80	Intalnit(met), ajutat(helped), traiasca(live)
Anx	1.23	Frica(fear), emotionala(emotional)	Friends	0.79	Prietenie(friendship), prieten(friend)

This analysis shows some interesting patterns. There are several classes that are shared among the cultures. For instance, the deceivers in all cultures make use of negation, negative emotions, and references to others. Second, true tellers use more optimism and friendship words, as well as references to themselves. An interesting finding is the use of the Religion and Family classes by Romanian true-tellers, which seems to be very related to cultural background, as religion is an important cultural component. In contrast with the other cultures, Romanian speakers use more positive feeling (Posfeel) and Optimism related words when expressing deceptive statements.

These results are in line with previous research, which showed that LIWC word classes exhibit similar trends when distinguishing between deceptive and non-deceptive text [9]. Moreover, there are also word classes that only appear in some of the cultures; for example, time classes (Past, Future) appear in English-India and Spanish-Mexico, but not in English-US, which in turn contains other classes such as Insight and Metaph.

4 Deception Detection Using Short Sentences

One limitation of the experiments presented in the previous section is that they all rely on domain-specific datasets, which may bias the deception detection. To address this potential concern, as a final experiment, we explore the detection of deception in a less-constrained environment, where the topic of the deceptive statements is not set a priori.

We collect and experiment with two datasets consisting of short open-domain truths and lies, contributed by speakers of English-US and Romanian.

For English, we set up a Mechanical Turk task where we asked workers to provide seven lies and seven truths, each consisting of one sentence, on topics of their choice. For Romanian, we designed a web interface to collect data, and recruited participants through contacts of the paper's authors. Romanian speakers were asked to provide five truths and five lies, again on topics of their choice. In both cases, the participants were asked to provide plausible lies and avoid non-commonsensical statements such as "A dog can fly." In addition to the one-sentence truths and lies, we also collect demographic data for the contributors, such as gender, age, and education level. The class distribution for these datasets is shown in Table 8.

Similar to the domain-specific experiments, for these open-domain datasets we run within- and across culture experiments. Table 9 shows the results of the decep-

Table 8 Class distribution for the Romanian and English-US open-domain deception datasets

Language	Contributors	Male	Female	Truths	Lies	Total
English	512	214	298	3,584	3,584	7,168
Romanian	136	35	101	680	680	1,360

Table 9 Within-culture classification, using LIWC word classes and unigrams

Language	Linguistic (%)	Psychological (%)	Relativity (%)	Personal (%)	All LIWC (%)	Unigrams (%)
English	52.01	52.92	51.92	50.33	56.86	58.33
Romanian	56.76	50.22	52.35	50.66	55.29	57.86

Results are obtained using ten-fold cross validation

Table 10 Cross-cultural experiments using LIWC categories and unigrams

Training: English-US Test: Romanian					
Linguistic	Psychological	Relativity	Personal	All LIWC	Unigrams
56.25 %	51.69 %	51.69 %	50.07 %	56.91 %	59.70 %

tion classification experiments run separately on the English and Romanian datasets, whereas Table 10 shows the results obtained in the cross-cultural experiments.

Not surprisingly, the accuracy of the deception detection method on the open-domain data is below the accuracy obtained on the domain-specific datasets. In addition to the domain-specific/no-domain difference, this drop in accuracy can also be attributed to the fact that the open-domain data consists of short sentences rather than full paragraphs, which could also further explain why using the LIWC derived features does not lead to noticeable improvements over the use of unigrams.

A similar trend is observed in the cross-culture experiments reported in Table 10, where unigrams outperform the use of LIWC classes. It is important to note however, that the use of linguistic classes is still preferable over the use of unigrams, with a rather small accuracy drop of only 2.79 % over the use of costly and more time consuming translations.

To further analyze the nature of the lying process in the open-domain datasets, we obtained the psycholinguistic classes most strongly associated with deception and truth sentences. The results are presented in Table 11. Interestingly, the analysis confirm our findings for the domain-specific experiments, where shared lying patterns among cultures include the use of negation, negative emotions, and references to others. Furthermore, true-tellers related patterns are also shared among cultures, where the most salient classes are family, positive emotions, and positive feeling.

At the same time, we can observe interesting differences among cultures, for instance the use of the words associated with the classes We and Achieve by the Romanian speakers as indicative of truthful responses. Moreover, unlike the American deceivers, Romanian deceivers use Eating, Senses and Body classes more frequently.

Table 11 Top ranked LIWC classes for English and Romanian, along with sample words

Class	Score	Sample words	Class	Score	Sample words
English-US					
Deceptive			Truthful		
Certain,	1.93	Completely, all, never, always	Sleep	0.87	Bed, tires, sleeps, wake, dream, asleep
Negate	1.79	Can't, cannot, not, without, nothing	Incl	0.86	Here, include, into, together, also, too
Anger	1.64	Fight, destruction, poisonous, lied	Posemo	0.84	Richest, enjoyed, fun, better, trust, honest
Down	1.42	Under, off, bottom, lowest, down	Relig	0.65	Church, minister, religion, faith, religious
Motion	1.41	Fly, take, traveled, ran, walk	Posfeel	0.73	Agrees, enjoy(ed), care, love(ed),happy
Money	1.37	Richest, buy, sell, dollars, bank	Music	0.73	Listening, songs, music, sing, song, radio
Friends	1.3	Friend, neighbor,(boy/girl)friend	See	0.74	Vision, see, look(ing), watch, eyes, shows
Otheref	1.35	They, yourself, you, we, someone	Family	0.82	Wife, sister, dad, father, parents, family
Other	1.25	They, he, them, she, himself, him	Tv	0.79	Film, channel, movie, tv, show, television
Romanian					
Deceptive			Truthful		
Negate	2.24	deloc,niciodata,nimic,fara,nu Not at all, nothing, without, not	Motion	0.62	Intregul,alergat,iei,fugit,intr,vizita Entire, running, take, ran, in, visit
Eating	1.91	gateste, mancarea, slabire, mancare Cook, food, weakening, food	Cause	0.66	Cum, judecati, reactii, scopul, deoarece Why, judgments, reactions, order, because
Past	1.85	Zbura, fost, invatat, facut, mintit, luat Flee, former, learned, made, lying, taken	We	0.72	Ne, noi, noastra, noua, noastre, nostru Us, we, our, us, our, our
Money	1.80	Cumparat, bogata, monede, bani Bought,rich,coins,money	Posemo	0.72	Fericita, bun, bucuria, fericirea, frumoasa Blessed, good, joy, happiness, beautiful
Anger	1.70	Nebunie, rau, mintit, urasc Madness, evil, lying, hate	Friends	0.74	Colega, fosta, prietena, iubita, prietenii Colleague, former, friend, girlfriend, friends

Senses	1.69	Apuc, simtit, mancat, simti, mananca Grab, felt, ate, feel, eat	Achieve	0.75	Pierd, prima, inainte, succesul, munca Lose, first, before, success, work
Physical	1.63	Trezesc, cap, degete, gata, picioare Walking, head, fingers, ready, feet	Tentav	0.76	Putea, orice, ori, doar, mult, multi, Can, any, and/or, only, much, many
Certain	1.58	Incredere, intotdeauna, niciodata Confidence, always, never	Home	0.76	Apartamentul, casa, familia, traieste, acasa Apartment, home, family, lives, at home
Body	1.51	Picioare, nascut, degete, limba Feet, born, fingers, language	Posfeel	0.78	Fericita, dragi, romantica, place, zambesti Blessed, dear, romantic, like, smile

5 Conclusions

In this paper, we addressed the task of deception detection within- and across-cultures. Using four datasets from four different cultures each covering three different topics, as well as two additional datasets from two cultures on free topics, we conducted several experiments to evaluate the accuracy of deception detection when learning from data from the same culture or from a different culture. In our evaluations, we compared the use of unigrams versus the use of psycholinguistic word classes.

The main findings from these experiments are: (1) We can build deception classifiers for different cultures with accuracies ranging between 60–70 %, with better performance obtained when using psycholinguistic word classes as compared to simple unigrams; (2) The deception classifiers are not sensitive to different topics, with cross-topic classification experiments leading to results comparable to the within-topic experiments; (3) We can use data originating from one culture to train deception detection classifiers for another culture; the use of psycholinguistic classes as a bridge across languages can be as effective or even more effective than the use of translated unigrams, with the added benefit of making the classification process less costly and less time consuming; (4) Similar findings, although with somehow lower classification results, can be obtained for open-domain short sentence texts in both within- and across-cultures experiments, which confirm the portability of the classification method presented in this paper.

The datasets introduced in this paper are publicly available from http://lit.eecs.umich.edu.

Acknowledgements This material is based in part upon work supported by National Science Foundation awards #1344257 and #1355633 and by DARPA-BAA-12-47 DEFT grant #12475008. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Defense Advanced Research Projects Agency.

References

1. Almela A, Valencia-García R, Cantos P (2012) Seeing through deception: a computational approach to deceit detection in written communication. In: Proceedings of the workshop on computational approaches to deception detection. Association for Computational Linguistics, Avignon, pp 15–22. http://www.aclweb.org/anthology/W12-0403
2. DePaulo B, Lindsay J, Malone B, Muhlenbruck L, Charlton K, Cooper H (2003) Cues to deception. Psychol Bull 129(1):74–118
3. Feng S, Banerjee R, Choi Y (2012) Syntactic stylometry for deception detection. In: Proceedings of the 50th annual meeting of the Association for Computational Linguistics: short papers, ACL '12, vol 2. Association for Computational Linguistics, Stroudsburg, pp 171–175. http://dl.acm.org/citation.cfm?id=2390665.2390708
4. Fofiu A (2012) The romanian version of the liwc2001 dictionary and its application for text analysis with yoshikoder. Studia Universitatis Babes-Bolyai-Sociologia 57(2):139–151

5. Fornaciari T, Poesio M (2013) Automatic deception detection in italian court cases. Artif Intell Law 21(3):303–340. doi:10.1007/s10506-013-9140-4. http://dx.doi.org/10.1007/s10506-013-9140-4
6. Lewis C, George J (2008) Cross-cultural deception in social networking sites and face-to-face communication. Comput Human Behav 24(6):2945–2964. doi:10.1016/j.chb.2008.05.002. http://dx.doi.org/10.1016/j.chb.2008.05.002
7. Lewis C, George J, Giordano G (2009) A cross-cultural comparison of computer-mediated deceptive communication. In: Proceedings of Pacific Asia conference on information systems
8. Mihalcea R, Strapparava C (2009) The lie detector: explorations in the automatic recognition of deceptive language. In: Proceedings of the Association for Computational Linguistics (ACL 2009), Singapore
9. Newman M, Pennebaker J, Berry D, Richards J (2003) Lying words: predicting deception from linguistic styles. Personal Soc Psychol Bull 29:665–675
10. Ott M, Choi Y, Cardie C, Hancock J (2011) Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies - HLT '11, vol 1. Association for Computational Linguistics, Stroudsburg, pp 309–319. URL http://dl.acm.org/citation.cfm?id=2002472.2002512
11. Peng H, Xiaoling C, Na C, Chandramouli R, Subbalakshmi P (2011) Adaptive context modeling for deception detection in emails. In: Proceedings of the 7th international conference on machine learning and data mining in pattern recognition, MLDM'11. Springer, Berlin/Heidelberg, pp 458–468. http://dl.acm.org/citation.cfm?id=2033831.2033870
12. Pennebaker J, Francis M (1999) Linguistic inquiry and word count: LIWC. Erlbaum Publishers, Mahwah
13. Ramírez-Esparza N, Pennebaker JW, García FA, Suriá Martínez R, et al (2007) La psicología del uso de las palabras: un programa de computadora que analiza textos en español (The psychology of word use: a computer program that analyzes texts in Spanish), pp 85–99
14. Rubin V (2010) On deception and deception detection: content analysis of computer-mediated stated beliefs. Proc Am Soc Inf Sci Technol 47(1):1–10. doi:10.1002/meet.14504701124. http://dx.doi.org/10.1002/meet.14504701124
15. Toma C, Hancock J (2010) Reading between the lines: linguistic cues to deception in online dating profiles. In: Proceedings of the 2010 ACM conference on computer supported cooperative work, CSCW '10. ACM, New York, pp 5–8. doi:10.1145/1718918.1718921. http://doi.acm.org/10.1145/1718918.1718921
16. Toma C, Hancock J, Ellison N (2008) Separating fact from fiction: an examination of deceptive self-presentation in online dating profiles. Personal Soc Psychol Bull 34(8):1023–1036. doi:10.1177/0146167208318067. http://psp.sagepub.com/content/34/8/1023.abstract
17. Xu Q, Zhao H (2012) Using deep linguistic features for finding deceptive opinion spam. In: Proceedings of COLING 2012: posters. The COLING 2012. Organizing Committee, Mumbai, pp 1341–1350. http://www.aclweb.org/anthology/C12-2131
18. Zhang H, Wei S, Tan H, Zheng J (2009) Deception detection based on svm for chinese text in cmc. In: Sixth international conference on information technology: new generations, ITNG '09, pp 481–486. doi:10.1109/ITNG.2009.66
19. Zhou L, Shi Y, Zhang, D (2008) A statistical language modeling approach to online deception detection. IEEE Trans Knowl Data Eng 20(8):1077–1081. doi:10.1109/TKDE.2007.190624. http://dx.doi.org/10.1109/TKDE.2007.190624
20. Zhou L, Twitchell D, Qin T, Burgoon J, Nunamaker J (2003) An exploratory study into deception detection in text-based computer-mediated communication. In: Proceedings of the 36th annual Hawaii international conference on system sciences (HICSS'03) - Track1 - HICSS '03, vol 1. IEEE Computer Society, Washington, p 44.2. http://dl.acm.org/citation.cfm?id=820748.821356

Sentiment Analysis: What's Your Opinion?

Jonathan Sonntag and Manfred Stede

Abstract For more than 10 years now, Sentiment Analysis has enjoyed enormous popularity in Computational Linguistics, one main reason being its great potential for practical applications, predominantly (but not only) for industrial purposes. We observe a tendency that early work referred to certain theoretical notions of Subjectivity, whereas a lot of the later approaches follow the 'engineering' perspective that can include using terminology somewhat indiscriminately and are not aiming at making progress with the underlying theoretical issues. In this paper, we first survey some important notions surrounding "Subjectivity" in Linguistics and Psychology, trying to broaden the perspective of standard opinion analysis. Thereafter, we take a snapshot of the state of the art in computational Sentiment Analysis, as it has developed since roughly 2000. Combining these two viewpoints leads us to assessing the gap between the broader notion of Subjectivity Analysis and the subfields that language technology research tends to focus on. We suggest a few potential research directions that could help narrowing this gap.

1 Introduction

Sentiment Analysis has become popular over the last 15 years, due to various reasons: (a) the rise of social media, (b) the technological developments; especially the possibilities and problems of "big data" and, lastly, (c) the progress of natural language processing tools, which lead to a shift of attention towards more complicated and thus more semantic/pragmatic problems, such as Sentiment Analysis, Question Answering, or Textual Entailment.

Mining product reviews is one of the most promising NLP problems for industrial uses today. The grand goal of being able to automatically detect customer feedback in large quantities would help merchandisers and manufacturers in developing specialised marketing campaigns tailored to the standing a product has among its customers, and also contribute to improve the products.

J. Sonntag (✉) • M. Stede
Department Linguistik, Universität Potsdam, Haus 14, Karl-Liebknecht-Straße 24-25, 14476 Potsdam, Germany
e-mail: jonathan.sonntag@yahoo.de; stede@uni-potsdam.de

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_9

Besides the industrial application, Sentiment Analysis is now also attracting attention in the Social Sciences. For example, some researchers investigate the ways of conveying opinion in parliamentary debates; others are interested in automatically gathering standpoints on political issues from newspapers or from social media.

Sentiment Analysis is only one, albeit an important, element of a battery of Text Mining tools necessary to extract the relevant information from large amounts of arbitrary text. Focusing again on the commercial application, the issues are: who buys your products? What other products do your customers buy? How do they get attracted to your products? How do they learn about your products? How often do they use them? Which attributes or *aspects* of the products are evaluated? The role of Sentiment Analysis within the larger Text Mining task in industrial uses is twofold: (a) to detect polar statements, which can be interpreted as relations between entities,[1] and (b) to provide a clear distinction between sentences conveying or revealing a sentiment on the one hand, and objective statements on the other. Traditional Question Answering systems, for example, are designed to extract not opinions and evaluations, but *facts* from a text. Consider the query in Example 1 and the two possible answers, which a QA System might find in a large corpus.

(1) Who was the 15th president of the United States of America?

(2) a. James Buchanan was the 15th, and a horrible, president of the U.S.
    b. James Buchanan was the 15th president of the U.S.

Here, the QA system should prefer the pure 'factual' statement in Example 2b to the subjective one in Example 2a.

The term 'Sentiment Analysis' is used in different ways in the literature. Even attempts at clarifying the term seem to ignore the underlying problem: [2], for instance, suggests that the terms 'Sentiment Analysis' and 'Opinion Mining' can be used interchangeably. Yet, the author bases this assessment on [22, p. 10], who say:

> A sizable number of papers mentioning "Sentiment Analysis" focus on the specific application of classifying reviews as to their polarity, a fact that appears to have caused some authors to suggest that the phrase refers specifically to this narrowly defined task. However, nowadays many construe the term more broadly to mean the computational treatment of opinion, sentiment, and subjectivity in text.

Yet, consumer reviews are not restricted to opinions; they also contain factual, yet polar, statements—see the discussion below in Sect. 4. Therefore, the two terms cannot mean the same.

In this chapter, we use 'Sentiment Analysis' in a broad sense that subsumes opinions, evaluations, emotions, judgements, polar facts, and other kinds of subjective utterances.

[1] We provide just a minimal property of sentiment at this point, which goes beyond coarse-grained Sentiment Analysis, but it is deliberately a rather abstract description. We come back to this issue in Sect. 4.

The rest of this chapter is organised as follows: in Linguistics (and related disciplines), the term 'Sentiment Analysis' is highly uncommon. Instead, 'Subjectivity' is a general notion being studied, inter alia, in Linguistics. In Sect. 2, we describe various facets of this concept and how they relate to each other. Then, Sect. 3 provides a brief survey of work in computational Sentiment Analysis, before in Sect. 4 we give our personal opinion on how certain notions should be defined, and suggest some directions for future work in computational analysis, partly inspired by the insights provided by the theoretical disciplines. Finally, Sect. 5 summarizes the chapter.

2 The Counterpart of *Sentiment* in Linguistics and Psychology

Rather than aiming at a "grand overview", in this section we will take a look at a number of prominent aspects of Subjectivity, as they are being studied in Linguistics and Psychology. The selection is largely motivated by our judgement of relevance to Computational Linguistics. In the second part, we briefly touch upon the relation between Subjectivity and its dual notion: objectivity.

2.1 Subjectivity

> The term Subjectivity refers to the way in which natural languages, in their structure and their normal manner of operation, provide for the locutionary agent's expression of himself and his own attitudes and beliefs. [16]

This well-known definition reminds us of the fact that human language fulfills a variety of purposes, and it also goes some way in suggesting a particular classification of those purposes. An earlier proposal in this vein had been made by Bühler (1934), who assessed that language has three different functions:

- Representation *(Darstellung)*: the speaker describes a state of affairs in the world.
- Expression *(Ausdruck)*: the speaker conveys his or her own feelings or state of mind.
- Appeal *(Appell)*: the speaker wants the addressee to change their mind, or to act in a certain way.

Expression corresponds quite closely to the main point of Lyons's definition, which can be called the "internal" view of Subjectivity; it will be discussed below in Sects. 2.1.1 and 2.1.2. Bühler's *appeal* function, on the other hand, points to an additional role: communication involves multiple partners, and aligning with them is a part of an interlocutor's linguistic behavior. This aspect is nowadays sometimes called "Inter-Subjectivity", and we will address it in Sect. 2.1.3.

2.1.1 The 'Private State'

If the "objective" is in principle observable to everybody, then a reasonable reading of "subjective" is that of a particular agent's "inner world", which is not observable to anyone except for that agent him- or herself. Quirk et al. [25] used the term *private state* for this and illustrate the idea with this example (p. 1181; emphasis by those authors): "A person may be observed to *assert that God exists*, but not to *believe that God exists*. Belief is in this sense 'private'." Then, agents may choose to communicate (certain aspects of) their private states, and at this point, the linguistic interest sets in: What are the linguistic means for verbalizing different aspects of a private state? In order to study this, the fairly general concept of 'private state' needs to be broken up into a number of distinct, simpler realms.

One proposal to this end can be found in 'Appraisal Theory' stemming from systemic-functional linguistics (SFL), and proposed by Martin and White [18]. The overall goal of the SFL approach to language is to delineate and taxonomise the semantic and pragmatic dimensions that are assumed to be responsible for the spectrum of syntactic variety within a language. The portion of this endeavor that is relevant for our purposes is the following sub-taxonomy:

- ATTITUDE encompasses different options for expressing positive or negative evaluation
    - AFFECT: emotional evaluation of things, processes or states of affairs; main subclasses: un/happiness, in/security, dis/satisfaction
    - JUDGEMENT: ethical evaluation of human behavior (e.g., good/bad)
    - APPRECIATION: aesthetic or functional evaluation of things, processes and states of affairs (e.g., beautiful/ugly, useful/useless)
- ENGAGEMENT: addresses options for expanding and contracting space for other voices (i.e. how much does the writer endorse the statements of others)
- GRADUATION: adjustments of attitude and engagement in terms of strength

For our purposes, the central part of the taxonomy is ATTITUDE with its three daughter nodes, which all revolve around a speaker *evaluating* something. The first way of doing this is by expressing an emotion or affect; since this has been studied extensively in Psychology, we discuss it in somewhat more detail in Sect. 2.1.2. For the non-emotional evaluations, Martin and White distinguish two classes of targets of the evaluation: ethical judgment of human behavior versus aesthetic/functional evaluation of "things" in a wide sense. It is this particular notion that is at the heart of the vast majority of computational work on opinion mining.

The third term in the Martin/White sub-taxonomy is GRADUATION, which refers to the linguistic means of marking a strengthening or down-toning of a subjective utterance. Which of these means are appropriate depends on the particular type of utterance. Some distinctions to be made among so-called *epistemic stances* are:

- marking the degree of precision or truth or appropriateness of a category label
  She is almost a Ph.D. now.

- marking the probability of truth
  Most likely she is a Ph.D. by now.
- marking an expectation on the probability of a statement
  She ought to be a Ph.D. soon.

2.1.2 Emotions and Their Reflection in Language

Emotions are quite clearly distinct from non-emotions,[2] and the automatic identification and displaying of emotions has become a research discipline in its own right, also with close ties to Computational Linguistics (CL) [26].

A major foundation of emotion research in Psychology is the work of [19]. Their theory has become influential also in CL, under the term "OCC Model of Emotions" [31]. Here, the realm of emotion is split into two different concepts: *arousal* and *appraisal*.[3] Arousal is described as the "hot" part of emotion and has a similar meaning to the term "stress" in everyday language. It involves all biochemical reactions to stimuli. For Computational Linguistics, the second term is more interesting. Appraisal is the cold, calculating side of emotions; it can be characterised as reactions to three types of entities: objects, agents, and events. (i) Objects are being evaluated in terms of their *appealingness*: How well does the object fit to a person's attitudes? If one's favourite colour is 'green', then a green bicycle is perceived more favourably than a pink bicycle. (ii) Reactions to the actions of another agent are evaluated in terms of *praiseworthiness*, which refers to normative expectations on how an agent should act in a certain situation. (iii) Finally, events are evaluated with respect to their *desirability*, i.e. they lead to positive emotions if they support the agents goals (Fig. 1).

All reactions that are relevant to appraisal share one basic feature: they are *valenced reactions*—we respond either positively or negatively to the stimulus. We discuss this commonality and possible implications for Sentiment Analysis in Sect. 4.

In addition to the different types of appraisal, Ortony et al. describe *intensity* as a major factor: the strength with which a person experiences the emotion. Intensity variables can be either global, i.e., relevant for every appraisal type, or local, i.e., dependent on the appraisal type at hand. An interesting variable affecting the intensity is the *sense of reality*. The authors hypothesise that emotion-inducing situations have a certain temporal, locational and psychological proximity: distant situations induce the same emotions as near situations, but the intensity changes. Another factor worth mentioning is *unexpectedness*: situations that have a high probability of happening are deemed less intense than surprising ones.

[2] Ortony et al. [19, pp. 29–32] give examples for non-emotional events giving rise to emotions. Yet, the distinction is clear.

[3] Notice the identity to the linguistic term discussed in the previous section (Martin/White). There seems to be no direct (established) connection between the two approaches.

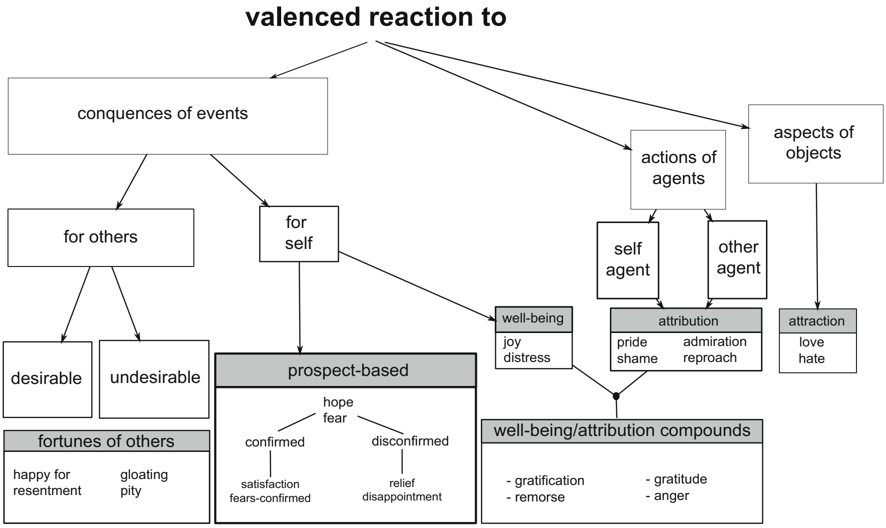


Fig. 1 Emotion schema described in [19, p. 19]

Another interesting proposal by Ortony et al. is the *balance principle* for valenced reactions. It originates from [10] and describes how relations between people evolve when they form a triangle, i.e., when three people interact with each other (*triadic relationships*). Balance is achieved if and only if the product of the edge weights is positive. Applied to text analysis, this means: if all valenced reactions are combined into a weighted graph in which the nodes represent discourse entities and the edges the intensity of the reactions between them (say, -1 to 0 for negative reactions, and 0 to 1 for positive reactions), then for each triangle in the graph, the product of their edge weights tends to be positive.

Contrary to the desire of Computational Linguists to have a clearly defined inventory of emotions, Ortony et al. refrain from defining such an inventory, due to the disagreement on the term *basic emotions* in Psychology. The only possibly 'minimal' emotions those authors would agree with are positive and negative emotions, as already proposed by [32]. Another, rather pessimistic claim for the engineering task is the authors' diagnosis that the words of English are underspecified with respect to their emotion type. This is an issue that has been studied under the heading of lexical *connotation*: the idea is that words have a 'kernel meaning', essentially the real-world entity that they stand for (denotation), plus additional traits of meaning, the so-called connotations. The challenge is to produce a list of connotative dimensions that can be productively used to differentiate between words that have the same denotation. The best-known such dimension is *formality* (e.g., motion picture vs. movie); others include *pejorative* (e.g., man, jerk) and *euphemism* (e.g., genocide, ethnic cleansing).

2.1.3 Intersubjectivity

In some situations, a speaker may convey a private state just for herself, as by uttering "ouch!" or "phew!". Typically, however, communication is directed toward some addressee, which in our context leads to the notion of "Intersubjectivity". One aspect of this notion, which is particularly relevant for Sentiment Analysis, is the question to what extent a speaker assumes responsibility for her statement: am I stating my own conviction, or am I attributing the responsibility to somebody else? Quoted speech is the clearest case here: its boundaries are unambiguously marked. For indirect speech, this need not be the case. If it stretches over more than one sentence, it can be ambiguous whether some material is still attributed to a source cited earlier, or whether the speaker has resumed to stating his own position. The linguistic notion at stake here is *evidentiality*, which refers to the variety of means that languages offer for marking this relationship between statement and alleged responsibility. For an overview on the linguistic discussion, see [7]. For some languages, this marking is obligatory by means of grammatical categories; but for English or German, speakers have the choice of choosing lexical expressions to mark Evidentiality, in particular via modal verbs:

(3) Es wird morgen regnen.
It will tomorrow rain.
'It will be raining tomorrow.'

a. Es soll morgen regnen.
It should tomorrow rain.
'It is said to be raining tomorrow.'

There is thus a continuum between explicitly stating the source of a statement and clearly marking the boundaries of that statement on the one hand, and vaguely hinting at "some" external source. In fact, one special case of an "external" source or viewpoint can be the speaker's own perception, as in:

(4) It seems to be raining.

This can be paraphrased as "My sensory organs indicate that it is raining, but I don't fully commit to the truth of the statement." The speaker thus puts some distance between himself and the statement, and we can see that there is a fuzzy boundary between the realms of evidentiality and what we have discussed above as 'epistemic stance', in particular the marking of reliability of information.

The term 'Intersubjectivity' has many more facets, and here we want to just briefly mention the work of cognitively-oriented linguists such as Langacker [17] or Verhagen [38]. In contrast to the more standard linguistic analysis "pipeline" (syntax followed by semantics followed by pragmatics/context), they emphasize that basically any linguistic utterance should be seen *foremost* as being directed to an addressee and as managing the relation between the interlocutors. In Langacker's theory, an utterance has to be analyzed in tandem in terms of the interlocutor relationship and the real-word states of affairs that is being talked about. In a similar vein, Verhagen elaborates on the idea of Anscombe and Ducrot [1], who posit that language use is essentially always ‘argumentative’ in the sense that a speaker by making an utterance intends to influence the mental state of the addressee. These (and other) authors demonstrate with many examples that linguistic constructions are sensitive to the ‘argumentative orientation’ of individual statements and, hence, that Subjectivity is deeply built into the linguistic system.

2.2 Factuality

We use the term *factuality* for the linguistic marking whether a certain event happened or an object exists. Factuality is relevant to Sentiment Analysis because it can contribute to the decision whether a sentence is understood as conveying a sentiment.

2.2.1 The Semantic Viewpoint: Evidentiality and Veridicity

Linguistic semantics is interested in modality in general, and the marking of *evidentiality* is an important subgroup here (cf. Sect. 2.1.3). Coming from the CL perspective, [12] use the term *Vericidity*, which is to deal with these questions on a certain event:

1. has the event really occured?
2. who said that the event occured?
3. does the author believe the event occured?
4. how does the author of the text refer to it?

The first question is the central, practical question. Questions 2 to 4 are part of the first question and are observable in text, while the answer to the first one is not.

Sentences 5a and 5b demonstrate how different reporting verbs can convey different stances of the author. While the author does not take any stance in Sentence 5a, he supports Bush’s claim in Sentence 5b.

(5) a. Bush *said* that Iraq had aided al Qaida.
    b. Bush *acknowledged* that Iraq had aided al Qaida.

Karttunen et al. [12] embed their research within the Advanced Question Answering for Intelligence (AQUAINT) project. In Sect. 1 we pointed out why knowledge about subjective versus objective statements is important for QA systems.

2.2.2 Interpretation

Before a Sentiment evolves in a human being as a reaction to some real world event, the event has to be interpreted. This interpretation can be straightforward or involve some further inference. Example 6 typically creates some negativity inside a reader: a reasonable interpretation is that 'Carlo' wants[4] to do bad things to the people inside the cafe. Then, a complex interaction evolves: the reader probably develops some sympathy for the people inside the cafe, since, for all he knows, they were ordinary people just like the reader himself. Additionally, Carlo obviously wants to harm the people inside the cafe. Then, the reader reacts to the negativity of Carlo towards people like himself and the last sentiment relation emerges: the reader dislikes Carlo. This interpretation also coincides with the balance principle described above in Sect. 2.1.2.

(6) Carlo threw a hand-granade inside the cafe.

This example shows that facts are an integral part of emotion detection and thus also of Sentiment Analysis. Not every interpretation of the real world is as straightforward as the example above, though. As readers of news in political discourse we rely on the activity of the media providing interpretations for us.

In a newspaper article, Zastrow [43] criticises the role of the media in political discourse. Analyzing the media coverage of the 2013 elections in the German state of Lower Saxony, he describes how the analysis and reflection in the media changes significantly due to small changes in the numerical results of the elections. Commenting on the interpretation of the results of the election, he says:

> And, thus, just about everything that has been said was turned into the opposite a little later.

Further debating the interpretations of the media, Zastrow wonders:

> What is going on there? Nothing special, it is simply the good old manipulation. The analyses only pretend to be analyses. In fact, they are political demands masked as objective analysis.

Finally, the author makes a very strong point about the usage of facts in argumentation:

> There is no bigger success in a political debate than to convince a majority that your opinion, evaluations or demands are facts.

These quotations not only deal with the interpretation of events but also with their veridicity. A possible reason is that assessing the veridicity of facts is part of constructing the mental representation of real-world affairs.

3 Sentiment Analysis in Computational Linguistics

We now turn to describing the major developments of Sentiment Analysis within the "engineering" part of Computational Linguistics. For the reader interested in a more extensive introduction, Pang et al. [22] provide an overview over early work

[4] Note that the use of the term 'want' suggests Subjectivity.

in the field, with some tools having been developed as early as 1979 (the POLITICS software for Sentiment Analysis on political text [5]).

Much work in early Sentiment Analysis focuses on the assignment of polarity values on the level of words. We touch upon this in Sect. 3.1, in which we briefly discuss different lexicons and corpora, which form the basis for any sentiment system. Afterwards, Sect. 3.2 discusses rule-based analysis systems, and then Sects. 3.3 and 3.4 talk about *aspect analysis* and machine learning approaches, respectively. The two are closely related because most fine-grained sentiment approaches relying on machine learning are aspect analysis systems.

3.1 Resources: Lexicons and Corpora

For all approaches to Sentiment Analysis, annotated corpora are required. Their minimal usage is the evaluation of automatic systems, but of course, for machine-learning approaches, corpora are also essential as training data.

Of the various corpora that have been built and annotated with sentiment, we mention only a few. The MPQA corpus [39] is one of the most prominent for English, while the MLSA corpus [6] is the first publicly available resource for German.

The MPQA corpus consists of Chinese newspaper articles translated into English and articles from U.S. newspapers. They have been annotated for ‘subjective frames’, which are based on the notion of ‘private state’ as introduced in Sect. 2.1.1. Very briefly, a frame consists of the opinion holder, opinion target, and the expression of the sentiment. The original corpus consists of 535 documents, corresponding to 11.114 sentences.

The MLSA corpus is a fine-grained corpus based on the DeWaC Corpus [3]. It is annotated with three different layers: (a) at the sentence-level, objective/subjective[5] and positive/neutral/negative are specified, (b) polarity and modifiers are annotated at the phrase-level and (c) private states are annotated at the expression-level similar to the annotation of the MPQA corpus.

A recent corpus of amazon.com reviews, the USAGE corpus for aspect analysis [15], consists of 800 German and 800 English reviews and is annotated for aspects and evaluative expressions.

Apart from the three corpora mentioned above, various others were assembled from reviews. Since many review sites provide textual comments as well as a numerical or star-based rating, information on sentiment in the text can be inferred straightforwardly assuming minimal simplifications [20, 21, 23].

One type of information that is required for virtually every solution to Sentiment Analysis is the so-called *prior polarity* of words. Simply put, *bad* is inherently

[5] In this case, the authors intention is to specify factuality, which relates to ‘Evidentiality’ and ‘Veridicity’ (cf. Sect. 2.2.1).

negative, while *joy* is inherently positive. Due to its importance, it makes sense to factor polarity information out of the individual approach and represent it externally in a lexicon—so that the result of the expensive production of such a lexicon can be re-used and shared with other researchers. Even machine-learning approaches to Sentiment Analysis typically use a lexicon containing information about the polarity of words. Often the polarity is supplied not as just a binary distinction but in terms of intensity values (either numerical or nominal).

While sentiment lexicons can be written by hand, they are usually extracted from large corpora and possibly hand-corrected afterwards. In general, the smaller the piece of text a human annotator has to judge is, the more difficulties they have in their judgement. Marking single words in isolation as being positive or negative, and maybe even providing a score, is therefore very difficult and hardly leads to high precision. (Our illustrative examples of *bad* and *joy* are comparatively easy cases.)

One of the first approaches to generate such a lexicon automatically is [9]. Adjectives are differentiated into positive and negative ones in order to detect antonyms and to distinguish near synonyms in text. The authors' algorithm is based upon the intuition that words with the same semantic orientations occur in coordinated constructions while words with different semantic orientations do not. For example (taken from [9]):

(7) a. fair and legitimate
    b. corrupt and brutal

(8) a. # fair and brutal
    b. # corrupt and legitimate

After the detection of conjunctions, a regression model is used to establish relationships between conjunctions with respect to their semantic orientation and the result is a graph with the words being the nodes and the edges representing whether the two connected nodes are of the same or different semantic orientation. Finally, the graph is clustered into two sets: one being the class of positive adjectives and the other one being the class of negative adjectives.

More recent methods of lexicon generation typically use seed words (just to name some pioneer work: [8, 36]). The idea behind seeding methods is the following: at first, a rather small, hand-crafted set of reliable instances of negative and positive semantically oriented words are built. In a second step, a similarity measure is established, and words that are sufficiently similar to the seed words are added to the respective sets of positive and negative words.

An import sentiment lexicon of English is [42]. An excerpt from that dictionary is presented in Table 1. The first major lexicon for German is [27] which contains entries consisting of the lemma, the PoS-Tag using the Stuttgart-Tübingen Tagset [35], a weight and inflected forms. The weights are machine generated from various sources. A first step was to machine translate entries from the General Inquirer [30] into German and review them manually to remove bad entries. To extend these entries, a co-occurrence analysis is performed on a corpus of product reviews. The

Table 1 Example entries from a Subjectivity Lexicon [42]

Word	Prior-polarity	Reliability
Abhorrent	Negative	Strong
Absence	Negative	Weak
Obsolete	Negative	Weak
Bankrupt	Negative	Weak
Lack	Negative	Strong
Obstacle	Negative	Strong
Odd	Negative	Weak
Opportunity	Positive	Strong
Originality	Weak	Strong

Notice that some entries seem to be rather objective but polar, and that some entries seem to be very underspecified with respect to their polarity and strength. The authors explicitly mention that the entries in their lexicon *can* have subjective meanings

machine-translated entries are added to the product reviews and high co-occurrence words are extracted and, again, manually inspected and selected to be added to the lexicon. Finally, the German Collocation Dictionary [24] is used to extract polar noun clusters. The German Collocation Dictionary groups words by their semantic similarity and the groups with a strong relation to sentiment are calculated and added to the lexicon. Semantic orientation and the strength is then calculated using Pointwise Mutual Information.

3.2 Rule-Based Approaches

Rule-based, or symbolic, approaches to Sentiment Analysis have the advantage to work fairly reliable; besides, it is easier to repair unintended behaviour of symbolic systems than to fix models for statistical classifications. Furthermore, especially companies are interested in tracking the continuous improvement of their systems over time, and it is easier to achieve consistent improvement of a system if it is rule-based, since additional rules for false negatives can be added to the system. Of course, machine-learning systems can also be improved, by way of providing more training data. However, increasing the size of the training set does not automatically lead to a better performance. Instead, the performance can reach a plateau or even drop, and it is unknown how much more training data is required to leave the plateau. Also, when the desire is to fix a particular problem or class of problems, it can be very difficult to obtain precisely the “right” training data for it.

This is important because companies need to respond to customer feedback. Clients may complain about missing or wrong sentiment relations, and then it is important to work on those cases in particular. Those corrections might not make the system much better, nor might there be any change in f-score, but the customer satisfaction may be more important than that.

On the other hand, the central disadvantage of rule-based systems is that the number of hits per rule is usually pretty low and follows a Zipfian distribution. A large percentage of the rules may deal with hapax legomena. And the rules which hit very frequently can easily be too general and produce many mistakes.

A prominent example for a symbolic system is the Semantic Orientation Calculator (SO-CAL) [34]. It is based on prior polarities of words and on rules for combining them to an aggregate sentence polarity, which account for the effects of specific contexts involving irrealis blocking, negations, diminishers and intensifiers.

(9) I do [not]$_{\sim 4}$ [like]$_{+1}$ this dishwasher, although the dishes are [really]$_{\text{intensifier:1.15}}$ [clean]$_{+2}$ afterwards.

Sentence 9 contains an intensifier and a negation. SO-CAL models intensification by multiplication, and 'really' has a value of 1.15. All polar words in proximity of the intensifier are modified accordingly and thus, clean is increased from 2 to 2.3. For negations, SO-CAL estimates the negation scope by looking forward as well as backwards up to a potential clause boundary. The effect of a negation is a polarity shift: The value of 'like' is shifted by 4 from 1 to −3.

SO-CAL was evaluated on reviews from epinions.com, movie reviews [21], and camera reviews. The average accuracy is reported as 0.7874. The accuracy is quite stable across the tested corpora. It also relatively robust against change of domains, where accuracies between 0.7938 and 0.8898 are given.

3.3 Aspect Analysis

Aspect analysis is a compromise between text-level analysis, which is more suitable for machine learning algorithms, and phrase-level analysis, which is a requirement for accurate Sentiment Analysis, since leaving out the detection of opinion targets generates many mistakes. Typically, aspect analysis is employed in the analysis of product reviews.

(10) a. This is the quietest dishwasher I have ever owned.
b. And yes, it's so quiet that you can't tell it's running [. . .]
c. Another great surprise was to see how clean our glassware and dishes come out.

Sentences 10a and 10b both refer to the same aspect of a dishwasher: its loudness. Sentence 10c, on the other hand, evaluates a different aspect.[6] Thus, aspect analysis consists of (a) knowing what possible aspects of a product are, (b) detecting aspects

[6] All Sentences are taken from http://www.amazon.com/Bosch-SHP65T55UC-Stainless-Integrated-Dishwasher/dp/B00CWX0KDA/ref=sr_1_2?ie=UTF8&qid=1401714426&sr=8-2&keywords=dishwasher.

in text and mapping them to their *aspect category* and (c) deciding about the polarity, intensity and possible other attributes of the sentiment.

Hu et al. [11] describe the task of aspect analysis as a special instance of text summarisation. A set of product reviews has to be summarised in order for potential buyers to get a quick overview over all the reviews. Such summarisation is advantageous, because popular items on large online stores can have thousands of reviews. (As of June, 2014, the most reviewed book on a large online store has $17,500$ reviews). The authors define two stages in their approach to aspect analysis. The first is to extract the product features (or *aspects*) that the reviews comment upon. The second step detects the polarity of the statements within the sentences that talk about an aspect. This latter step is the same as in sentence- or phrase-level Sentiment Analysis, so we ignore it here. In step 1, Hu et al. make a distinction between frequent and infrequent aspects. This distinction is only based on the differences in finding the aspects and not in a different role within Sentiment Analysis. Frequent aspects have to occur in at least 1 % of all sentences from the reviews of a product. All other aspects are treated as infrequent ones. A rough outline of the algorithm:

1. detect frequent aspects
2. prune frequent aspects to reduce the noise generated within the detection of frequent aspects, and to remove redundancy stemming from more or less coarse-grained features (e.g., ‘battery’ and ‘battery life’)
3. create a list of opinionated words from the contexts of the previous step: a modifying adjective close to a frequent aspect is an opinionated word
4. detect infrequent aspects based on the opinion words gathered in the last step: the nearest noun phrase is an aspect

In this approach, aspects and opinion words co-depend; if the aspects are known, it is easier to compute the opinion, and vice versa.

Klinger et al. [14] directly investigate this dependency between the evaluative expressions and aspects using factor graphs. The major finding is that the knowledge about aspects significantly improves the detection of evaluative expressions: 0.54 f_1-score for the detection of evaluative expressions in isolation increases to 0.65 f_1-score for its detection with gold-knowledge of aspects. The independent detection of targets is reported with an f_1-score of 0.32 and rises to 0.58 with gold-knowledge of evaluative expressions. The f_1-score for partial overlap is higher, but the tendency remains.

3.4 Machine Learning Approaches

The majority of systems for detecting and classifying sentiment uses machine learning (henceforth ML) approaches. These are ideal for very complex problems that are hard to describe or even to understand for the human analyst. And since no comprehensive and detailed theory for sentiment in natural language exists, it is an obvious candidate to be tackled with ML approaches.

Wiegand [40] reports an accuracy of 0.775 for sentence-level polarity classification with an optimal feature set consisting of prior-polarity information, bag of words and a range of linguistic features. Unfortunately, the author only reports results for experiments on the MPQA corpus and it is thus unclear how the results carry over to different corpora. The author relates his work to Pang et al. [23] and compares the bag-of-word feature classifications to his own work. Pang et al. achieve an accuracy of 0.829 while Wiegand reports an accuracy of 0.672. The difference stems from the granularity of the analysis: Pang et al. work at document-level. Another (not mentioned) difference are the used domains and genres: newspaper articles on the one hand, and movie reviews on the other hand. Still, the comparison provides evidence for an intuitively obvious observation: it is easier to classify sentiment at document-level than at sentence-level.

Recently, Sentiment Analysis was set as a task in the SemEval-2013 and 2014 challenges. The task provides micro-blogging data annotated at message level with a four-way classification: 'objective', 'neutral', 'positive', and 'negative'. The results that were achieved range from 0.1628 to 0.6902 in 2013 and from 0.396 to 0.7484 for this task in 2014.

As indicated above, fine-grained Sentiment Analysis is harder than coarse-grained analysis. But, fine-grained analysis is also the more interesting and challenging problem and has become increasingly popular. Unfortunately, fine-grained sentiment analysis cannot easily be formulated as a set of classification problems. Therefore, fine-grained ML approaches can either try to model compositional sentiment relying on syntactic or semantic representations [29], or do aspect analysis (cf. Sect. 3.3).

Socher et al. [29] introduce a sentiment treebank ($11,855$ sentences from movie reviews) that contains syntactic analyses where each constituent is assigned a polarity: very positive , positive, neutral, negative or very negative. The authors also describe a classification system trained on this corpus using neural networks and semantic vector spaces. The sentiment of a phrase is computed by applying a compositionality function to each pair of sister nodes in a binary tree. Semantic vector representations of the words are used to learn and compute prior-polarities for the word or phrase.

The current interest in aspect analysis led to another SemEval task in 2014 on sentiment detection in customer reviews of restaurants and laptops. The winners, Kiritchenko et al. [13] cast the problem of aspect term extraction as a tagging task: every token in a sentence is tagged as either belonging to an aspect term or not. The second sub-task provides gold-standard aspect terms within sentences, and the polarity of the sentence towards the aspect is to be determined. They describe a *support vector machine* (SVM) using surface, lexicon, and parse features. For all three classes, they define features that are essentially uni- and bigrams anchored at the aspect term. The performance varies significantly between the restaurant and the laptop data-sets. 0.7049 accuracy is reported for the laptop reviews, and 0.8016 for the restaurant reviews.

4 What Is Your Opinion, What Is Ours?

After relatively objective surveys of the linguistic notion of Subjectivity and the field of Sentiment Analysis, we now turn to a relatively subjective synthesis. At first, we offer a set of definitions to clear up the terminology; then, we collect a number of questions that arise from the previous two sections and suggest some personal answers.

4.1 Terminology

By making a *factual* statement, the speaker asserts something about the real world that she regards as (in principle) verifiable by others. This is in contrast to *subjective* utterances. We distinguish the conveying of private states (subjective$_1$) from phenomena of Intersubjectivity (perspective-taking etc.; subjective$_2$). In the remainder of this section, we will be concerned only with subjectivity$_1$. Language offers means of signalling the difference between factual and subjective$_1$, but speakers are not obliged to make it explicit; hence there is often room for interpretation by the hearer.

We regard *evaluations* as utterances that are often difficult to classify as either subjective or objective. They obligatorily mention a target, i.e., the entity being evaluated, and they seem to neutrally state (for example) an attribute of the target, usually situating it on some scale. The evaluation can have an underlying polarity (Examples 11a–11d) but it need not (11e).

(11) a. The weather is nice.
b. The food is salty.
c. The dishwasher is quiet.
d. The dishes come out spotless.
e. This lecture hall is huge.

The speaker may introduce such a statement with *I think*, and accordingly, an addressee may dispute such a statement, e.g., by responding *Well, not quite*. This indicates that these cases are not as objective as *snow is white*, but at the same time they are no prototypical cases of private states: The two genuinely-subjective$_1$ types (emotions and opinions) cannot be disputed by an addressee. Emotional statements may have a target (12a) or not (12b), whereas opinions always have one (13a, 13b).

(12) a. I'm afraid of spiders.
b. I'm feeling great today.

(13) a. I like this kind of wine.
b. This has always been my favourite restaurant.

Notice that there is no point in the addressee replying *Not quite* or *That's not true* to any of the above utterances.

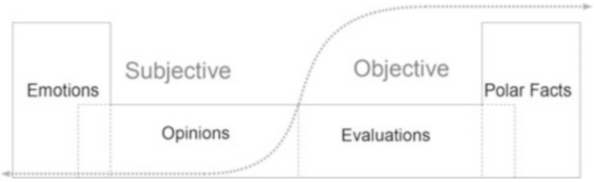


Fig. 2 Mapping the Sentiment terminology. *Dotted lines* indicate the room for interpretation by recipients of language. *Solid lines* represent the underlying model which we assume. The *sloped line* thus indicates how likely a recipient is to interpret an utterance supposed to convey an opinion, evaluation, etc. as subjective or objective, and even as an opinion, evaluation, etc

We thus see the evaluations as being situated on a middle ground between subjective and objective, but as leaning toward the objective: They are open to verification by others, and they will often be agreeable to a majority of the audience. The MacMillan Dictionary [28] defines evaluating as:

> to think carefully about something before making a judgment about its value, importance, or quality

Finally, we view *polarity* as a basic emotional category by which humans respond to experiences: positive or negative. Conveying an emotion usually includes polarity, but there is an ambiguous or non-polar range (e.g., *I am excited*). As stated above, opinions are always polar, whereas evaluations need not be. Utilizing lexical connotations is one way for a speaker to convey a polar evaluation: *The gentleman/man/jerk asked me a question*. Polarity, however, is not generally tied to Subjectivity. So-called *polar facts* [37,41] convey positive or negative consequences for some agent (as generally assumed, i.e. being the common knowledge), without evaluation or opinion being part of it:

(14) a. Joe Smith was murdered.
b. George Myers received the nobel prize.

Notice that adding *I think* does not have the same effect as with evaluations (with polar facts, it merely conveys degree of belief, not personal judgement).

Figure 2 summarizes the distinctions we have proposed.

4.2 Issues (1): Polarity and Lexicons

How Should Subjectivity and Polarity be Handled in an Ideal Sentiment Lexicon?

An inspection of various prior-polarity lexicons reveals a large number of entries which are not subjective according to our terminology landscape as introduced above, but very relevant for the analysis of product reviews or political discussions. Examples are shown in Table 2.

Table 2 Comparing selected entries from the Subjectivity Lexicon [42] and the lexicon developed for SO-CAL [34]

Word	Subjectivity Lexicon	SO-CAL
Eliminate	w	−4
Assassinate	w	−2
Veto	s	n.a.
Erosion	w	−2

‘s’ = ‘strongly subjective’; ‘w’ = ‘weakly subjective’

The examples indicate that Subjectivity as a lexical feature is very difficult to agree on, and therefore we would suggest to eliminate it from (or at least, significantly reduce its role in) lexical description. Instead, we would clearly focus on polarity, which is a central ingredient of all the three of our categories subjective opinion, semi-objective evaluation, and polar fact.

Evaluations, as in product reviews, behave much like facts—recall examples 11a–11e. They do not include linguistic markers of Subjectivity/opinion, so the question is how human readers understand that, e.g., examples 11c and 11d are positive evaluations of a dishwasher? Arguing in the terms of [19], we can say that the speaker’s expectations towards the dishwasher are exceeded, which in turn creates positive emotions. Then, something positive is stated about an object, and it is not an opinion.

If we reduce sentiment lexicon construction to assigning positive/negative values, the process becomes much easier, and the lexicon can then be used for a range of different tasks where “standard” opinion or domain-specific evaluations, or polar facts can play a role. These facts would not have to be encoded in separate lists anymore but simply were part of a prior-polarity lexicon. The rather artificial distinction between entries for a polar fact lexicon and an emotion/Subjectivity Lexicon vanishes.

However, using such a lexicon for a broad range of tasks makes it necessary to pay more attention to context and to calculate posterior polarity in appropriate ways; that is the issue we address next.

4.3 Issues (2): Context

Can Sentiment Analysis Benefit From Considering Coarse-Grained Text Structure?

For the task of text-level Sentiment Analysis, it can help to take the genre-specific text structure into account. In our work with movie reviews, we tested this by implementing a prior step of ‘zone identification’: Movie reviews usually provide information about what happens in the plot (description), and they present the author’s opinion on various aspects (comment). We found that description and comment are most often clearly separate in the paragraphs of a review. Using a classifier making this distinction and then restricting the text given to the sentiment analyzer to the comment paragraphs yields to improvements ranging from 2 to 12 % depending on the quality of the prior describe/comment classification [33].

What About Argumentation and Its Structure?

By extension to the previous point, when sentences are composed of multiple clauses, the argumentative orientation can be modulated by connectives like *but* or *although*, which also renders the "single-number" Sentiment Analysis as a great simplification.

(15) a. Big Brother is back these days, but in the meantime, the country has invested itself so deeply in its fantasy of cyber-liberation that no outrage will be sufficient to move it.[7]

Many Systems Compute Sentiment Score at Sentence-Level. Is That Adequate?

The basic idea that is regularly implemented is to see sentence-level sentiment as the average of the lexical polarities in the sentence. This is a simple rule that often works, but it cannot do justice to sentences like 16a to 16c, where complex opinions are being stated that cannot just be "averaged". Likewise, the interesting recent approach of Socher et al. [29], which propagates sentiment values from node to node in the syntactic tree (see Sect. 3.4), does not capture the sentiment, because still, "overall sentiment" of the sentence is being reduced to a single number.

(16) a. The Germans are partners and adversaries at the same time.[8]

b. To snub and even to wound your most zealous supporters, as Obama has done, is regarded as a mark of maturity in Washington.[9]

c. Die Schweiz will die Zuwanderung von EU-Bürgern
the Switzerland wants the immigration of EU-citizens
beschränken—und Europa ist empört.
restrict—and Europe is outraged.

Switzerland wants to restrict the immigration of EU-citizens—and Europe is outraged.

[7] From http://www.faz.net/aktuell/feuilleton/debatten/the-u-s-and-the-n-s-a-scandal-freedom-the-big-american-lie-12263704.html?printPagedArticle=true.

[8] From http://www.spiegel.de/international/germany/why-spiegel-is-posting-leaked-nsa-documents-about-germany-a-975431.html.

[9] From http://www.faz.net/aktuell/feuilleton/debatten/the-u-s-and-the-n-s-a-scandal-freedom-the-big-american-lie-12263704.html.

Example 16c[10] is only interpretable in terms of sentiment if the relations between the entities involved in the sentence are examined. Entities are “Switzerland”, “the immigration of EU-citizens”, “EU-citizens”, “EU”, “Europe” and lastly, the author. Those relations can then be investigated and classified into being negative, positive, neutral in one dimension and being an opinion, evaluation, fact or emotion in another dimension.

Just as aspect analysis re-interprets the task of text-level Sentiment Analysis, compositional Sentiment Analysis needs to be re-interpreted as well: sources and targets should be a central part of Sentiment Analysis. It is neither feasible to assign the sentence-level sentiment to all entities within the sentence, nor to the topic of the text or sentence. Instead, what is missing is a systematic detection of sources and targets.

How Can We Model Sentiment Interaction When Multiple Entities Are Affected?

When multiple entities are involved, interesting opportunities for sentiment classification arise. Let us assume that each sentiment is a relation between a source S and a target T. In longer texts, we will probably encounter different relations with a common source and target. Our intuition is that all relations between the same source and target tend to have the same polarity. This principle could be used as an optimisation principle and therefore help out in dubious relations. If two entities share relation $R_1 \dots R_n$ and R_3 to R_n are clearly positive and, additionally, the classification system is insecure about assigning a negative polarity to R_1, then the principle can influence the decision and R_1 gets a “neutral” label or none at all. If the author of the sentence wants to convey a different sentiment from the source towards the target, he can do it but he has to do it very explicitly. For example by using discourse markers as discussed above on argumentation structure.

(17) a. When Italian Prime Minister Matteo Renzi now offers the prospect of support for Juncker, what we are seeing is really part of a larger offensive against Berlin’s so-called austerity diktat.

Extending this principle to multiple entities is possible via the Balance Principle, which we introduced in Sect. 2.1.2: All relations connecting three entities into a fully connected (sub-)graph follow the tendency that the product of their polarities is positive. To illustrate this, we consider the entities “Renzi”, “Juncker” and “Berlin’s so-call austerity diktat”. An explicitly positive relation can be established between “Renzi” and “Juncker” because of “offers the prospect of support”. And the relations from “Renzi” and “Juncker”, respectively, to “Berlin’s so-called austerity diktat” are both negative. Thus, this triadic relationship fullfills the balance principle. If a classifier for the polarity of the relations is unsure about one of its three decisions, it can use the principle to gain additional evidence.

[10] From http://www.sueddeutsche.de/politik/steuergeheimnis-und-zuzug-stopp-warum-die-schweiz-europas-liebster-pruegelknabe-ist-1.1659263.

5 Summary

In this chapter, we provided an (arguably selective) overview of the central aspects of the computational Sentiment Analysis problem, and in particular pointed to some interesting recent work. We mentioned several performance results in order to give an impression on the extent to which the various problems can at present be solved with automatic methods.

The other survey was certainly very selective and discussed a number of notions from the Linguistics literature on Subjectivity, as we see them to be relevant for Sentiment Analysis; prominent topics here were the 'private state' and the facets of 'evidentiality' and 'factuality', which deserve close attention when extracting sentiment-related information from text.

In the final section of the chapter, our goal was to identify several critical issues with Sentiment Analysis and at some points to suggest possible steps toward finding solutions. As a general sentiment (pardon the pun) we believe that more detailed linguistic analysis would be instrumental for making progress with high-quality and fine-grained Sentiment Analysis, which requires careful analysis of contextual effects for identifying sources of opinions, for computing polarity in compositional ways, and for a more sophisticated identification of the entities that can be assigned sentiment values in complex sentences.

As a final remark, we want to point out that we touched only very superficially on the difference between opinions or emotions as mental states on the one hand, and the linguistic utterances speakers produce to express them on the other hand. This distinction is connected to various parts of the picture we presented. At the beginning of a text production process, there are mental states and stances of the author; they influence both the selection of information that gets verbalised (*what* to say) and the actual shape of the verbalisation (*how* to say it: the linguistic choices). The decision on what to say is already a matter of Subjectivity, as we mentioned briefly at the end of Sect. 2.2.2. Then, when the author makes choices among lexical and grammatical options, she can opt to clearly mark the intended factuality or the various dimensions of Subjectivity, or she can leave that underspecified (deliberately or inadvertently), which in turn leaves room for interpretation by the addressee. That is one major reason why Sentiment Analysis is *inherently* very challenging—not only for the machine.

Acknowledgements We thank our anonymous reviewers for detailed and constructive criticism, which enabled us to improve this work at several points.

References

1. Anscombre J-C, Ducrot O (1983) L'argumentation dans la langue. Editions Margana, Sprimont
2. Balahur A (2011) Methods and resources for sentiment analysis in multilingual documents of different text types. Dissertation, Department of Software and Computing Systems, University of Alacant, Alacant
3. Baroni M, Bernardini S, Ferraresi A, Zanchetta E (2009) The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Lang Resour Eval 43(3):209–226
4. Bühler K (1934) Sprachtheorie: Die Darstellungsfunktion der Sprache (=UTB für Wissenschaft; 1159). 3 Auflage. G. Fischer, Stuttgart [u.a.], 1999. ISBN 3825211592
5. Carbonell JG (1979) Subjective understanding: computer models of belief systems. PhD dissertation, Yale University, Also Yale U. Comp. Sci. Report #150
6. Clematide S, Gindl S, Klenner M, Petrakis S, Remus R, Ruppenhofer J, Waltinger U, Wiegand M (2012) MLSA – a multi-layered reference corpus for german sentiment analysis. In: Proceedings of the 8th international conference on language resources and evaluation (LREC'12), Istanbul, pp 3551–3556
7. Dendale P, Tasmowski L (2001) Introduction: evidentiality and related notions. J Pragmatics 33:339–348
8. Esuli A, Sebastiani F (2005) Determining the semantic orientation of terms through gloss classification. In: Proceedings of the 14th ACM international conference on information and knowledge management, CIKM'05, New York, pp 616–624
9. Hatzivassiloglou V, McKeown KR (1997) Predicting the semantic orientation of adjectives. In: Proceedings of the 35th annual meeting of the association for computational linguistics and 8th conference of the European chapter of the association for computational linguistics. Association for Computational Linguistics, Madrid, pp 174–181
10. Heider F (1958) The psychology of interpersonal relations. Wiley, New York
11. Hu M, Liu B (2004) Mining opinion features in customer reviews. AAAI 4(4):755–760
12. Karttunen L, Zaenen A (2005) Veridicity. In: Katz G, Pustejovsky J, Schilder F (eds)Annotating, extracting and reasoning about time and events. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl
13. Kiritchenko S, Zhu X, Cherry C, Mohammad S (2014) Detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th international workshop on semantic evaluation exercises (SemEval-2014), Dublin
14. Klinger R, Cimiano P (2013) Bidirectional inter-dependencies of subjective expressions and targets and their value for a joint model. In: Proceedings of the 51st annual meeting of the association for computational linguistics. Association for Computational Linguistics, Sofia, pp 848–854
15. Klinger R, Cimiano P (2014) The USAGE review corpus for fine grained multi lingual opinion analysis. In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, pp 2211–2218
16. Lyons J (1982) Deixis and subjectivity: Loquor, ergo sum? In: Jarvella R, Klein W (eds) Speech, place, and action. Studies in Deixis and related topics. Chichester, pp 101–124
17. Langacker RW (1987) Foundations of cognitive grammar. Volume 1, Theoretical prerequisites. Stanford University Press, Stanford
18. Martin JR, White PRR (2007) The language of evaluation: appraisal in english. Palgrave Macmillan, Basingstoke
19. Ortony A, Clore GL, Collins A (1990) The cognitive structure of emotions. Cambridge University Press, Cambridge
20. Pang B, Lee L (2003) A sentimental education: sentiment analysis using subjectivity based on minimum cuts. In: Proceedings of the 42nd meeting of the association for computational linguistics (ACL'04), Main Volume, Barcelona, pp 271–278
21. Pang B, Lee L (2004) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05), Arbor, pp 115–124

22. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inform Retrieval 2:1–135
23. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002), Philadelphia, pp 79–86
24. Quasthoff U (2010) Deutsches Kollokationswörterbuch. deGruyter, Berlin/New York
25. Quirk R, Greenbaum S, Leech G, Svartvik J (1972) A grammar of contemporary english. Longman, London
26. Ramaswamy S (2011) Visualization of the sentiment of the tweets. Department of Computer Science, North Carolina State University, Raleigh
27. Remus R, Quasthoff U, Heyer G (2010) SentiWS – a publicly available German-language resource for sentiment analysis. In: Proceedings of the 7th international language resources and evaluation (LREC), Istanbul
28. Rundell M, Fox G (2002) Macmillan english dictionary for advanced leaners. Macmillan, Oxford
29. Socher R, Perelyglin A, Wu J, Chuang J, Manning C, Ng A, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: EMNLP 2013, Seattle
30. Stone P, Dexter D, Smith M (1966) The general inquirer: a computer approach to content analysis. MIT Press, Cambridge
31. Al Masum Shaikh M, Prendinger H, Ishizuka M (2010) Emotion sensitive news agent (ESNA): a system for user centric emotion sensing from the news. Web Intell Agent Syst 8(4):377–396
32. de Spinoza B (1677) Ethics, and on the correction of understanding [Translated by Andrew Boyle (1986)]. Dent, London
33. Taboada M, Brooke J, Stede M (2009) Genre-based paragraph classification for sentiment analysis. In: Proceedings of the 10th SIGdial workshop on discourse and dialogue, London
34. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Ling 37(2):267–307
35. Thielen C, Schiller A (1996) Ein kleines und erweitertes Tagset fürs Deutsche. In: Feldweg H, Hinrichs E (eds) Lexikon & Text. Neimeyer, Tubingen, pp 215–226
36. Turney P (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, Philadelphia, pp 417–424
37. Toprak C, Jakob N, Gurevych I (2010) Sentence and expression level annotation of opinions in user-generated discourse. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Uppsala, pp 575–584
38. Verhagen A (2005) Constructions of intersubjectivity: discourse, syntax, and cognition. Oxford University Press, Oxford/New York
39. Wiebe J, Wilson T, Cardie C (2005) Annotating expressions of opinions and emotions in language. Lang Resour Eval 39:165–210
40. Wiegand M (2011) Hybrid approaches for sentiment analysis. Universität des Saarlandes, Saarbrücken
41. Wilson T (2008) Annotating subjective content in meetings. In: Proceedings of the 6th international language resources and evaluation conference (LREC), Marrakech, Morocco
42. Wilson T, Wiebe J, Hoffman P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of HLT-EMNLP-2005, Sydney
43. Zastrow V (2013) Das Amalgam. Frankfurter Allgemeine Sonntagszeitung. http://www.faz.net/-gpg-762nc

Multi-perspective Event Detection in Texts Documenting the 1944 Battle of Arnhem

Marten Düring and Antal van den Bosch

Abstract We present a pilot project which combines the respective strengths of research practices in history, memory studies, and computational linguistics. We present a proof-of-concept workflow for the semi-automatic detection and linking of narratives referring to the same event based on references to location names. We rely on the interaction between human evaluation, entity extraction, mapping, and network visualization techniques. We work with 83 narratives and reports surrounding the Battle of Arnhem in 1944. The liberation of the Netherlands led to frequent encounters between civilians and soldiers in the war zones. We seek to find multi-perspective descriptions of these interactions marked by a high degree of uncertainty, differing anticipations and sometimes violence. A proof-of-concept study shows that we cannot rely on standard named-entity recognition but need to develop fine-grained detection of street names, to capture the scenes that connect multi-perspective narratives.

1 Introduction

In this paper we present the first findings from an interdisciplinary research project titled MERIT—Machine-based Extraction of Relations in Text. In recent times, the digitization of archival records and secondary sources have resulted in easier access to historical sources. This availability of digital full-text documents and metadata has created new possibilities for the computer-assisted exploration of historical sources [23]. However, to date only few archives offer cross-catalogue (or federated) search, offering mostly only metadata search. Full-text search of object content

M. Düring (✉)
University of North Carolina, Chapel Hill, NC, USA
e-mail: martenduering@gmail.com

A. van den Bosch
Centre for Language Studies, Radboud University, Nijmegen, The Netherlands

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_10

is still rare; one example is Delpher,[1] developed by the National Library of the Netherlands.

Recent advances in computational linguistics have led to the availability of reasonably robust and trainable tools for the extraction of information from unstructured and semi-structured historical sources [24] and the more advanced study of the automatic identification of recurring story elements (motifs, topoi) in narratives [15]. The currently most reliable tools and methods developed in text analytics, an applied subfield of computational linguistics and the related area of information retrieval, are by now capable of identifying names of people, organizations, places and time expressions. A number of projects apply these tools to selected historical source collections already [13, 20, 22]. Other efforts focus on connecting these entities according to relations described in the text [5, 19, 27].

All of these tools are suited to aid research in contemporary history. Our own experiences with related methods show that the semi-automated analysis of sources can lead to the serendipitous discovery of hitherto unknown and unsuspected links between the events they depict [7, 31].[2]

A number of projects have taken on the task to model historical events while fitting into the historian's typical workflow, focusing on aspects such as crowdsourced annotations [12], combining machine annotations with human annotations [18],[3] and by modeling alternative interpretations of events [21]. Closest to our project is work by [19,20,22] who bring multi-perspective memories and space together using parallel axes. These projects share the necessity of elaborate visualization to become useful for people interested in a particular set of sources. Most scholars in the field of digital humanities are ready to acknowledge that computational approaches alone can not compete with the close reading of domain experts, the hallmark of traditional historical research. The inherent imprecision of the computational approach is a major drawback for those interested in the detective side of history, i.e. coupling sources with the aim to reconstruct individual events.

Here we present our approach to reconciling historical research methods with text analytics and visualization tools. Our project's historical component focuses on the end of the Second World War and the task of collecting, identifying, analyzing, and linking a large number of sources related to the Battle of Arnhem in 1944, online and spread across archives and museum collections in the Netherlands, Germany, the United Kingdom, and the USA. The liberation of the Netherlands led to frequent encounters between civilians and soldiers in the war zones. These interactions were marked by a high degree of uncertainty, differing anticipations and violence. The study of multi-perspective representations of a large number of these encounters helps us to better understand how key elements of European post-war concepts such as "Occupation", "Liberation", "Defeat", and "Victory" unfolded at ground level

[1] http://www.delpher.nl/.

[2] For a more playful approach, cf. Serendip-o-matic, http://www.serendipomatic.org/.

[3] Cf. the heureCLÉA project, http://www.heureclea.de/ and Recogito, http://www.pelagios.org/recogito/static/docs/index.html.

and transformed into memories. We concentrate on micro-historical reconstructions of this epochal period of transition, and ask how memories of individual encounters have been shaped within and across social communities. For a multi-perspective reconstruction of these encounters and their commemoration we rely mainly on digitized sources and metadata on eyewitness reports of all sides, oral history interviews, and secondary sources. We start from the above mentioned techniques and develop them further for the analysis of multi-lingual texts and different text genres (civilian recollections, soldier and historical narratives, war diaries, etc.).

We begin by outlining current practices and limitations (Sect. 2), go on to present our conceptual approach for the project (Sect. 3) and conclude with a report on a proof-of-concept study using primary sources on the Battle of Arnhem in 1944 (Sect. 4).

2 Synthesizing Computational and Historical Research Practices

Digitized primary sources of relevance to historical research are now omnipresent. Interactive digital history projects today provide roughly five ways of searching through them:

- Browsing, similar to working with archival finding aids (this of course is not strictly searching);
- Keyword searches in metadata and in full texts;
- Computational approaches to identify patterns and structure in text collections (such as topic modeling);
- Visualizations based on laborious manual annotation;
- Visualizations based on automated annotations.

Keyword and metadata searches alone allow us to find information in seemingly unlimited amounts of text. Some projects already combine several strategies. One of them is ePistolarium,[4] a tool developed in the course of the Circulation of Knowledge and Learned Practices in the seventeenth-century Dutch Republic project[5] at the Huygens ING Institute in the Netherlands. ePistolarium provides basic search, faceted search, ways to add Named Entity Recognition output to the search and various data visualization tools in a well-designed interface. Projects like this require the collaboration of historians with specialists in interface design, computational linguistics, information retrieval, and data visualization. Its overall ambition is to be the go-to place for anyone interested in seventeenth century scientific letters. ePistolarium is custom-built and users are therefore limited to work with materials provided by the project.

[4] http://www.ckcc.huygens.knaw.nl/epistolarium/.

[5] http://www.ckcc.huygens.knaw.nl/.

Computational approaches are great to process large quantities of texts and to find statistically significant patterns in them. Good examples of their application to historical texts can be found in literature research such as in Ted Underwood's work on constellations of genres in nineteenth century literature [26] and in Matthew L. Jockers' works [14] challenging existing notions of literary epochs. A good example of this approach in historical research is *Mining the Dispatch*,[6] created by Robert Nelson at the University of Virginia's Digital Scholarship Lab. The project bases itself on the near complete run of the digitized Richmond Daily Dispatch. Nelson makes the most of topic modeling using MALLET [17]. *Mining the Dispatch* provides visitors to the site with some historical context on each of the topics, an explanation of how topics were identified, and most importantly with plenty of examples taken from the Dispatch and opportunities to change thresholds and chart types.

Keyword searches are a simple yet powerful way to process large collections of texts, but require researchers to know precisely what they are looking for—in information retrieval terms, this is referred to as *known-item search*. Yet, anything, not just predefined known items, can turn out to be of relevance for researchers, and relevant documents in which we can find these items are not necessarily known beforehand to historians. As they read through documents historians become aware of notable individuals, institutions, events or places. Ideas about how they relate together emerge as part of this process. This provides them with a sense for the texts they study, how they are written, by whom and why. In-depth interaction with text allows historians to handle ambivalences, negotiate contradictions, and is the basis of their attempt to understand historical processes. A crucial part of this process is the "coupling" (Raul Hilberg) of sources, relating them to each other and having them shed light on each other. This approach captures the art of doing history and the ambition to provide "appropriately complex descriptions of similarly complex situations" as historian Achim Landwehr puts it.

In practice, historians typically gather a deep understanding of a limited body of texts they manage to read themselves, relying on archival finding aids, archivists, the grapevine, and other sources when it comes to finding them. These finds prompt them to revise search strategies and direct future research endeavours. On the one hand, historians require highly precise results; on the other hand they face limited capacities to process a larger number of texts by reading and keyword searches alone. Based on previous experience [6,10,28,30] and on our proof-of-concept study we argue that this problem can partly be solved by integrating human evaluation and information processing with machine-based text analytics as outlined in Table 1. We envision a system that supports the analytical skills of humans through smart and flexible suggestions, and through search, filter and visualization techniques.

Table 1 lays out the components of such an interaction that combines the respective strengths of historians and machines.

[6] http://www.dsl.richmond.edu/dispatch/.

Table 1 Schema for historian-machine interaction

Step	Action	Historian	Machine
1	Text selection	Identify corpus	—
2	Identifying relations between texts	Exploration of types of relations	Cluster texts based on types of relations
3	Visualizing relations between texts	Selection/prioritization of relevant texts	Networks, maps
4	Information processing	Close reading	Visualizations as representations of cognitive maps
5	Adjustment	Revise corpus, search queries and parameters. Update Step 1	Update Steps 2–4
6	Communication	Selection of suitable contents	Interactive, illustration and extension of textual contents

Digitization has changed the ways in which historians approach Step 1, *Text selection*: Huge repositories of digitized documents are freely available online and their value for any given research question is unclear initially. We entrust historians to make this first choice and decide which corpora are of relevance.

Step 2, *Identifying relations between texts*, is concerned with the identification of relations in and between texts. Humans should make these choices as they depend on their research interests and knowledge of the source materials. Inasmuch as they texts can be expressed in machine-comprehensible form, e.g. as vectors in multi-dimensional spaces (‘bags of words’) or mixtures of topic models, we rely on machines to cluster related texts together by spatial similarity and to thereby provide a first overview of how common certain types of relations are.

In Step 3, *Visualizing relations between texts*, historians use the machine output to prioritize and plan their analysis of the corpus. In our case studies we made use of network visualizations to help us with this.

Step 4, *Information processing*, is characterized by the in-depth human analysis of the pre-selected documents. Domain context knowledge and external (non-digitized) sources are needed in this phase. Visualizations can interact with cognitive maps as a means to plot existing knowledge and to gain new perspectives on it.

Step 5, *Adjustment*, suggests where new knowledge could be found, leading to the revision or specification of the previous steps.

Finally, step 6, *Communication*, describes the construction of a narrative, still the preferred means of communicating historical research. Traditional narrative structural constraints force historians to organize their findings into a more or less linear narrative and to make choices regarding which contents to include and which to neglect. Non-linear, enriched, and other alternatives which make the most of digital technology are too numerous to mention.

We will revisit this workflow throughout the description of our pilot study in Sect. 4. In summary, we rely on text analytics and visualizations throughout the research process as a means to help us identify relevant materials, to process information, to expand and reconfigure queries, and finally as a means to communicate findings.

3 About MERIT

In this paper we focus on the problem of cross-referencing information across a large number of sources, a common task in historical research. Historians have always relied on tools to help them with information management. Notebooks, file boxes, and relational databases have all been put to good use for this purpose. MERIT, short for Machine-based Extraction of Relations In Text, started out as a research proposal for a call by the Dutch funding agency NWO titled "Creative Industries" in 2012. At the time, MERIT failed to get funded but it has laid the foundation for ongoing discussions between computational linguistics, historians, social psychologists, and cultural heritage professionals on how to best combine strengths and compensate for weaknesses in each others practices.

With MERIT we intend to develop new methods to make these sources usable for historians. We envision new methods for the aggregation and analysis of digitized historical sources, specifically with the goal to (1) identify and group related documents across thousands of sources and (2) compare the structure of narratives and study how they have evolved over time. MERIT's technologies are being developed with general applicability in mind, so that they may be used for the analysis of other digitized texts such as source editions, newspapers, or interviews.

In this early stage MERIT seeks to help scholars to partially automatize and expand information management, by detecting named entities in sources and creating links between them. By mapping locations mentioned in sources both geographically and as networks, MERIT provides a new perspective on how sources relate to each other when discussing events that occurred at a certain location. Before we discuss these aspects further, we provide context on our case study.

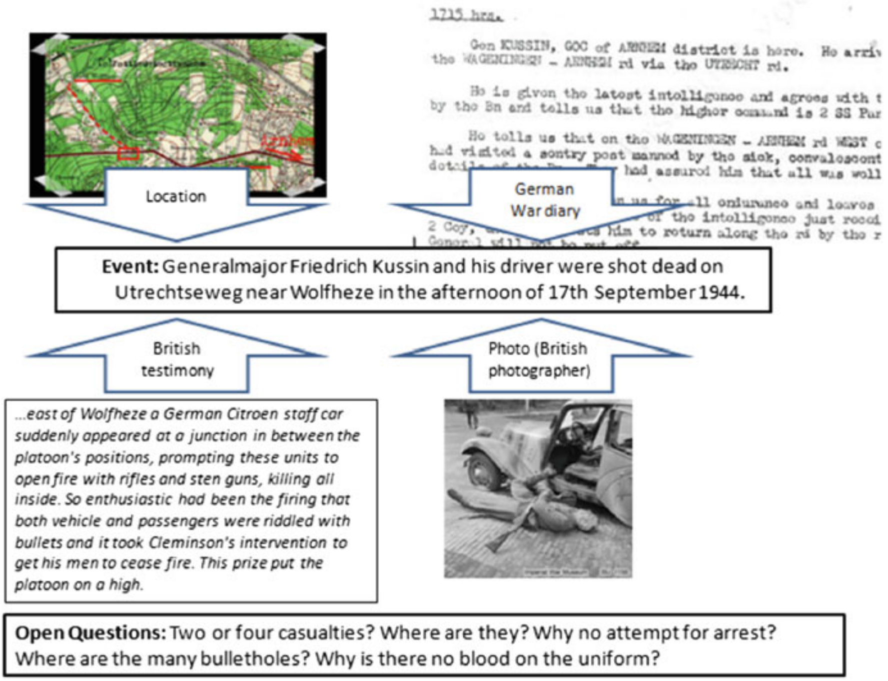
3.1 Proof of Concept Study: The Battle of Arnhem

Between September 1944 and March 1945 the Dutch and German border region between the cities of Arnhem, Nijmegen, and Kleve became one of the frontlines of the Second World War. The German historian Klaus-Dietmar Henke refers to this period as one of "highest historical acceleration" ("größter historischer Beschleunigung") [11, 31] and the touching point between two epochs: The end of fascism and the beginning of the divide between the Western and the Eastern block. On 17 September, civilians and Wehrmacht soldiers witnessed Operation Market Garden, the biggest air landing operation of its time. Following the arrival of tens of thousands of Allied paratroopers, German troops seemed to flee at first, but then quickly regrouped and withstood the Allied offensive. Mainly due to poor planning, equipment and intelligence, Operation Market Garden failed its ambitious goal to occupy six strategically important bridges in the region. What followed was a standoff during which the frontline remained stable across the Arnhem–Nijmegen–Kleve region and which turned the region into a war zone. Significantly weakened, but still unexpectedly strong, German troops remained on one side, while Allied reinforcements built up on the other.

Most historical analyses of the Arnhem–Nijmegen–Kleve region in the period September 1944 to March 1945 have an explicit interest in traditional military history [1, 16]. Despite their undoubted value for military historians, these accounts leave remarkably little room for the experiences of civilians: except for curt descriptions of resistance activities and evacuations, these appear to mainly have "danced in the streets and cheered their liberators" [25]. Analogously, local histories of the region typically stick to a strictly local focus on the experiences of Dutch or German civilians and do not seek to explore their place in local and national memory cultures [8]. An exception is Klaus-Dietmar Henke's classic study on the American occupation on the German side, which seeks to integrate military and German civilian perspectives, however admits to its merely anecdotal references to the civilian experience of the events and does not investigate the memory discourses of the period it covers [11].

Consider the example depicted in Fig. 1. Each of the four source snippets[7] contains enough hints to group them together as referring to the same event: mentions of the location, the "Utrechtseweg" (Utrecht Road) near the village of "Wolfheze", the date and time (as noted in a German war diary, translated to English by Allied intelligence), and an annotated photo taken by an Allied photographer. Rather than merely providing parts of a story that can simply be concatenated, the aggregation of sources from different perspectives and a close look at what they depict helps us to reconstruct what happened. A key question with regard to this particular event is, why there was no attempt to arrest the high ranking Generalmajor Friedrich Kussin, who was in charge of all German troops in the Arnhem area. Additional research has revealed that Lieutenant Cleminson had simply failed to recognize Kussin.

[7] The sources surrounding this example were researched by amateur historian Tom Timmermans (www.battledetective.com/Kussin_Junction.html). The event has also been described by professional historians in most historical reconstructions of Operation Market Garden including [4, 263].



1715 hrs.

Gen KUSSIN, GOC of ARNHEM district is here. He arri
the WAGENINGEN – ARNHEM rd via the UTRECHT rd.

He is given the latest intelligence and agrees with t
by the Bn and tells us that the higher command is 2 SS Pz

He tells us that on the WAGENINGEN – ARNHEM rd WEST c
had visited a sentry post manned by the sick, convalescent
det had assured him that all was well

n us for all endurance and leaves
of the intelligence just recei
2 Coy, him to return along the rd by the r
General will not be put off

Event: Generalmajor Friedrich Kussin and his driver were shot dead on Utrechtseweg near Wolfheze in the afternoon of 17th September 1944.

...east of Wolfheze a German Citroen staff car suddenly appeared at a junction in between the platoon's positions, prompting these units to open fire with rifles and sten guns, killing all inside. So enthusiastic had been the firing that both vehicle and passengers were riddled with bullets and it took Cleminson's intervention to get his men to cease fire. This prize put the platoon on a high.

Open Questions: Two or four casualties? Where are they? Why no attempt for arrest? Where are the many bulletholes? Why is there no blood on the uniform?

Fig. 1 Four document snippets referring to the killing of German Generalmajor Kussin on 17th September 1944

In order to facilitate these kinds of source aggregation tasks, MERIT calls for the extension and combination of existing tools. We will make us of multi-lingual (Dutch, English, and German) tools for the detection of events [3]. There are substantial challenges in detecting that two documents describe the same event. The software needs to be able to identify that mentioned entities are the same ("A high-ranking German officer" and "Major-General Kussin"), and that the two documents describe the same event (the killing of Kussin) on the grounds of overlapping mentions of time and location.

3.2 *Methodology*

We foresee a broad experimental matrix for finding optimal choices of feature selection, event detection and tracking, and document matching methods, ranging from exploiting simple overlap metrics (the same entities being mentioned) to advanced metrics that exploit latent variables [9].

MERIT does not operate with a predefined list of what counts as an event and what does not. Instead, its bottom-up approach reveals automatically what kinds of events were documented by contemporaries on multiple occasions. Their occurrence in more than a single source is analogous to a node having a larger number of links in an underlying conceptual network. The connectivity of the event node may determine its ranking in the list of potentially significant events [2]. During the analysis we intend to explore different means to rank, filter and threshold candidate events to be presented to the expert user.

We intend to move beyond the current state of the art in automatic event recognition: Aside from being able to identify names of persons, locations, organizations,[8] and time expressions,[9] As we work with texts written in English, Dutch, and German, we will combine the use of pre-existing tools with new cross-lingual tools. The lack of proper multi-lingual domain ontologies furthermore calls for a data-driven, bottom-up approach, although we will use specific ontologies (e.g. military) if they provide useful prior knowledge.

MERIT's knowledge base will initially contain hundreds to thousands of candidate events, which will subsequently be merged if they turn out to refer to the same event mentioned in sources from different perspectives. The knowledge base then allows for targeted querying, for example, which other events Kussin was involved in before leading to his death. It will also allow for more abstract questions to be asked, e.g. on the occurrence of events within a particular time or location range.

4 A Pilot Study

4.1 Step 1: Text Selection

We started with a first proof-of-concept study to explore the extent to which place names can aid the exploration of a collection of texts. To this end we present both a geographical map and a network graph. For this first experiment we worked with 83 texts, most of them eye-witness accounts and memoirs but also letters to the museum, a detailed police report in Dutch, three digitized historical monographs on the event and digitized book indices, all referring to the nine days of Operation Market Garden with a focus on the city of Arnhem (see Table 2). The eye-witness accounts provide us with descriptions of the events from local citizens who describe the landing of Allied troops, German reactions, combat, destruction, interactions with soldiers, and their personal evacuation from Arnhem. The police report

[8] For Dutch named entity recognition we can rely on Frog (http://www.ilk.uvt.nl/frog); for German and English we can use the Stanford NER tools (http://www.nlp.stanford.edu/software/CRF-NER.shtml).

[9] The Heideltime tagger can automatically detect time expressions in English, German, and Dutch texts (http://www.dbs.ifi.uni-heidelberg.de/index.php?id=129).

Table 2 Distribution of text types and languages

Language	Civilian narratives	Soldier narratives	Historical monographs	Other
Dutch	55	—	—	16
English	1	7	3	1

contains a selection references to human remains and graves discovered in Arnhem between 1945 and 1954 and was compiled by Airborne Museum “Hartenstein”.[10] This is a small subset of the overall collection, most of which requires OCR analysis to extract the text from scanned images. At this stage we relied only on digitally transcribed or digitally born documents, all of which where either written in Dutch or English. Finally, we digitized indices of 21 historical narratives which describe the Battle of Arnhem from Allied, Dutch and German perspectives. This corresponds to Step 1 as outlined in Table 1.

We sought to focus on creating a workflow which would teach us about the chances and challenges along the way. This workflow included the transformation of Word documents into text files, the identification of street names, the identification of corresponding latitude and longitude values, the creation of an Evernote archive file with one note for each streetname found, as well as the creation of a network file based on co-occurrences of streetnames in documents (Table 2).

4.2 Step 2a: Named Entity Recognition

In the case study we present here we chose space as the means by which we relate documents to each other (see: Step 2/Historian in Table 1); references to the same area indicate potential relatedness. We first applied automatic machine-learning-based named-entity recognition methods to the Dutch and English texts (Step 2/Machine). The named-entity recognition that is part of the Dutch morpho-syntactic parser Frog [29] finds 6,927 locations in the 71 Dutch texts. The five most frequently mentioned location names are Dutch city and village names, unsurprisingly headed by Arnhem with 201 mentions, followed by two villages close to Arnhem that saw heavy fighting in the Arnhem Battle (Rhenen and Oosterbeek). Other frequently used names mention Germany and the river Rhine, as well as cities where civilians were evacuated to after the battle.

The Stanford named-entity recognition module finds no less than 17,221 locations in the 12 English texts. Many of these entities are found in one book [25] covering the history of the Canadian Army during the Second World War. Discarding this particular book, the module still detects 8,170 locations. As most of the English documents are soldier memoirs, we find frequent references to

[10] http://www.en.airbornemuseum.nl/.

major cities (Antwerp, Aachen, Nijmegen, Arnhem), countries, and regions such as Normandy where the soldiers fought.

Despite the fact that many of the location entities discovered by the module are correct locations, the modules accept too many different types of locations under their common location entity denominator to be useful for our purposes. Our method is based on finding documents that mention common location names. The hundreds of mentions of "Arnhem" link any particular event in the Arnhem region to any other event. Linking documents should be done at the granularity level at which events occur that are significant for our goal. Battles like the Battle of Arnhem were fought street by street, and civilians tend to report from their a limited radius around their house, the street and possibly nearby streets. Thus, buildings and streets constitute the granularity level we need to restrict our methods to.

4.3 Step 2b: Regular Expressions for Street Names

The restriction to street names, a finer granularity level than the 'location' label identified by the machine-learning-based named-entity recognizers, suggests at least two alternative approaches. First, we could retrain the machine-learning classifiers at the finer granularity level, to recognize only street names. Second, given the fair amount of systematicity of Dutch street names, we could formulate regular expressions that would cover this systematicity. We established a list of twelve common Dutch and English suffixes (as English writers would often choose English street suffixes when referring to Dutch roads), listed in Table 3.

Table 3 The 12 tested street name suffixes and their numbers of matches, including matching cases of house numbers

Suffix	Matches	With house numbers
–weg	301	56
–road	143	0
–straat	129	19
–dijk	63	0
–brug	56	0
–laan	47	18
–plein	28	0
–singel	11	0
–pad	10	0
–street	8	0
–park	8	0
–path	3	0

Table 4 Numbers of gold and predicted names, and false positives and negatives

# Texts	# Street names in text		False positives	False negatives
	manual (gold)	automatic		
30	203	205	23	14

We searched for these occurrences and retained matching words starting with a capital letter; we then attached any left-neighboring words starting with a capital letter as well. This method succeeds in finding for example "Bakenbergse weg" as well as "Eusebius Buitensingel". Note that rare street names not ending in a typical street name suffix are not detected by this method. This approach is satisfactory given the explorative nature of this project.[11] The middle column of Table 3 lists the number of matches we find of the fourteen suffixes, totalling to 849 matched street names.

In order to estimate the precision and recall of this regular expression method we manually annotated 30 of the documents for actual occurrences of street names. Their number, as well as the number of matched street names in these 30 documents and the number of false positives and negatives of the automatically matched names are listed in Table 4. We observe that the number of mistakes is overall relatively low and that most of them are false positives. These numbers amount to a precision of 88.8 %, a recall of 89.7 %, and an F-score (with $\beta = 1$) of 89.2 %, showing the success of the relatively simple regular expression-based approach.

We plan to extend this necessarily limited pattern further by for example also considering references to places of significance such as "the old railroad bridge in the city center" and "St. Elizabeth Hospital" by building a repository of place names. Some names of significant buildings may be captured with the patterns "–huis" ("–house"), "–gebouw" ("–building"), and "–kazerne" and "–bureau" ("–barracks", "–station").

Another extension that could be captured with regular expressions is the recognition of house addresses, which in Dutch are typically designated by a street name followed by a digit. Table 3 shows in its rightmost column the number of hits of street names followed by integer numbers, which on visual inspection all refer to house addresses (e.g. "Bennekomseweg 73", "Hekerenlaan 35"). The total of hits is 93, all occurring with the most typical Dutch street name suffixes "–weg", "–straat", and "–laan". Having this information allows an extra level of precision in pinpointing the co-ordinates of events.

[11] In the future we will combine the suffix-based identification of potential place names with openly available gazetteer data such as Geonames and Nominatim (Mapquest/Openstreetmap) to run searches through the entire texts. This approach will be capable of identifying for example "Onder de Linden" but fail to detect any misspelled street names. mapalist.com's lookup functionality which relies on Google Maps and others would find for example the misspelled "Utrechseweg" and correctly identify it as "Utrechtseweg".

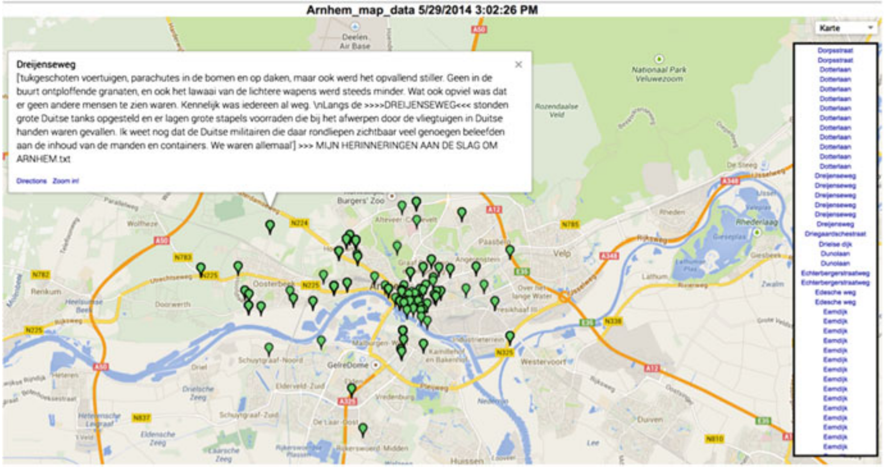
4.4 Step 3: Visualization of Relations Between Texts

For now we rely on mapalist.com to retrieve geo-coordinates based on the 849 automatically detected street names, and map them together with corresponding excerpts on Google Maps, as shown in Fig. 2.

The advantage is obvious: One no longer needs to know all streets in Arnhem to identify which reports refer to neighbouring streets. Mapalist identified 456 of the 849 street names as streets in Arnhem.[12] The remaining 393 street names refer to places outside the city or are other false positives. This output however needs to be evaluated and organized by a human.

In parallel, and to get a better sense of how our sources relate to each other, we created a network which represents texts as nodes and co-occurring street names as edges. The network is displayed in Fig. 3, in which the nodes are colored according to Gephi's[13] modularity class clustering method.

This graph gives an intuitive sense of the number of street names detected in each text and how they cluster together. This acts as guidance for the selection of texts to read together: Since co-occurring street names indicate at least some overlap, it makes sense to organize one's reading by cluster. For instance, the graph clusters a number of English historical resources together (the bottom cluster of four). It furthermore suggests that the police reports of 1945–1954 are quite central


Arnhem_map_data 5/29/2014 3:02:26 PM

Dreijenseweg
[tukgeschoten voertuigen, parachutes in de bomen en op daken, maar ook werd het opvallend stiller. Geen in de buurt ontploffende granaten, en ook het lawaai van de lichtere wapens werd steeds minder. Wat ook opviel was dat er geen andere mensen te zien waren. Kennelijk was iedereen al weg. \nLangs de >>>>DREIJENSEWEG<<< stonden grote Duitse tanks opgesteld en er lagen grote stapels voorraden die bij het afwerpen door de vliegtuigen in Duitse handen waren gevallen. Ik weet nog dat de Duitse militairen die daar rondliepen zichtbaar veel genoegen beleefden aan de inhoud van de manden en containers. We waren allemaal'] >>> MIJN HERINNERINGEN AAN DE SLAG OM ARNHEM.txt

Dorpsstraat
Dorpsstraat
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dotterlaan
Dreijenseweg
Dreijenseweg
Dreijenseweg
Dreijenseweg
Dreijenseweg
Dreijenseweg
Driegaardschestraat
Drielse dijk
Dunolaan
Dunolaan
Echterbergerstraatweg
Echterbergerstraatweg
Edesche weg
Edesche weg
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk
Eemdijk

Fig. 2 Visualizing excerpts using mapalist.com and Google Maps

[12] Mapalist.com was queried with triples of detected street names ("Bennekomseweg") with "Arnhem" and "The Netherlands", and either returned longitudinal and latitudinal coordinates, or defaulted to a central point designating Arnhem.

[13] https://www.gephi.org/.

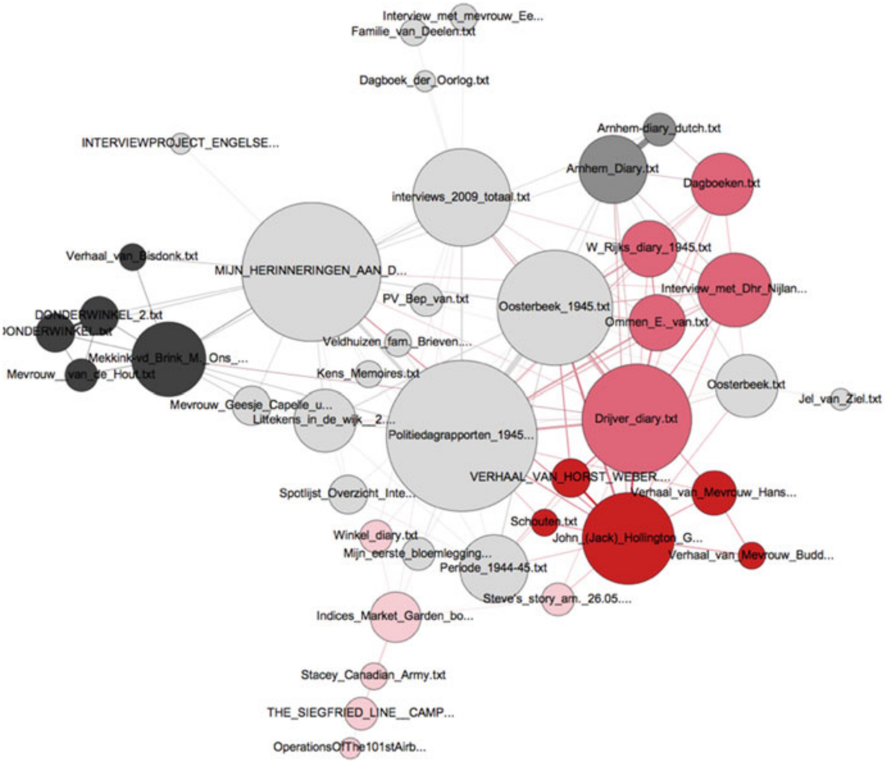

Interview_met_mevrouw_Ee...
Familie_van_Deelen.txt
Dagboek_der_Oorlog.txt
INTERVIEWPROJECT_ENGELSE...
Arnhem-diary_dutch.txt
Arnhem_Diary.txt
Dagboeken.txt
interviews_2009_totaal.txt
Verhaal_van_Bisdonk.txt
MIJN_HERINNERINGEN_AAN_D...
W_Rijks_diary_1945.txt
Interview_met_Dhr_Nijlan...
PV_Bep_van.txt
Oosterbeek_1945.txt
Ommen_E._van.txt
DONDERWINKEL_2.txt
ONDERWINKEL.txt
Mekkink-vd_Brink_M._Ons_...
Veldhuizen_fam._Brieven...
Mevrouw_van_de_Hout.txt
Kens_Memoires.txt
Oosterbeek.txt
Jel_van_Ziel.txt
Mevrouw_Geesje_Capelle_u...
Littekens_in_de_wijk_2...
Drijver_diary.txt
Politiedagrapporten_1945...
VERHAAL_VAN_HORST_WEBER...
Verhaal_van_Mevrouw_Hans...
Spotlijst_Overzicht_Inte...
Schouten.txt
John_(Jack)_Hollington_G...
Winkel_diary.txt
Mijn_eerste_bloemlegging...
Verhaal_van_Mevrouw_Budd...
Periode_1944-45.txt
Steve's_story_am._26.05....
Indices_Market_Garden_bo...
Stacey_Canadian_Army.txt
THE_SIEGFRIED_LINE__CAMP...
OperationsOfThe101stAirb...

Fig. 3 Texts connected by co-occurring street names. Node size represents degree, node colour modularity class and edge with number of co-occurring street names. 41 nodes (hiding 10 isolates), 126 edges visualized in Gephi (Force Atlas 2, Label Adjust)

(as they mention many explicit street names) and could be therefore be relevant accompanying reading material to most other documents.

5 Step 4: Information Processing

Sifting through the material and sorting out mismatches requires a tool (we chose Evernote, as shown in Fig. 4) which at the very least can store excerpts, full texts and names of events, makes these searchable, and lets one organize them using tags and a hierarchical system. This list of created tags which also indicate the number of notes associated to them helps users to keep an overview of the process. Other advantages of using Evernote are easy editing of notes (including images and pdfs) and tags (as shown in Fig. 5) as well as multiple opportunities for future retrieval, adding and sharing of notes using e.g. Browser add-ons, file import and export, an

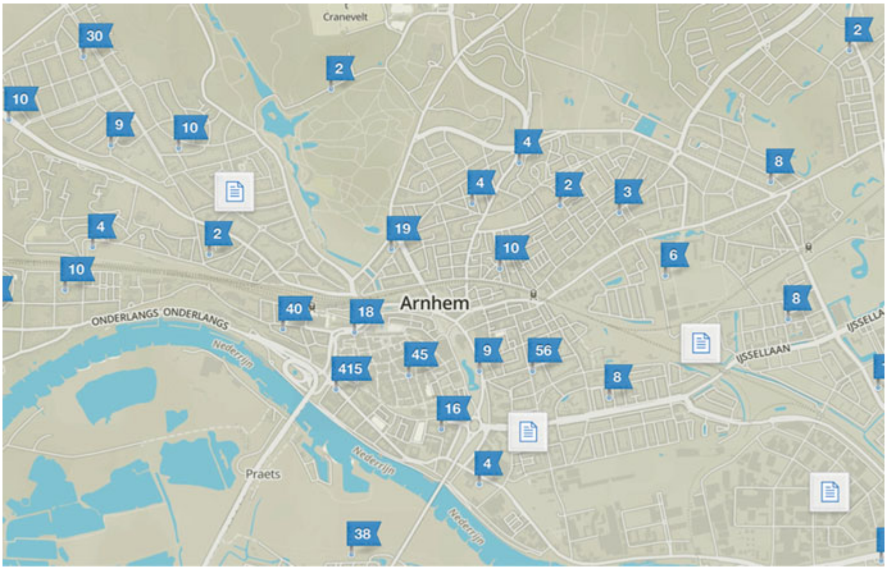
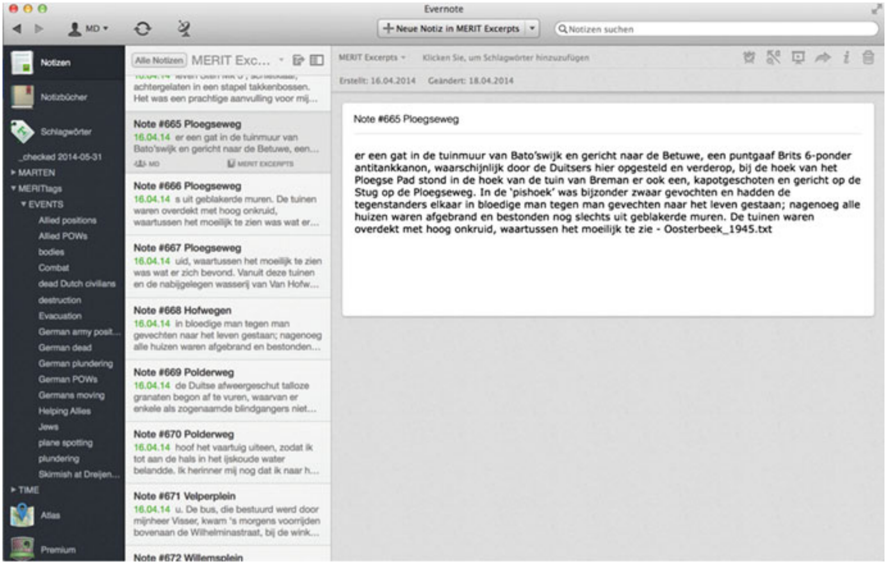


Fig. 4 Evernote's Atlas groups notes by location



achtergelaten in een stapel takkenbossen. Het was een prachtige aanvulling voor mij...

Note #665 Ploegseweg
16.04.14 er een gat in de tuinmuur van Bato'swijk en gericht naar de Betuwe, een...

Note #666 Ploegseweg
16.04.14 s uit geblakerde muren. De tuinen waren overdekt met hoog onkruid, waartussen het moeilijk te zien was wat er...

Note #667 Ploegseweg
16.04.14 uid, waartussen het moeilijk te zien was wat er zich bevond. Vanuit deze tuinen en de nabijgelegen wasserij van Van Hofw...

Note #668 Hofwegen
16.04.14 in bloedige man tegen man gevechten naar het leven gestaan; nagenoeg alle huizen waren afgebrand en bestonden...

Note #669 Polderweg
16.04.14 de Duitse afweergeschut talloze granaten begon af te vuren, waarvan er enkele als zogenaamde blindgangers niet...

Note #670 Polderweg
16.04.14 hoof het vaartuig uiteen, zodat ik tot aan de hals in het ijskoude water belandde. Ik herinner mij nog dat ik naar h...

Note #671 Velperplein
16.04.14 u. De bus, die bestuurd werd door mijnheer Visser, kwam 's morgens voorrijden bovenaan de Wilhelminastraat, bij de wink...

Note #672 Willemsplein

Note #665 Ploegseweg

er een gat in de tuinmuur van Bato'swijk en gericht naar de Betuwe, een puntgaaf Brits 6-ponder antitankkanon, waarschijnlijk door de Duitsers hier opgesteld en verderop, bij de hoek van het Ploegse Pad stond in de hoek van de tuin van Breman er ook een, kapotgeschoten en gericht op de Stug op de Ploegseweg. In de 'pishoek' was bijzonder zwaar gevochten en hadden de tegenstanders elkaar in bloedige man tegen man gevechten naar het leven gestaan; nagenoeg alle huizen waren afgebrand en bestonden nog slechts uit geblakerde muren. De tuinen waren overdekt met hoog onkruid, waartussen het moeilijk te zie - Oosterbeek_1945.txt

Fig. 5 Note view in Evernote with a rudimentary tagging system on the *left*, a list of notes in the *middle*, and the note text in the main viewing window

API, and shared notebooks. Note that given the modest aim for this study, we have skipped Step 5, Adjustment.

We rely on Evernote's Atlas functionality to plot and group the excerpts on a zoomable map. A map like this provides an immediate sense of where eyewitnesses experienced events and we can link different events based on geographical proximity. Again, for this small-scale scale proof-of-concept we relied on visual identification of relative proximity alone. Given that streets in the region differ considerably in length, we conducted a visual analysis of the map by means of which we can account for references to particularly long streets whose latitude/longitude value (typically in the middle) differs considerably from those of for example cross-streets a few kilometres down the road. We will consider linking by geographical radius in a future stage of the project.

This visualization-based linking of place names worked particularly well for "Dreijenseweg", a street leading northwards out of Oosterbeek, west of Arnhem, where German troops defended a hill against British troops. We learn from the linked police reports that after the Second World War the bodies of several British soldiers were discovered and exhumed.

18 September 1944—Civilian memoirs: German tanks sighted along road. (*translated*) Large German tanks were placed along the DREIJENSEWEG, and large amounts of supplies were lying around, which had been captured by Germans after they had been dropped from airplanes. I remember German soldiers being visibly pleased by the contents of the baskets and containers.

19 September 1944—Allied soldier describing the skirmish. Waddy was wounded during our advance through these woods. Enemy were using self-propelled 88 mm and six barrelled mortars. I was separated from Nick Nicholls about this time. I believe he was instructed to be bodyguard to General Hackett. Advanced through woods in area between Johannahoeve and DREIJENSEWEG, with Sgt Shepley, and three others of my company, I cannot recall their names; we were held up trying to cross a track. A SP 88 mm was firing down the track and an enemy sniper was firing from our front. We took cover behind a felled tree trunk trying to spot a sniper.

20 August 1945—Police Report. (*translated*) Ms. Jansen, living at AMSTERDAMSEWEG number 244 notifies us that an English soldier is buried in the ditch of the DREIJENSEWEG just beyond the corner of the Van Buuren home.

30 October 1948—Police Report. (*translated*) In the garden of the house at DREIJENSEWEG inhabited by De Gruijter, excavations have produced various human bones and remains of military gear. Judging from the covers found, one body is still in the ground. The gear is English.

4 November 1948—Police Report. (*translated*) Dutch and English body recovery services have exhumed three bodies of English soldiers from the garden of De Gruijter, DREIJENSEWEG, and have transferred the remains to the English military cemetery.

In this case, we were able to reconstruct an event from multiple perspectives. This sense of spatial co-occurrence of events strikes us as valuable independently of the multi-perspective reconstruction of events that we are looking for.

6 Conclusion

In this paper we have outlined a workflow which describes our first findings from the research project MERIT—Machine-based Extraction of Relations in Text. MERIT is informed by the practice of historical source analysis and aids researchers with the task of identifying related source documents. We worked with 83 texts concerned with the Battle of Arnhem in 1944 which we linked by mentions of space using regular expressions rather than pre-trained Named Entity recognition method. As the former method is focused on the right granularity level, streets, we are able to find linked documents describing specific wartime events at the street level, and their aftermath.

Arguably, streets are the right level of granularity as they represent on the one hand the limited outlook that civilians have when war arrives at their doorstep, trapping them in their homes, while on the other hand they are commonly found in primary sources produced by the military as a means to recount their actions. Such references to places act as a linking element between otherwise unconnected perspectives on the same historical events. Our approach to organize sources micro-spatially has revealed the spatial proximity of individual events and has provided us with a geographical map and a network visualization which indicates source similarities based on co-occurring street names.

Our case study exemplifies a model of a workflow which integrates the specific strengths of historians with those of machines. At the fringe of the historian-computer interface our case study shows that at this moment we cannot simply rely on off-the-shelf NLP modules, as their preset categories do not capture the correct level of granularity we require. In order for our workflow to work smoothly some amount of knowledge is required, in this case of applying regular expressions to text. The text analytics toolbox that we foresee should therefore support advanced types of search such as regular expression search.

Acknowledgements The authors wish to thank the anonymous reviewers for their comments and suggestions. We are grateful to Berry de Reus en Jan Hovers of the Airborne Museum 'Hartenstein', Oosterbeek, for their willingness to give us access to their museum's digital resources.

References

1. Bennett D (2008) Magnificent disaster: the failure of market garden, the arnhem operation. Casemate, Oxford (Sept 1944)
2. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30(1):107–117
3. Byrne K, Klein E (2010) Automatic extraction of archaeological events from text. In: Proceedings of computer applications and quantitative methods in archaeology, Williamsburg
4. Cornelius R (1995) A bridge too far, 1st edn. Simon & Schuster, New York

5. Diesner J, Carley KM, Tambayong L (2012) Extracting socio-cultural networks of the sudan from open-source, large-scale text data. Comput Math Organ Theory 18(3):328–339. doi:10.1007/s10588-012-9126-x. http://www.link.springer.com/10.1007/s10588-012-9126-x
6. Düring M (2012) Verdeckte soziale netzwerke im nationalsozialismus. die Entstehung und Arbeitsweise von Berliner Hilfsnetzwerken für verfolgte Juden. Ph.D. thesis, Universität Mainz, Mainz
7. Düring M (2014) Netzwerkvisualisierungen in den geschichtswissenschaften zwischen explorativer quellenanalyse und der suggestionskraft des bildes. In: Haussling R (ed) Visualisierung sozialer netzwerke. VS Verlag für Sozialwissenschaften, Wiesbaden
8. Gerritsen S, Lenders W (2006) Verhalen die blijven: beleefde geschiedenis in de grensregio. Nationaal bevrijdingsmuseum 1944–1945, Groesbeek, the Netherlands
9. Hall D, Jurafsky D, Manning C (2008) Studying the history of ideas using topic models. In: Proceedings of the conference on empirical methods in natural language processing. ACL, Stroudsburg, pp 363–371
10. Hendrickx I, Düring M, Zervanou K, Van Den Bosch A (2013) Searching and finding strikes in the New York Times. In: Proceedings of the 3rd workshop on annotation of corpora for research in the humanities (ACRH-3). The Institute of Information and Communication Technologies, Bulgarian Academy of Scienes, Sofia, pp 25–36
11. Henke KD (1996) Die amerikanische Besetzung Deutschlands. Oldenbourg Verlag, Munich
12. Hunter J, Macarthur J, der Plaat DV, Gosseye J, Muys A, Macnamara C, G Bannerman G (2014) Extracting relationships from an online digital archive about post-war queensland architecture. http://www.dharchive.org/paper/DH2014/Paper-826.xml
13. Isaksen L (2014) Pelagios: pelagios 3 overview. http://www.pelagios-project.blogspot.be/2013/09/pelagios-3-overview.html
14. Jockers ML (2013) Macroanalysis: digital methods and literary history. University of Illinois Press, Champaign
15. Karsdorp F, Van den Bosch A (2013) Identifying motifs in folktales using topic models. In: Proceedings of the 22 annual Belgian-Dutch conference on machine learning, Nijmegen, pp 41–49
16. Kershaw R (1990) It never snows in september: the German view of market-garden and the battle of Arnhem, september 1944. Crowood, Marlborough
17. McCallum AK (2002) Mallet: a machine learning for language toolkit. http://www..mallet.cs.umass.edu
18. Meister JC, Jacke J (2014) Pushing back the boundary of interpretation: concept, practice and relevance of a digital heuristic. http://www.dharchive.org/paper/DH2014/Paper-76.xml
19. Miller B, Shrestha A, Derby J, Olive J, Umapathy K, Li F, Zhao Y (2013) Digging into human rights violations: data modelling and collective memory. In: 2013 IEEE international conference on big data, pp 37–45. doi:10.1109/BigData.2013.6691668
20. Miller B, Shrestha A, Olive J (2014) Visualizing computational, transversal narratives from the world trade towers. http://www.dharchive.org/paper/DH2014/Paper-660.xml
21. Nuessli MA, Kaplan F (2014) Encoding metaknowledge for historical databases. http://www.dharchive.org/paper/DH2014/Paper-366.xml
22. Shrestha A, Miller B, Zhu Y, Zhao Y (2013) Storygraph: extracting patterns from spatio-temporal data. In: Proceedings of the ACM SIGKDD workshop on interactive data exploration and analytics, IDEA ’13. ACM, New York, pp 95–103. doi:10.1145/2501511.2501525. http://www.doi.acm.org/10.1145/2501511.2501525
23. Sporleder C (2010) Natural language processing for cultural heritage domains. Lang Linguist Compass 4(9):750–768
24. Sporleder C, Van Erp M, Porcelijn T, Van den Bosch A (2006) Identifying named entities in text databases from the natural history domain. In: Proceedings of the 5th international conference on language resources and evaluation, LREC-2006, Trento
25. Stacey CP (1960) Official history of the Canadian army in the second world war: the victory campaign: the operations in Northwest Europe, 1944–1945. Official history of the Canadian

army, vol 3. Queen's Printer, Ottawa. http://www.cmp-cpm.forces.gc.ca/dhh-dhp/his/docs/Victory_e.pdf

26. Underwood T, Black ML, Auvil L, Capitanu B (2013) Mapping mutable genres in structurally complex volumes. In: Proceedings of the 2013 IEEE international conference on big data. IEEE, Santa Clara, pp 95–103
27. Van de Camp M, Van den Bosch A (2012) The socialist network. Decision Support Syst 53(4):761–769
28. Van den Bosch A, Sporleder C, Van Erp M, Hunt S (2007) Automatic techniques for generating and correcting cultural heritage collection metadata. In: Proceedings of digital humanities 2007, the 19th joint international conference of the association for computers and the humanities and the association for literary and linguistic computing. University of Illinois at Urbana-Champaign, Champaign, pp 223–224
29. Van den Bosch A, Stroppa N, Way A (2007) A memory-based classification approach to marker-based EBMT. In: Eynde FV, Vandeghinste V, Schuurman I (eds) Proceedings of the METIS-II workshop on new approaches to machine translation, Leuven, pp 63–72
30. Van den Bosch A, Lendvai P, Van Erp M, Hunt S, Van der Meij M, Dekker R (2009) Weaving a new fabric of natural history. Interdisciplinary Sci Rev 34(2–3):206–23
31. Van den Hoven M, Van den Bosch A, Zervanou K (2010) Beyond reported history: strikes that never happened. In: Darányi S, Lendvai P (eds) Proceedings of the first international AMICUS workshop on automated motif discovery in cultural heritage and scientific communication texts, Vienna, pp 20–28

Towards a Historical Text Re-use Detection

Marco Büchler, Philip R. Burns, Martin Müller, Emily Franzini, and Greta Franzini

Abstract *Text re-use* describes the spoken and written repetition of information. *Historical text re-use*, with its longer time span, embraces a larger set of morphological, linguistic, syntactic, semantic and copying variations, thus adding a complication to *text-reuse* detection. Furthermore, it increases the chances of redundancy in a Digital Library. In *Natural Language Processing* it is crucial to remove these redundancies before applying any kind of machine learning techniques to the text. In Humanities, these redundancies foreground textual criticism and allow scholars to identify lines of transmission. This chapter investigates two aspects of the historical *text re-use* detection process, based on seven English editions of the *Holy Bible*. First, we measure the performance of several techniques. For this purpose, when considering a verse—such as book Genesis, Chapter 1, Verse 1—that is present in two editions, one verse is always understood as a paraphrase of the other. It is worth noting that *paraphrasing* is considered a hyponym of text re-use. Depending on the intention with which the new version was created, verses tend to differ significantly in the wording, but not in the meaning. Secondly, this chapter explains and evaluates a way of extracting paradigmatic relations. However, as regards historical languages, there is a lack of language resources (for example, WordNet) that makes non-literal text re-use and paraphrases much more difficult to identify. These differences are present in the form of replacements, corrections, varying writing styles, etc. For this reason, we introduce both the aforementioned

M. Büchler (✉)
Göttingen Centre for Digital Humanities, Georg August University Göttingen, Papendiek 16, 37073 Göttingen, Germany
e-mail: mbuechler@gcdh.de

P.R. Burns
Academic and Research Technologies, Northwestern University, Evanston, IL, USA
e-mail: pib@northwestern.edu

M. Müller
Department of English, Northwestern University, Evanston, IL, USA
e-mail: martinmueller@northwestern.edu

E. Franzini • G. Franzini
Digital Humanities Chair, Department of Computer Science, Augustusplatz 10/11, 04009 Leipzig, Germany
e-mail: efranzini@informatik.uni-leipzig.de; franzini@informatik.uni-leipzig.de

C. Biemann, A. Mehler (eds.), *Text Mining*, Theory and Applications of Natural Language Processing, DOI 10.1007/978-3-319-12655-5_11

and other correlated steps as a method to identify text re-use, including language acquisition to detect changes that we call *paradigmatic relations*. The chapter concludes with the recommendation to move from a "single run" detection to an iterative process by using the acquired relations to run a new task.

1 Introduction

In the last few years, research methodologies in the Humanities have significantly changed. As recently as 30 years ago, access restrictions to libraries posed numerous challenges to humanists working with printed books and manuscripts. Today, the efforts of mass digitisation provide broader access to these items in digital form [12, 37]. The increasing availability of digitally encoded texts expedites and facilitates the exploration of text patterns. Google's mass digitisation efforts, for example, is driving the improvement of close reading methods. The question "*What do you do with a million books?*" [19] did not only kickstart the *Digging into Data*[1] programme,[2] but also addressed the potential use of distant reading methods in virtually any form of text mining.

This chapter offers a contribution to the detection of paraphrased relationships between text passages. The key issue is how to define "paraphrase". Definitions range from simple rewording to full synonymic replacement of all words in a text passage. The latter is of particular concern since in a corpus of a million books, a quantitative mining algorithm might link uncorrelated text passages to each other.

The set up of an evaluation basis for text re-use, and especially paraphrasing, has already been attempted a few times. The *Microsoft Research Paraphrasing Corpus*[3] has compiled 5,801 paraphrases [8]. The PAN challenge [39] provides a test bed of plagiarism cases of up to 30,000 examples. Nevertheless, we decided to conduct our research using seven English translations of the Holy Bible (cf. Sect. 2) that all have one common origin. We selected the Bible for several reasons. One is that a verse in any given Bible translation should contain the same information as the same verse in any other translation. The Bible has also been translated into English many times across the centuries, providing a basis for investigating changes in the English language over time, including changes in the form, spelling, and meaning of words. This property makes it possible to link verses by their identifiers, such as *Book Genesis, Chapter 1, Verse 1*, pairwise across two or more editions. By doing so, we easily create a testbed of almost 600,000 reliable links—some orders of magnitude larger that what can be achieved by other testbeds. This method is applicable to make more Bible editions and need not stop at the seven editions chosen for this chapter. Mayer and Cysouw [34] compiled 52 English Bible editions that can be

[1] http://www.diggingintodata.org/.

[2] Supported by the National Endowment for the Humanities (NEH)http://ww.neh.gov/.

[3] http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/.

linked to each other in the same way.[4] Another online resource[5] comprises 1024 Bible editions in 670 languages, making this method of evaluation easily available for research in languages other than English. Besides this, the Bible also contains a lot of inner text re-use between e.g. the Apostle Mark and Luke [32]. This obtains us to check stability of text re-use within one edition compared to other editions.

As the Bible has been translated into English several times from Hebrew, Latin or Ancient Greek, paraphrasing occurs frequently. In addition, the manner in which the original material has been paraphrased certainly reflects the personal style and interest of the translator and his proximity to the original. For example, we find that the *King James Version* translation of the sixteenth century exhibits the wide variation in orthography and word choice characteristic of *Early Modern English*, whereas the *Bible in Basic English* (Middle of twentieth century) exhibits significantly less of this variation. Those variations are *could* vs. *couldeth* or word forms like *saith* or *sayeth*. Another form of variation is illustrated by *Young's Literal Translation* (completed in 1862) which, generally speaking, does not follow the syntax of English but that of Hebrew instead. In 1833, Noah Webster revised the *King James Version* by removing all archaic word forms and modernised this version in the style of the nineteenth century.

Our chapter seeks answers to two complementary and correlated research questions. First, how well do different *text re-use* detection techniques work to link the same verse in different translations of the Bible to one another? Second, as the words in a verse in the different Bible translations are not usually exact duplicates, can we mine and investigate these differences within a verse? This chapter focuses on the mining of paradigmatic relations as shown in the following example from Genesis, Chapter 1, Verse 1:

ASV	In the beginning God **created** the heavens and the earth.
Basic English	At the first God **made** the heaven and the earth.
Darby	In the beginning God **created** the heavens and the earth.
KJV	In the beginning God **created** the heaven and the earth.
Webster	In the beginning God **created** the heaven and the earth.
WEB	In the beginning God **created** the heavens and the earth.
YLT	In the beginning of God's **preparing** the heavens and the earth.

In this example *create*, *make*, and *prepare* are mined as a paradigmatic relationship. The key question behind the second task is to find out what we can learn from historical data about the usage of words in a paradigmatic relationship—e.g. as historical synonyms—even if these words might not be considered synonyms in contemporary usage.

[4] www.paralleltext.info.

[5] www.mysword.info.

Table 1 Basic properties of all seven Bible translations

Bible version	Word tokens	Word types	Token/type ratio
ASV	741,267	13,485	54.97
Basic	791,367	7,350	100.85
Darby	732,928	14,971	48.96
KJV	746,746	13,466	55.45
WEB	722,817	13,556	54.68
Webster	744,137	13,655	54.50
YLT	745,422	13,973	53.34

2 Data: Investigated Corpus and Initial Setup

We selected seven English translations of the Bible [3]:

- American Standard Version (ASV)
- Bible in Basic English (Basic)
- Darby Bible (Darby)
- King James Version (KJV)
- World English Bible (WEB)
- Webster Bible (Webster)
- Young's Literal Translation (YLT)

Each translation of the Bible contains books in common with all other translations such as *Genesis*, *Mark* or *Luke* as well as texts that do not appear in all translations such as *Baruch*, *Ecclesiasticus*, and other deuterocanonical texts. We have chosen to extract only the $28,632$ verses which appear in all seven translations, as we require parallel text from all Bible translations for our task.

Table 1 displays some basic frequency-based properties of the words in each translation such as the total number of words and the number of unique words. The difference between the number of unique word types in the Bible in *Basic English* and all the other translations is especially striking.

3 Related Work

The process of detecting text re-use comprises various applications, such as information retrieval, plagiarism, text summarisation, question answering, and text decontamination. Text decontamination is the use of duplicate detection to remove passages that are similar. The reason this is done is to allow for further machine learning methods to be applied to the text at a later stage. Independently from the source of the application, all researchers can compare their approaches to a test bed compiled for the annual PAN challenge [39]. Together with these monolingual approaches, [6] reports on aligning passages called 'text-fragment'—in a multi-lingual domain by *divergence from randomness*.

Zesch et al. [44] makes a distinction between *content*, *structure*, and *style* of a text that can be re-used. While *content* is analysed by similarities on a word level, structure is measured by ngrams. *Token-type-ratio*, as well as length of sentences and tokens, are the *style* features.

Brück et al. [10] goes beyond the level of surface features, such as words or ngrams. A semantic network is created, and derived by a syntactic-semantic parser. This approach also allows for the detection of paraphrases through the application of inferences by lexico-semantic relations or the meaning postulated to this network.

The top of aligning one text to another with the intention of deriving rewriting rules is explored in [22]. The authors propose a four step approach: (1) the induction of the topical structure by clustering, (2) learning about structural mapping rules, (3) the macro alignment of mapping paragraphs to their topics by including the learned structural rules, and (4) a local micro alignment at a sentence level.

4 Algorithms: Text Re-use Techniques

What is *text re-use*? *Text re-use* concerns the recycling of complex textual information which contains a statistically significant number of words in common between a source document and a target document. This contrasts with *Topic Detection and Tracking* [1] which instead deals with small units of text such as single words or multiword units.

Brute Force techniques of linking every *re-use unit* to each other only work on small data. For this reason, recent approaches adopt *Feature-Based Linking (FBL)* strategies [4, 20, 21, 25]. While the *Brute Force* method checks all entries A_{ij} in an adjacency matrix A, the *FBL* only checks those entries in A which have one or more features in common. The latter method can significantly reduce performance by both ignoring function words such as *and* or *the* and by comparing those $(v_i, v_j) \in E$ that have valuable re-use candidates.

Text re-use techniques create a re-use graph $G = (V, E)$ consisting of a set V of vertices and a set $E = V \times V$ of edges between elements of V. The set V contains large linguistic units such as verses, sentences, or paragraphs.

The question of how best to characterize a *re-use unit* like a sentence or a verse with features depends largely on the type of text. For example, syntactic features work well for philosophic texts. On the other hand, re-use in historiography is much more distant from the original. With this type of text the intent and interest of the author reusing the earlier text is important.

Both the complexity issue and the fact that different kinds of texts demand different featuring techniques informed our choice to break up the monolithic algorithms typically used in this research area. We designed a simple, general seven step architecture for *text re-use*. The Java based *TRACER* tool implements this approach [11].

Historical texts often contain a large number of spelling mistakes, linguistic variations, dialectal variations or scribal errors. For this reason, we may understand the process of text re-use detection as an instance of a *Locality Sensitive Hashing (LSH) h* [16] (cf. Eq. 1). Unlike *md5* or *crc32*, an *LSH* hash function does not aim at flipping 50 % of all output bits if one input bit changes, but at mapping similar inputs to the same, or at least similar, representation. This is shown in Eq. 1:

$$\mathbf{Pr}_{h\in\mathscr{F}}[h(x)=h(y)]=sim(x,y), \tag{1}$$

where $sim(x, y)$ is the *Min-wise Independent Permutation* [17] computed by Eq. 2. A and B represent the sets of features of the two *re-use units* v_i and v_j.

$$sim(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{2}$$

Like the *Brute Force* method, the feature-based linking *fbl* is also of squared complexity. While the *BFL* depends on the number of elements in V, the *FBL* depends on the frequency f of all features. The linking costs for *fbl* can be computed by Eq. 3.

$$fbl=\sum|f_i|\cdot\left(|f_i|-1\right) \tag{3}$$

Zipf's Law [45] as a power law on word-type distribution states that only function words appear frequently in a text, whereas roughly half of all words occur only once. This fact can be employed for *max pruning* and *min pruning* [11]. *Max pruning* removes all frequent features such as function words. Even if it includes only a few hundred words, it significantly speeds up performance. *Min pruning* of rare features occurring only once helps reduce the feature index and often makes data structures faster. Features with a frequency of 1 can be ignored for text re-use detection since re-use implies that a feature needs to occur at least twice.

A *Locality Sensitive Hashing* function *h* includes and needs, on the basis of the aforementioned *re-use diversity*, numerous parameters. We thus propose to break *LSH* for the text re-use function down into the seven levels or steps: *segmentation*, *pre-processing*, *featuring*, *selection*, *linking*, *scoring*, and an optional *post-processing*.

A digital library can be processed by the *overlapping* or *disjoint segmentation*. An *overlapping segmentation* is beneficial whenever we wish to detect text re-use on small sentence sections. An overlapping segmentation test on the Perseus Digital Library revealed that 80 % of the identified text re-uses contained four or less common words. *Disjoint segmentation*, a commonly used method, splits re-use units into sentences [29], paragraphs [20] or even documents [18]. The size of the re-use units strongly depends on the research question and the underlying task.

The pre-processing function is one of the most complex. The objective of this function is to form equivalence classes of similar words, which [26] calls *concepts*. Stein maps different techniques to these conceptual classes [42]. Richly structured lexical resources such as WordNet [23, 35], support the use of linguistic and semantic relations, including synonyms or co-hyponyms. Automatic methods of forming semantically similar concept classes are known as *pLSA* [27] and *Topic Modeling* [5]. Bordag describes the usage of word *co-occurrence* profiles as a means of comparing their semantic similarities [7]. The underlying *distributional hypothesis* [24] assumes that words being used in similar contexts are semantically close.

Historical documents heavily fraught with linguistic variation lend themselves well to string processing techniques, including the *Levenshtein distance* and the *FastSS* approach [30, 33], both of which can also be applied to noisy data such as OCR output.

The featuring function transforms the re-use unit into features that can be compared, like *word-types* [15, 29] and *n-grams* [18]. The *word-type feature* technique can be deployed to detect paraphrase or allusion, whereas the *n-gram* approach identifies duplicates and near-duplicates. *N-gram* techniques can be further divided into *shingling* and *hash-breaking* featuring methods [11].

The selection function seeks to remove from the digital fingerprint features that are considered to be irrelevant. Alongside the *min* and *max pruning*, selection can also be achieved by adapting information retrieval techniques such as the *tf.idf* measure of term-weighting [40]. Talavera [38] describes another approach that involves feature dependencies, which occasionally lead to clusters. As per Schleimer's recommendation [43], clusters can be avoided by virtue of the winnowing algorithm, which selects features all over the re-use unit rather than solely over a local cluster. This latter avenue, however, is best suited to larger re-use units such as *pages* or *books*.

There are two classes of linking functions. The *Intra Digital Library Linking* [11] maps re-use units to other units within the same database, leading to a graph $G = (V, E)$. The advantage of the *Intra Digital Library Linking* is that the feature index can be compressed by removing all features with a frequency of 1 as these do not contribute to the process. The *Inter Digital Library Linking* [11] detects re-use between at least two different textual databases resulting in a bipartite graph $G = (V_1, V_2, E)$. Features with a frequency of 1 cannot be removed from the feature index owing to the fact that the same feature could appear in the other database or digital library.

The scoring function investigates the *re-use overlap* of two linked re-use units. The *vector space model* [40] considers not only the size of the overlap but also the weight of its features [2, 4, 29]. Broder [9] reports that easier methods such as *resemblance* and *containment* already provide commensurable results. Following these latter methods, all existing features of an overlap have the same weight (or are weightless). This indicates that the output of the previous processed feature selection function has a higher influence on the scoring. A broader overview of scoring metrics is provided in [11].

The optional post-processing function operates on top of the re-use graph $G = (V, E)$, eliminating, for instance, noisy links. Post-processing is advantageous if the research intent is to identify single links or clusters of links that indicate a passage has been copied over from one work to the other. For these use-cases, one can algorithmically pinpoint linear sequences in the manner of dot plot view visualisations [13, 32].

5 Initial Setup

We segmented all seven Bible translations verse by verse. That gave us 28, 632 verses per translation and 200, 424 verses in all.

From the around one million algorithmic permutations of the TRACER tool, we select twelve combinations for the evaluation task of *text re-use* techniques for detecting paraphrases and one out of these for the task of extracting paradigmatic relations.

Step 1: We perform four different pre-processing stages:

- *Base*: In the pre-processing *Base*, we map all uppercase letters to their lowercase equivalents. This step is necessary since e.g. God is written as GOD as well. This mapping is used for all other pre-processing stages as well.
- *StringSim*: As described in Sect. 4, a string similarity word graph is computed based upon letter bigrams. Given all seven Bible translations, we compute 83,866 non-directed links between similar written words.
- *Lemma*: We compute lemmatization data using the morphy function of WordNet as provided by the Natural Language Toolkit [31]. We determine 7,479 lemmatization mappings based upon the word types found in all translations. We use the observed word variant when the base form cannot be determined.
- *Lemma+Syn*: In another setup, not only is the text lemmatized but also the base forms are replaced by synonyms. Using WordNet [23] we extract 33,074 synonymic relations of nouns, verbs, and adjectives that are relevant to the words of all Bible translations.

Neither the string similarity data nor the synonym data form directed word graphs. That means these data are quite difficult to handle for pre-processing since we need to decide which variant W is to be replaced by W'. For this reason we use the word frequency to weight the nodes (words) in the string similarity graph and synonym graph. We then use this weighting information to generate a directed graph that points from the lower frequency to the higher frequency word.

Step 2: For the task of evaluating how well the *text re-use* techniques detect paraphrases, we select trigram and bigram shingling as well as unigram fingerprinting as shown in Table 2.

As feature selection strategy (step 3), we use max pruning (cf. Sect. 4) with a feature density of 0.8. Feature density is the relative ratio between the selected feature and all found features. Assuming that a verse has twenty features, then we select the sixteen least frequently occurring features. A feature density of 0.8 is a very passive value. In recent experiments performed in tandem to those reported here it could be shown that even a feature density between 0.5 and 0.6 works quite well and the computation speed increases significantly. We choose 0.8 to achieve a

Table 2 Settings S_i for all 12 experiments by combining 4 pre-processing steps (*L1*) and 3 fingerprinting techniques(*L2*)

	Featuring		
Preprocessing	Trigram	Bigram	Word
Base	S_{01}	S_{05}	S_{09}
StringSim	S_{02}	S_{06}	S_{10}
Lemma	S_{03}	S_{07}	S_{11}
Lemma+Syn	S_{04}	S_{08}	S_{12}

better recall; runtime was considered less important for these experiments. On the scoring level (step 5), Broder's resemblance metric [9], which ranges between 0.0 and 1.0, is computed with a threshold of 0.7.

For this chapter's second task, computing paradigmatic relations, the setting S_{09} is selected. In Sect. 6 (Results), it is shown that unigram fingerprinting works significantly better than trigram or bigram shingling. Furthermore, for this task we use text which has been preprocessed less so that different inflected variants and synonyms can be extracted.

6 Results

We can easily evaluate both the task of evaluating *text re-use* techniques and that of extracting paradigmatic relations. Since we selected only the 28,632 verses that are contained in all seven Bible translations, we can map a reference such as Genesis, Chapter 1, Verse 1 as it appears in one translation to the same verse in any other translation. This allows us to pairwise link and evaluate the same verse in different translations.

The second task of extracting paradigmatic relations from the pairwise links uses the synonym data provided by WordNet (cf. introductory example). We argue against the view, shared by a number of scholars, that paraphrasing is a simple replacement of words by synonyms. Instead we suggest that paraphrasing is a much more complex process. For this reason we understand our contribution offers just one step towards 'real' paraphrase detection from a humanities perspective.

6.1 Evaluation of Text Re-use Techniques for Paraphrase Detection

We set up the aforementioned twelve independent user testings and compared all seven Bible translations pairwise. All 28,632 paired verses (almost 600,000 overall) are used to compute the recall. It must be noted that in this chapter we focus primarily on *recall* over *precision*, since humanities scholars tend to rank this higher. The precision of the data can be, in this case, considered as less essential than providing scholars with good candidates. It is for the same reason that we neglect *F-Measure*. Some results that can be deduced from Table 3 include:

1. *Relationship between KJV and Webster*: Both underlined rows in Table 3 represent the recall for all twelve experiments. The results are quite good regardless of the choice of fingerprinting technique and pre-processing. Webster produced his Bible translation in 1833 based upon the *KJV*. However, he corrected grammar and removed archaic words.
2. *YLT and Basic*: Of greater interest are the *YLT* (syntactically close to Hebrew) and the *Basic* (rewording in Basic English) translation. For both of these translations the trigram and bigram shingling results are significantly worse than for the other Bible translations. However, the recall values for the *YLT* with unigram fingerprinting significantly increase by a bit more than an additional 0.5. Representing a fingerprint as a bag of words all syntactical dependencies are ignored with unigram fingerprinting. For this reason, the significant boost by switching from n-gram shingling to unigram fingerprinting is completely reliable. The recall on the *Basic* translation, using unigram fingerprinting remains between 0.46 and 0.68 (average of all Bible translations compared to the Basic translation is 0.58) even if the Bible is lemmatized as well as synonyms are used. The *YLT* translation has an average recall over all Bibles of 0.76. On all other Bible translations, at best an average recall between 0.85 and 0.88 is reached.

In this chapter we focus on pre-processing and fingerprinting for the task of paraphrase detection. Comparing pairwise *StringSim* and *Lemma* columns with each other for all three fingerprinting techniques in Table 3, we observe that both pre-processing steps lead to almost the same results. Intuitively, this seems unexpected. A more detailed examination of the data shows this result is reliable since e.g. *heaven* and *heavens* are used (cf. the introductory example). In addition to the large overlap between the string similarity and lemmatization data, the benefit of lemmatization is the ability to deal with irregular verbs like *to be* or *to have*. By contrast, string similarity catches the historical orthographic variants observed over the centuries such as *believedst* vs. *believest*, *gibeah* vs. *gibeath*, *galilaeans* vs. *galileans*, or *gishpa* vs. *gispa*. Here, we used 7,479 lemmatization mappings as well as 83,866 string similarity relations on word type level (cf. Sect. 5). At the token level, however, both types of data cover about the same number of words. Lemmatization works well for frequently occurring words, while string similarity works better in mapping words to each other for the less frequent words in a corpus.

Table 4 aggregates the recall of Table 3 to average values so that we can compare in pairwise fashion all the pre-processing steps with all the fingerprinting techniques. It is easy to see that on the fingerprinting level, unigram fingerprinting works significantly better than n-gram shingling.

One result that emerges from this research is that paraphrasing is not just about rewording or replacing words by synonyms. For other Bible editions, on the other hand, a recall of 0.8 (or higher) can be computed, whereas the heavily paraphrased *Bible in Basic English* the recall is worse, at less than 0.6 (cf. Table 3). This led us

Table 3 Recall of all 12 experiments (cf. Sect. 5) on a total of 21 different pairwise Bible comparisons

	Trigram shingling				Bigram shingling				Word based featuring			
	S_{01}	S_{02}	S_{03}	S_{04}	S_{05}	S_{06}	S_{07}	S_{08}	S_{09}	S_{10}	S_{11}	S_{12}
ASV vs. Basic	0.035	0.037	0.038	0.043	0.096	0.100	0.106	0.126	0.506	0.534	0.561	0.607
ASV vs. Darby	0.421	0.441	0.441	0.458	0.649	0.667	0.673	0.684	0.932	0.942	0.945	0.949
ASV vs. KJV	0.665	0.686	0.683	0.695	0.830	0.845	0.842	0.847	0.958	0.965	0.963	0.965
ASV vs. WEB	0.604	0.623	0.612	0.617	0.786	0.798	0.788	0.788	0.942	0.951	0.948	0.951
ASV vs. Webster	0.593	0.615	0.613	0.624	0.795	0.811	0.809	0.815	0.954	0.961	0.960	0.961
ASV vs. YLT	0.052	0.060	0.065	0.072	0.184	0.208	0.235	0.252	0.764	0.804	0.838	0.850
Basic vs. ASV	*0.035*	*0.037*	*0.038*	*0.043*	*0.096*	*0.100*	*0.106*	*0.126*	*0.506*	*0.534*	*0.561*	*0.607*
Basic vs. Darby	*0.023*	*0.025*	*0.025*	*0.028*	*0.067*	*0.073*	*0.074*	*0.088*	*0.468*	*0.496*	*0.534*	*0.577*
Basic vs. KJV	*0.022*	*0.025*	*0.024*	*0.028*	*0.065*	*0.073*	*0.076*	*0.094*	*0.484*	*0.513*	*0.545*	*0.587*
Basic vs. WEB	*0.039*	*0.041*	*0.041*	*0.049*	*0.115*	*0.120*	*0.125*	*0.152*	*0.594*	*0.615*	*0.650*	*0.689*
Basic vs. Webster	*0.025*	*0.028*	*0.026*	*0.032*	*0.076*	*0.083*	*0.086*	*0.108*	*0.505*	*0.532*	*0.564*	*0.611*
Basic vs. YLT	*0.008*	*0.009*	*0.009*	*0.010*	*0.024*	*0.027*	*0.028*	*0.032*	*0.305*	*0.335*	*0.407*	*0.462*
Darby vs. ASV	0.421	0.441	0.441	0.458	0.649	0.667	0.673	0.684	0.932	0.942	0.945	0.949
Darby vs. Basic	0.023	0.025	0.025	0.028	0.067	0.073	0.074	0.088	0.468	0.496	0.534	0.577
Darby vs. KJV	0.376	0.394	0.395	0.410	0.609	0.627	0.630	0.637	0.916	0.929	0.929	0.934
Darby vs. WEB	0.257	0.275	0.271	0.287	0.500	0.520	0.519	0.527	0.867	0.885	0.888	0.896
Darby vs. Webster	0.371	0.391	0.393	0.410	0.613	0.631	0.632	0.643	0.920	0.931	0.930	0.936
Darby vs. YLT	0.053	0.060	0.068	0.075	0.181	0.206	0.235	0.254	0.764	0.802	0.839	0.852
KJV vs. ASV	0.665	0.686	0.683	0.695	0.830	0.845	0.842	0.847	0.958	0.965	0.963	0.965
KJV vs. Basic	0.022	0.025	0.024	0.028	0.065	0.073	0.076	0.094	0.484	0.513	0.545	0.587
KJV vs. Darby	0.376	0.394	0.395	0.410	0.609	0.627	0.630	0.637	0.916	0.929	0.929	0.934
KJV vs. WEB	0.324	0.349	0.341	0.353	0.570	0.594	0.583	0.590	0.878	0.897	0.891	0.897
KJV vs. Webster	0.895	0.910	0.904	0.910	0.962	0.969	0.964	0.965	0.990	**0.993**	0.990	0.991

(continued)

Table 3 (continued)

	Trigram shingling				Bigram shingling				Word based featuring			
	S_{01}	S_{02}	S_{03}	S_{04}	S_{05}	S_{06}	S_{07}	S_{08}	S_{09}	S_{10}	S_{11}	S_{12}
KJV vs. YLT	0.043	0.049	0.054	0.059	0.155	0.179	0.200	0.211	0.731	0.776	0.814	0.827
WEB vs. ASV	0.604	0.623	0.612	0.617	0.786	0.798	0.788	0.788	0.942	0.951	0.948	0.951
WEB vs. Basic	0.039	0.041	0.041	0.049	0.115	0.120	0.125	0.152	0.594	0.615	0.650	0.689
WEB vs. Darby	0.257	0.275	0.271	0.287	0.500	0.520	0.519	0.527	0.867	0.885	0.888	0.896
WEB vs. KJV	0.324	0.349	0.341	0.353	0.570	0.594	0.583	0.590	0.878	0.897	0.891	0.897
WEB vs. Webster	0.351	0.376	0.367	0.376	0.596	0.620	0.608	0.614	0.892	0.908	0.903	0.908
WEB vs. YLT	0.033	0.038	0.043	0.047	0.116	0.135	0.156	0.172	0.643	0.689	0.750	0.770
Webster vs. ASV	0.593	0.615	0.613	0.624	0.795	0.811	0.809	0.815	0.954	0.961	0.960	0.961
Webster vs. Basic	0.025	0.028	0.026	0.032	0.076	0.083	0.086	0.108	0.505	0.532	0.564	0.611
Webster vs. Darby	0.371	0.391	0.393	0.410	0.613	0.631	0.632	0.643	0.920	0.931	0.930	0.936
Webster vs. KJV	0.895	0.910	0.904	0.910	0.962	0.969	0.964	0.965	0.990	**0.993**	0.990	0.991
Webster vs. WEB	0.351	0.376	0.367	0.376	0.596	0.620	0.608	0.614	0.892	0.908	0.903	0.908
Webster vs. YLT	0.047	0.052	0.059	0.065	0.164	0.187	0.214	0.227	0.736	0.777	0.817	0.834
YLT vs. ASV	*0.052*	*0.060*	*0.065*	*0.072*	*0.184*	*0.208*	*0.235*	*0.252*	*0.764*	*0.804*	*0.838*	*0.850*
YLT vs. Basic	*0.008*	*0.009*	*0.009*	*0.010*	*0.024*	*0.027*	*0.028*	*0.032*	*0.305*	*0.335*	*0.407*	*0.462*
YLT vs. Darby	*0.053*	*0.060*	*0.068*	*0.075*	*0.181*	*0.206*	*0.235*	*0.254*	*0.764*	*0.802*	*0.839*	*0.852*
YLT vs. KJV	*0.043*	*0.049*	*0.054*	*0.059*	*0.155*	*0.179*	*0.200*	*0.211*	*0.731*	*0.776*	*0.814*	*0.827*
YLT vs. WEB	*0.033*	*0.038*	*0.043*	*0.047*	*0.116*	*0.135*	*0.156*	*0.172*	*0.643*	*0.689*	*0.750*	*0.770*
YLT vs. Webster	*0.047*	*0.052*	*0.059*	*0.065*	*0.164*	*0.187*	*0.214*	*0.227*	*0.736*	*0.777*	*0.817*	*0.834*

Marked by underlines are the comparisons between *KJV* and *Webster* translations that have good results on all experiments. Marked by bold are these setting for a pairwise Bible comparison that provides best results. Marked by italic are the *YLT* and *Basic English* translations that are syntactically and semantically paraphrased

Table 4 Recall comparison between the four investigated pre-processing and three fingerprinting techniques using averaged recall values from Table 3

	Featuring		
Preprocessing	Trigram	Bigram	Word
Base	0.249	0.269	0.750
StringSim	0.261	0.403	0.773
Lemma	0.261	0.409	0.794
Lemma+Syn	0.269	0.420	0.812

to look at ways to figure out what other types of non-synonymic replacements can be observed. That is the second task we consider here.

6.2 *Extraction and Typing of Paradigmatic Relations*

We asked ourselves what the options of performing paraphrasing are. Paraphrasing is often described as the simple rewording or replacement of words through the use of, for example, synonyms or cohyponyms, as made available by *WordNet* [23, 35]. For historical texts, however, semantic databases are often not available. A further matter to consider is that, due to the longer time span, semantic relations such as *synonyms* are not always valid. This is because, due to the evolution of language, synonym relations have not necessarily remained stable over the centuries. To this we must add further layers of complexity. Thanks to the re-use links we discovered during the first task, we can investigate exactly what changes occur in paraphrases. This research can be viewed as an example of Shannon's Noisy Channel Theorem (cf. Fig. 1, [41]). As we have both a source and a target text, we can measure the differences in the noisy channel and look for any kind of systematic change. Here we assume that no change happens at random, but that each variation was made with purpose, so that this work can, in future, be extended to a *Noisy Channel Mining* [11]. For this investigation we used the setting S_{09}.

We begin by using all the positively linked verses from the first task. We ignore all the other links that exhibit too low a similarity (cf. Sect. 5). Next we process the positively linked verses using word bigrams. Finally, we tag as a candidate for a paradigmatic relation each word possessing other words in common on both the left-hand and right-hand side. Our introductory example of the text from Genesis 1:1 demonstrates how this works for the words *created*, *made*, and *preparing*.

Altogether we extract about 12,000 candidates for words in a paradigmatic relationship. In Table 5, we classify 8,193 of them. Of these, 4,055 data points can be simply identified by the already processed synonym and lemmatization data. The other 4,138 data points are not covered by those techniques. Word pairs that are tagged as e.g. *inflected variant* are not also explicitly tagged in Table 3 as string similar. The main idea of these classes is to provide a deeper understanding of what can be extracted.

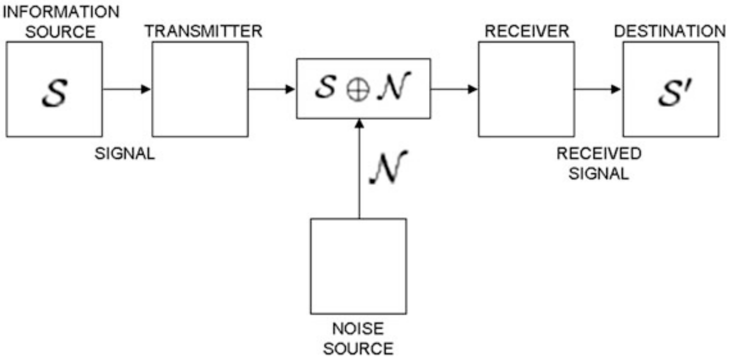


Fig. 1 Noisy Channel Theorem (cf. [41]) for *Historical Text Re-use Detection*: a source text or text passage S is often not re-used in a target text S' in the same exact way. Changes of the "noisy source" N influence the appearance of S'

Table 5 Identified systematic classes extracting paradigmatic relations

Relation type	Count	Sum
Synonyms	3,066	
Inflected variants	989	4,055
Similar written words	1,245	
Hyphen	451	
Prefix	545	
Suffix	84	
Compositions	512	
Archaic inflected variants	669	
Archaic synonyms	632	4,148
Sum	8,193	

Archaic inflected variants is a class that employs some basic rules to identify archaic word variants. If one of these rules can be applied and an English inflected variant is identified, the counter for this class is increased. A word pair is also tagged by the class *Archaic synonym* when an Archaic variant can be mapped to a synonym.

Composition is class that implies that at least one word of the candidate for a paradigmatic relation contains a hyphen. However, one of the 'subwords' must be similar such as *sea-beast* vs. *sea-monster*, *sea-gull* vs. *sea-mew* vs. *sea-hawk*, or *apple-tree* vs. *citron-tree*. This class is of special interest since in addition to synonyms like *sea-beast* and *sea-monster*, this class also contains cohyponyms such as *sea-mew* and *sea-hawk* or *apple-tree* and *citron-tree*.

The *Hyphen* class contains word pairs like *birth-day* vs. *birthday*, *back-bone* vs. *backbone*, *zareth-shahar* vs. *zarethshahar* that are the same variant when the hyphen is removed.

Prefix and *Suffix* are special classes that cover those candidates for a paradigmatic relation in which the longer word of the pair starts or ends with the shorter one.

Examples include *ambush* vs. *ambushment*, *shimite* vs. *shimites*, or *bearing* vs. *childbearing*.

All other string similar candidates are tagged by *Similar written word*. Example: *anathothite* vs. *anethothite* vs. *anetothite* vs. *annethothite* vs. *antothite*.

Almost 4,000 candidate words remain untagged. Besides some noise there exist further semantically similar words such as *punishment* vs. *torment*. Additionally, at this step of research we ignore any kind of relationship that included negations. An example is Genesis, chapter 34, verse 19: *not defer* (ASV, KJV, Webster) vs. *without loss of time* (Basic), *not delay* (Darby, YLT), and *not wait* (WEB).

7 Further Work

Paraphrase detection remains one of the most challenging tasks in the field of *text re-use* detection. Additional work is necessary to deal with spelling variants, especially for historical documents ranging over several centuries. For English texts, the previous work done as part of the VosPos [36] and MorphAdorner [14] projects should prove useful in expanding the ability of the re-use detection algorithms to uncover paradigmatic relationships in orthographically diverse texts.

We also intend to run the same types of experiments to see if we can detect modern examples of plagiarism. The problem is similar to detecting re-use in historical texts.

We believe it will be interesting to compare our results with those obtained by other approaches to uncovering *text re-use* such as the sequence alignment method suggested by Horton and Henderson [28].

We also expect that the methods proposed here will improve the ability of programs such as MorphAdorner [14] to adorn *Early Modern English* texts with parts of speech and lemmata. Including earlier *Middle English* translations of the Bible such as that of Wyclif and Hereford will also improve the ability of MorphAdorner and similar programs to process and morphologically adorn *Middle English* texts.

8 Conclusion

Working on the Bible has many benefits for the field of paraphrase detection since the many different translations all stem from one common origin, even though each individual translation reflects the variant interests of the translators and editors.

We determined that when a text is substantially rewritten as was done for the *Bible in Basic English*, the results are not better than a recall of 0.6. We do not expect significantly better results on an algorithmic level since the settings in Tables 3 and 4 circumscribe what is currently possible. For this reason, we worked on an additional task of extracting paradigmatic relations. In Table 5, we showed that dealing with lemmatization and synonyms is just one part of the possible changes that human beings perform when they paraphrase. It also became clear that language models can look for similarities. Paraphrasing, however, needs in many cases the cognitive ability of human beings such in the last example of Sect. 6: *not defer* vs. *without loss of time* vs. *not delay* vs. *not wait*.

For these reasons, we suggest our work offers one small but useful step in the direction of automated paraphrase detection in humanities research.

Acknowledgements This work has been made available by *eTRACES* (No. 01UA1101A) and the early career research group *eTRAP* (No. 01UG1409) of the German Ministry of Education and Research.

References

1. Allan J (2002) Topic detection and tracking: event-based information organization. Kluwer International Series on Information Retrieval. Kluwer Academic. ISBN: 9780792376644. http://books.google.de/books?id=50hnLI_Jz3cC
2. Basile C, Esposti MD, Rosso P, Barrón-Cedeño A (2010) Word length n-grams for text re-use detection. In: Proceedings of the 11th international conference on Computational Linguistics and Intelligent Text Processing, CICLing'10. Springer, Berlin/Heidelberg, pp 687–699. ISBN: 3-642-12115-2, 978-3-642-12115-9
3. Believers Resource (2011) XML encoded versions of several English language Bible translations. http://www.believersresource.com/categories/bible-raw-data.html. Accessed 11 Nov 2011
4. Bernstein Y, Croft WB, Moffat A, Zobel J, Metzler D (2005) Similarity measures for tracking information flow. In: Proceedings of the 14th ACM international conference on information and knowledge management, CIKM '05. ACM, New York, pp 517–524. doi:10.1145/1099554.1099695. ISBN: 1-59593-140-6. http://doi.acm.org/10.1145/1099554.1099695
5. Blei DM, Lafferty JD (2006) Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning, ICML '06, ACM, New York, pp 113–120. ISBN: 1-59593-383-2. doi:10.1145/1143844.1143859. http://doi.acm.org/10.1145/1143844.1143859
6. Bonzanini M, Roelleke T, Yahyaei S (2011) Cross-lingual text fragment alignment using divergence from randomness. In: Grossi R, Sebastiani F, Silvestri F (eds) String Processing and Information Retrieval, Lecture Notes in Computer Science, vol 7024, pp 14–25. ISBN: 978-3-642-24582-4
7. Bordag S (2007) Elements of Knowledge-free and Unsupervised Lexical Acquisition. Ph.D. thesis, Universität Leipzig
8. Brockett C, Dolan WB (2005) Automatically constructing a corpus of sentential paraphrases. In: Third International Workshop on Paraphrasing (IWP2005). Asia Federation of Natural Language Processing. http://research.microsoft.com/apps/pubs/default.aspx?id=101076
9. Broder AZ (1997) On the resemblance and containment of documents. In: In compression and complexity of sequences (SEQUENCES97. IEEE Computer Society, Los Alamitos, pp 21–29
10. Brück TVD, Eichhorn C, Hartrumpf S (2010) Semantic duplicate identification with parsing and machine learning. In: TSD, pp 84–92
11. Büchler M (2013) Informationstechnische aspekte des historischen text re-use. Ph.D. thesis, Leipzig University, Germany
12. Büchler M, Boehlke V, Heyer G (2011) Aspects of an infrastructure for eHumanities. In: Proceedings of Supporting Digital Humanities 2011

13. Büchler M, Scheuermann G, Jänicke S (2014) Visualizations for text re-use. In: Proceedings of the 5th International Conference on Information Visualization Theory and Applications, IVAPP 2014
14. Burns P (2012) MorphAdorner. http://morphadorner.northwestern.edu/. Accessed 1 Nov 2012
15. Burns PR, Crane G, Mueller M, Heyer G, Büchler M (2011) One step closer to paraphrase detection on historical texts: about the quality of text re-use techniques and the ability to learn paradigmatic relations. In: Proceedings of the 2011 Chicago Colloquium on Digital Humanities and Computer Science, Chicago, 2012
16. Charikar M (2002) Similarity estimation techniques from rounding algorithms. In: John HR (ed) STOC. ACM, New York, pp 380–388. ISBN: 1-58113-495-9
17. Charikar M, Frieze AM, Mitzenmacher M, Broder AZ (1998) Min-wise independent permutations. J Comput Syst Sci 60:327–336. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.121.8215
18. Cordell R, Dillon EM, Smith DA (2013) Infectious texts: Modeling text reuse in nineteenth-century newspapers. In: IEEE International Conference on Big Data, pp 86–94. doi:10.1109/BigData.2013.6691675
19. Crane G (2006) What do you do with a million books? D-Lib Magazine 12:3. doi:10.1045/march2006-crane. ISSN: 1082-9873. http://www.dlib.org/dlib/march06/crane/03crane.html
20. Croft WB, Seo J (2008) Local text reuse detection. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, pp 571–578. doi:http://doi.acm.org/10.1145/1390334.1390432. ISBN: 978-1-60558-164-4
21. Croft WB, Bendersky M (2009) Finding text reuse on the web. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09. ACM, New York, pp 262–271. ISBN: 978-1-60558-390-7. doi:10.1145/1498759.1498835. http://doi.acm.org/10.1145/1498759.1498835
22. Elhadad N, Barzilay R (2003) Sentence alignment for monolingual comparable corpora. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, EMNLP '03. Association for Computational Linguistics, Stroudsburg, pp 25–32. doi:10.3115/1119355.1119359. http://dx.doi.org/10.3115/1119355.1119359
23. Fellbaum C (ed) (1998) WordNet: an electronic lexical database. MIT, Cambridge. ISBN: 978-0-262-06197-1
24. Harris Z (1954) Distributional structure. Word 10(23):146–162
25. Hagen M, Beyer A, Busse M, Tippmann M, Rosso P, Stein B, Potthast M (2014) Overview of the 6th international competition on plagiarism detection. In: Cappellato L, Ferro L, Halvey M, Kraaij W (eds) Working Notes Papers of the CLEF 2014 Evaluation Labs, CEUR Workshop Proceedings. CLEF and CEUR-WS.org. http://www.clef-initiative.eu/publication/working-notes
26. Heyer G (2009) Analyse von Bedeutungsveränderungen in diachronen Textkorpora. Technical report, Natural Language Processing Group, University of Leipzig, Germany, Februar 2009. Vortrag im Forschungsseminar, Leipzig, Germany
27. Hofmann T (1999) Probabilistic latent semantic analysis. In: Kathryn BL, Henri P (eds) UAI. Morgan Kaufmann, Stockholm, pp 289–296
28. Horton R, Henderson L (2010) Sequence alignment and similarity in biology and the humanities. J Chicago Colloq Digit Humanit Comput Sci
29. Hose R (2004) CS490 final report: investigation of sentence level text reuse algorithms. At Bits On Our Minds workshop at Cornell University
30. Hunt E, Stiller B, Bocek T (2007) Fast Similarity Search in Large Dictionaries. Technical Report ifi-2007.02, Department of Informatics, University of Zurich. http://fastss.csg.uzh.ch/
31. Klein E, Loper E, Bird S (2009) Natural Language Processing with Python. Oreilly Series. O'Reilly Media. ISBN: 9780596516499. http://books.google.de/books?id=KGIbfiiP1i4C
32. Lee J (2007) A computational model of text reuse in ancient literary texts. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Prague, pp 472–479. http://www.aclweb.org/anthology/P07-1060

33. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8
34. Mayer T, Cysouw M (2014) Creating a massively parallel bible corpus. In: Calzolari N (Conference Chair), Choukri K, Declerck T, Loftsson H, Maegaard B, Mariani J, Moreno A, Odijk J, Piperidis S (eds) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland. European Language Resources Association (ELRA). ISBN: 978-2-9517408-8-4
35. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41. doi:10.1145/219717.219748. ISSN: 0001-0782. http://doi.acm.org/10.1145/219717.219748
36. Mueller M (2006) VosPos: a project for virtual orthographic standardization and part of speech tagging of early modern english texts. http://panini.northwestern.edu/mmueller/nupos.pdf. Accessed 13 Nov 2014
37. Niekler A, Wiedemann G, Heyer G (2014) Brauchen die Digital Humanities eine eigene Methodologie? Überlegungen zur systematischen Nutzung von Text Mining Verfahren in einem politikwissenschaftlichen Projekt. Proceedings, 03
38. Nord C, Girona J, Talavera L (2000) Dependency-based feature selection for clustering symbolic data. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1720
39. Potthast M, Beyer A, Busse M, Rangel F, Rosso P, Stamatatos E, Stein B, Gollub, T (2013) Recent trends in digital text forensics and its evaluation: plagiarism detection, author identification, and author profiling. In: 4th Int. Conf. of CLEF on information access evaluation meets multilinguality, multimodality, and visualization (CLEF 2013). Springer, New York
40. Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley/Longman, Boston. ISBN: 0-201-12227-8
41. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423
42. Stein B (2007) Principles of hash-based text retrieval. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '07. ACM, New York, pp 527–534. doi:10.1145/1277741.1277832. ISBN: 978-1-59593-597-7. http://doi.acm.org/10.1145/1277741.1277832
43. Wilkerson DS, Aiken A, Schleimer S (2003) Winnowing: local algorithms for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, SIGMOD '03. ACM, New York, pp 76–85. doi:10.1145/872757.872770. ISBN 1-58113-634-X. http://doi.acm.org/10.1145/872757.872770
44. Zesch T, Gurevych I, Bär D (2012) Text reuse detection using a composition of text similarity measures. In: Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012), pp 167–184
45. Zipf G (1949) Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge