Theory and Applications of Natural Language Processing
Edited volumes

Inguna Skadiņa · Robert Gaizauskas
Bogdan Babych · Nikola Ljubešić
Dan Tufiş · Andrejs Vasiļjevs *Editors*

Using Comparable Corpora for Under-Resourced Areas of Machine Translation

Springer

Theory and Applications of Natural Language Processing

Series editors
Julia Hirschberg
Eduard Hovy
Mark Johnson

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

Inguna Skadiņa • Robert Gaizauskas •
Bogdan Babych • Nikola Ljubešić • Dan Tufiş •
Andrejs Vasiļjevs
Editors

Using Comparable Corpora for Under-Resourced Areas of Machine Translation

Springer

Editors
Inguna Skadiņa
Tilde
Riga, Latvia

Robert Gaizauskas
Department of Computer Science
University of Sheffield
Sheffield, United Kingdom

Bogdan Babych
School of Modern Languages & Cultures
University of Leeds
Leeds, United Kingdom

Nikola Ljubešić
Faculty of Humanities & Social Sciences
University of Zagreb
Zagreb, Croatia

Dan Tufiş
Institute for Artificial Intelligence
Romanian Academy
Bucharest, Romania

Andrejs Vasiļjevs
Tilde
Riga, Latvia

ISSN 2192-032X ISSN 2192-0338 (electronic)
Theory and Applications of Natural Language Processing
ISBN 978-3-319-99003-3 ISBN 978-3-319-99004-0 (eBook)
https://doi.org/10.1007/978-3-319-99004-0

Library of Congress Control Number: 2018961403

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

Contents

Chapter 1
Introduction

Inguna Skadiņa, Robert Gaizauskas, Andrejs Vasiļjevs, and Monica Lestari Paramita

Despite the long history of research into automated translation technologies, the original goal of machine translation (MT)—to replace human translators—has not yet been met. Machine translation systems are still unable to produce output of the same quality as a human translator. However, machine translation has proven to be a very useful tool in such scenarios as gisting information in unknown languages and providing raw translation for post-editing. The need for fast and cheap translation has resulted in several freely available Web services (e.g. *Google Translate*, *Bing*) demonstrating acceptable translation quality for widely used languages.

In recent decades, data-driven approaches have significantly advanced the development of machine translation. Systems that are able to learn from huge parallel corpora provide an effective solution that minimizes time- and resource-consuming manual work in defining linguistic knowledge, which earlier approaches to machine translation required.

Cost-effectiveness is one of the key reasons why the data-driven paradigm, which includes both statistical and neural MT approaches, has come to be the dominant current framework for MT, in both theory and practice. Data-driven machine translation systems have demonstrated good performance for languages and domains for which large parallel corpora are available. However, the applicability of these methods directly depends on the availability of very large quantities of parallel corpus data. As a result, for many years data-driven machine translation research has mainly been focused on widely used languages, such as Arabic, Chinese, English, German and French.

I. Skadiņa (✉) · A. Vasiļjevs
Tilde, Riga, Latvia
e-mail: Inguna.Skadina@tilde.lv

R. Gaizauskas · M. L. Paramita
University of Sheffield, Sheffield, UK

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_1

1.1 Parallel Data

Parallel data are valuable linguistic resources for building MT systems. Unfortunately, the availability of parallel data, especially for 'smaller' or under-resourced languages, is very limited. There are only a few publicly available parallel corpora for the lesser-spoken languages of Europe. Amongst them the most popular are corpora from the legal domain—JRC-Acquis (Steinberger et al. 2006)—a huge collection of European Union legislative texts translated into more than 20 official European languages; the European Parliament Proceedings Parallel Corpus (Europarl corpus) which was extracted from the proceedings of the European Parliament and now includes 21 European languages (Koehn 2005); and DGT-TM (Steinberger et al. 2012)—multilingual Translation Memory for the Acquis Communautaire in 23 EU languages. Different corpora are presented in the OPUS collection (Tiedemann 2016)—books, a European Central Bank corpus, translated United Nations documents, TED talks, etc. SETimes (Tyers and Alpren 2010) is a parallel corpus from a multilingual news website including English and eight South-East European Languages (Albanian, Bulgarian, Croatian, Greek, Macedonian, Romanian, Serbian and Turkish). These corpora have a sufficient amount of data for some domains and allow MT systems to be trained that achieve high scores when evaluated on in-domain texts. However, they have poor results when used for other domains.

For 'small' or under-resourced languages, MT solutions are not so well developed due to the lack of linguistic resources and technological approaches that enable MT solutions for new language pairs to be developed cost-effectively (Rehm and Uszkoreit 2012). Since adequate amounts of parallel data for under-resourced languages are unlikely to become available any time soon, it is important to find technological solutions that compensate for this shortage of linguistic resources for under-resourced languages. The same problems arise even for better-resourced languages when translation is required in narrow, specialised domains (e.g. information technology, medicine, automotive) where inadequate amounts of domain-specific parallel text exist for data-driven approaches to deliver high-quality translation.

1.2 Comparable Corpora and Comparability

For many language pairs and specialized domains there is a dearth of *parallel text* data. However, in many cases large quantities of *comparable text* data are available, that is text pairs in different languages where although one text is not a direct translation of the other, both texts share a significant amount of common content. Examples include news feeds in different languages reporting the same event and multilingual Web pages, for example Wikipedia. A huge amount of such text is freely available on the Web and being added to daily. From it, multilingual

comparable corpora can be built. Could such comparable resources be useful for machine translation?

A comparable corpus is usually defined as a collection of similar documents that are gathered according to a set of criteria, for example the same proportions of texts of the same genre in the same domain from the same period (McEnery and Xiao 2007) in more than one language or variety of languages (EAGLES 1996) that contain overlapping information (Munteanu and Marcu 2005; Hewavitharana and Vogel 2008). While methods for the use of parallel corpora in machine translation are well studied (e.g. Koehn 2010; Sennrich et al. 2016a, b), similar techniques for utilising comparable corpora have not been thoroughly worked out. However, several studies have suggested that language pairs and domains with little parallel data can benefit from the exploitation of comparable corpora (Munteanu and Marcu 2005; Lu et al. 2010; Smith et al. 2010; Abdul-Rauf and Schwenk 2009, 2011).

One useful multilingual comparable corpus is Wikipedia, as it contains documents that describe a wide range of topics in a large number of languages. Moreover, multilingual Wikipedia documents that describe the same topic are linked together, allowing easy extraction for topically related documents, although their degrees of comparability may vary widely. Some of these documents may be translation equivalents, whilst others may be less similar. However, since these documents describe the same topic, they are likely to contain lexical overlap, which is useful for building resources for under-resourced languages. Methods to extract these texts to improve MT performance have been investigated in previous studies (Xu et al. 2012).

An important issue is that of assessing degree of comparability. There have been some attempts to determine different kinds of document parallelism in comparable corpora, such as complete parallelism, noisy parallelism and complete non-parallelism, and define criteria of parallelism of similar documents in comparable corpora, such as similar number of sentences, sharing sufficiently many links, similarity of document structure, and similarity of lexical content. Kilgarriff (2001) and Rayson and Garside (2000) have studied objective measures for detecting how similar (or different) two corpora are in terms of their lexical content. Further studies (Sharoff 2007) have investigated automatic ways for assessing the composition of Web corpora in terms of domains and genres. Li and Gaussier (2010) propose a comparability metric that measures the proportion of overlapping words translated from the source language corpus into the target language by looking in a bilingual dictionary.

1.3 Acquisition of Parallel Data from Comparable Corpora

Once comparable corpora are built, one needs to extract parallel data from within them. The extraction of parallel data—paragraphs, sentences, phrasal units, named entities (NEs) and terms—from comparable corpora is much more challenging than from parallel corpora. Depending on the nature of the comparable corpus, only some

or perhaps none of the sentences in any pair of texts from the two languages will be translations of each other.

Many methods that are designed for parallel texts perform poorly when applied to comparable corpora. Zhao and Vogel (2002) and Utiyama and Isahara (2003) extended algorithms designed to perform sentence alignment of parallel texts to apply them to comparable corpora. They started by attempting to identify similar article pairs from the two corpora. Then they treated each of those pairs as parallel texts and aligned their sentences by defining a sentence pair similarity score and used dynamic programming to find the least-cost alignment over the whole document pair. The performance of these approaches depends heavily on the degrees of similarity of the document pairs. Moreover, comparable article pairs, even those similar in content, may exhibit great differences at the sentence level (reordering, additions, etc.).

Fung and Cheung (2004) use a word translation similarity measure to discover similar documents. For identifying similar sentence pairs within comparable corpora, Munteanu and Marcu (2005) proposed a word-overlap filter (half the words of the source language sentence must have a translation in the target language sentence) together with a constraint on the ratio of lengths of the two sentences. If two sentences meet these criteria, then a maximum entropy classifier, trained over a small parallel corpus, decides whether they are parallel sentences. In these cases, translation resources are required to perform translations prior to identifying overlapping contents.

For Wikipedia documents, Adafre and de Rijke (2006) developed a method to retrieve parallel sentences by measuring the overlap of links between sentences. Smith et al. (2010) further developed this idea by using additional features, such as sentence length and longest aligned/unaligned words to develop a binary classifier trained on parallel corpora.

Another approach for detecting meaning-equivalent sentence pairs within comparable corpora is to use cross-lingual Q&A techniques. The main idea is to exploit dependency linking (Ion and Tufiş 2007) and the concepts of superlinks and chained links (Irimia 2009) for determining the most relevant search criteria. The keywords extracted from the dependency linking of a source paragraph/sentence are translated into a target language and search engines look for the most relevant candidate paragraph/sentences. The possible pairs of translation equivalent textual units are then scored by a reified sentence aligner (Ceauşu et al. 2006) and are accepted or rejected based on previously determined thresholds.

Common named entities (NEs) or technical terms in phrases from texts in different languages are powerful indicators that the phrases may be translation equivalents, and their absence almost certainly suggests that the phrases are not equivalents. Named entities typically fall into two types: those which are more or less phonetically equivalent in two languages and those some or all of whose component words are translated individually. In cases where the NEs or terms are not phonologically related, entity type equivalence together with dictionary matching on component words may be used to align them. In cases where they are

phonologically related, a process of matching based on transliteration similarity may be used.

Research on bilingual terminology extraction usually relies on the assumption that words with the same meaning in different languages tend to appear in the same context (Rapp 1995). The most common approach is to use context vectors and evaluate candidate translations. On single words, this approach has demonstrated good results (e.g. Chiao and Zweigenbaum 2002). Daille and Morin (2008) adapted this direct context vector approach for single and multi-word terms and added compositional translation methods for French–Japanese languages. This method increased the results of Morin et al. (2007) by 10%; however, they are still rather low for multi-word terms.

The need for extracting parallel data from comparable corpora has been also recognized in a recent shared task on parallel sentence extraction at the Workshop on Building and Using Comparable Corpora (Zweigenbaum et al. 2018). In this task, the best results were achieved by Azpeitia et al. (2018), with scores over 80% in terms of f1-measure and 90% in precision. Their approach is based on variants of the STACC method (Etchegoyhen and Azpeitia 2016), which computes similarity on expanded lexical sets via Jaccard similarity. They apply the weighted variant of the method and in addition introduce a variant that further penalizes mismatches in terms of named entities, improving over the already strong weighted variant baseline in most cases.

1.4 Comparable Corpora in Machine Translation

Research on applicability of comparable corpora to the MT task has shown that adding extracted aligned parallel lexical data from comparable corpora to the training data of an statistical machine translation (SMT) system improves the system's performance in respect of un-translated word coverage (Hewavitharana and Vogel 2008). It has been also demonstrated that language pairs with little parallel data can benefit the most from exploitation of comparable corpora (Lu et al. 2010). Munteanu and Marcu (2005) achieved significant performance improvements from large comparable corpora of news feeds for English, Arabic and Chinese. Irvine and Callison-Burch (2013) also used comparable corpora to improve the accuracy and coverage of phrase-based MT built on small amounts of parallel data. They showed that adding translations of low-frequency words from comparable corpora improves performance beyond what is achieved by inducing translations for out-of-vocabulary words alone.

However, before the ACCURAT project (Skadiņa et al. 2012)—see below—experiments with comparable corpora in the machine translation task were performed mainly with widely used language pairs, such as French-English (Abdul-Rauf and Schwenk 2009, 2011), Arabic–English (Abdul-Rauf and Schwenk 2011), or English–German (Ştefănescu et al. 2012).

Since 2014, the research field of machine translation has experienced a paradigm shift from traditional SMT technologies to neural machine translation technologies (Bahdanau et al. 2014; Devlin et al. 2014; Jean et al. 2015; Luong et al. 2015; etc.). In 2016, neural machine translation (NMT) systems were shown to achieve significantly better results than statistical systems for multiple language pairs including English–German, English–Czech, and English–Romanian (Sennrich et al. 2016a; Bojar et al. 2016), thereby paving the way for neural machine translation as the potential future technology for state-of-the-art machine translation system development.

With the paradigm shift from statistical MT to neural MT, there is still a need for large parallel corpora: Koehn and Knowles (2017) analysed the impact of corpus size on translation quality and showed that in low-resource settings, SMT demonstrates better quality than NMT, while in high resource settings, NMT outperforms SMT. One common way to overcome the bottleneck of insufficient parallel data for training NMT systems is to use back-translation (Sennrich et al. 2016b), that is to create an artificial strongly comparable corpus by machine translation of an in-domain monolingual corpus. This back-translated corpus is then added to the existing parallel corpus and either domain adaptation of a pre-trained NMT system is performed, or a new in-domain NMT system is trained. Back-translation has been shown to be a standard method for domain adaptation without parallel data in NMT. For instance, out of 22 participants who developed NMT systems in the 2017 shared task on news translation organised as part of the Conference on Machine Translation, only 2 participants did not use back-translation (Bojar et al. 2017).

1.5 Summary of the Book

This book addresses the full set of questions that arise when attempting to exploit comparable corpora to overcome the bottleneck of insufficient parallel corpora that affects any data-driven machine translation approach, particularly in relation to under-resourced languages and narrow domains. It describes a full set of methods and tools for identifying and assessing comparability, for gathering comparable corpora from the Web, for extracting translation equivalents from within comparable texts and discusses the evaluation of this pipeline of methods and tools by incorporating their outputs into a machine translation system and assessing its performance in real application settings.

Most of the methods discussed in a book are language independent. Special attention was paid to applicability for a number of under-resourced languages—Croatian, Estonian, Greek, Latvian, Lithuanian, Romanian, and Slovenian—and narrow domains (e.g. information technology and the automotive industry).

What is a good comparable corpus for the machine translation task, how to obtain such a corpus, and how to measure its usability in machine translation are central questions of this book. Chapter 2 investigates cross-lingual comparability and the methods by which comparability and likely utility for MT can be measured.

Chapter 3 explores ways in which comparable corpora can be acquired from the Web.

After proposing methods to assess usability of comparable corpus for the machine translation task and proposing methods for collecting comparable corpora from the Web, the book discusses different methods to extract data from comparable corpora that can be used for machine translation. Chapter 4 describes methods and tools to identify terms, named entities, and other lexical units in comparable corpora, and to cross-lingually map the identified units in order to create automatically extracted bilingual dictionaries that can be further utilised in machine translation. Chapter 5 presents different approaches to mining parallel sentences and phrases from comparable corpora.

Chapter 6 analyses methods that utilise data extracted from comparable corpora for the machine translation task and investigates their impact on MT output for under-resourced languages and in narrow domains. Three use cases are presented and analysed: (1) German–English MT adaptation to the automotive domain, (2) Croatian–English, Slovenian–English and German–English machine translation in Web authoring and (3) the application of English–Latvian MT systems, enriched with data from comparable corpora, in computer-aided translation.

Chapter 7 presents research on using comparable corpora beyond the narrow focus of improving MT for under-resourced languages and narrow domains. This chapter presents methods to extract parallel sentences from quasi-comparable and comparable corpora. Chapter 7 also addresses the task of creating resources for bilingual dictionaries by proposing methodology of creating bilingual dictionaries (1) using a seed lexicon and (2) without an initial lexicon.

1.6 The ACCURAT Project

The work reported in this book had its genesis in the ACCURAT project (Analysis and evaluation of comparable corpora for under resourced areas of machine translation). ACCURAT (2010–2012) was a research and technology development project funded as part of the EU Framework Programme 7 (FP7). The aim of the project was to find, analyse and evaluate novel methods that exploit comparable corpora in order to compensate for the shortage of linguistic resources, and ultimately to significantly improve MT quality for under-resourced languages and narrow domains (Skadiņa et al. 2012).

Among scientific objectives of the ACCURAT projects were creation of comparability metrics to measure the similarity of source and target language documents in comparable corpora; researching methods for alignment and extraction of lexical, terminological and other linguistic data from comparable corpora; researching on methods for automatic acquisition of a comparable corpus from the Web and researching usability of comparable corpora for machine translation development.

The research activities supported development of practical tools for comparable corpora acquisition, document alignment, data extraction from comparable corpora.

The project worked on nine EU languages (English, German, Latvian, Greek, Croatian, Estonian, Lithuanian, Slovenian and Romanian) and produced tools and resources for each of them.

Open source tools (Pinnis et al. 2012), project deliverables, and publications are all available from the ACCURAT website: http://www.accurat-project.eu.

ACCURAT was initiated at a time when statistical machine translation was the state of the art for widely used languages, but when it had become apparent that for many under-resourced languages, SMT translations were poor due to the lack of large parallel corpora. The project was not aimed at changing the MT paradigm but at exploring how to overcome data sparseness that affects any data-driven approach, specifically by investigating the promising but under-researched idea that comparable corpora could provide the key to providing the much-needed data for under-resourced languages and narrow domains.

While the orientation of this book is around a specific project, it is not merely a 'project report'. Authors of this book were constantly seeking novel and alternative approaches and were not bound to a pre-identified project-specific approach. This book presents an extensive set of scientific and technological issues that anyone wanting to rigorously investigate the exploitation of comparable corpora would need to address.

The series editors discussed the book's project-orientation vs. its 'neutrality,' and decided to accept it, thanks to the care and thoroughness reflected in the work in general and its quite extensive readings' lists and citations. We note further that despite the work having been completed several years ago, the absence of current initiatives of comparable scope means the material presented here remains highly relevant to any effort to improve data-driven MT capability for under-resourced languages and narrow domains by exploiting comparable corpora.

References

Abdul-Rauf, S., & Schwenk, H. (2009). On the use of comparable corpora to improve SMT performance. *EACL 2009: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 16–23), Athens, Greece.

Abdul-Rauf, S., & Schwenk, H. (2011). Parallel sentence generation from comparable corpora for improved SMT. *Machine Translation, 25*(4), 341–375.

Adafre, S. F., & de Rijke, M. (2006). Finding similar sentences across multiple languages in Wikipedia. *Proceedings of the EACL Workshop on New Text*, Trento, Italy.

Azpeitia, A., Etchegoyhen, T., & Martinez Garcia, E. (2018). Extracting parallel sentences from comparable corpora with STACC variants. *Proceedings of the 11th Workshop on Building and Using Comparable Corpora* (pp. 48–52).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *CoRR*, *abs/1409.0*. article. Retrieved from http://arxiv.org/abs/1409.0473

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., et al. (2016). Findings of the 2016 conference on machine translation. *Proceedings of the First Conference on Machine Translation (WMT 2016), Vol. 2: Shared Task Papers* (pp. 131–198).

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., et al. (2017). Findings of the 2017 conference on machine translation (WMT17). *Proceedings of the Second Conference on Machine Translation, Vol. 2: Shared Task Papers* (pp. 169–214). Association for Computational Linguistics, Copenhagen, Denmark. Retrieved from http://www.aclweb.org/anthology/W17-4717

Ceauşu, A., Ştefănescu, D., & Tufiş, D. (2006). Acquis communautaire sentence alignment using support vector machines. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)* (pp. 2134–2137).

Chiao, Y., & Zweigenbaum, P. (2002). Looking for candidate translational equivalents in specialized, comparable. *COLING '02 Proceedings of the 19th International Conference on Computational Linguistics* (Vol. 2, pp. 1–5).

Daille, B., & Morin, E. (2008). An effective compositional model for lexical alignment. *Proceedings, 3rd International Joint Conference on Natural Language Processing (IJCLNP)* (pp. 95–102).

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., & Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. *ACL* (1) (pp. 1370–1380). *In Proceedings*.

EAGLES. (1996). *Preliminary recommendations on corpus typology*. Electronic Resource: http://www.ilc.cnr.it/EAGLES96/corpustyp/corpustyp.html

Etchegoyhen, T., & Azpeitia, A. (2016). Set-theoretic alignment for comparable corpora. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Vol.1: Long Papers, pp. 2009–2018).

Fung, P., & Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)* (pp. 57–63), Barcelona, Spain.

Hewavitharana, S., & Vogel, S. (2008). Enhancing a statistical machine translation system by using an automatically extracted parallel corpus from comparable sources. *Proceedings of the Workshop on Comparable Corpora, LREC'08* (pp. 7–10).

Ion, R., & Tufiş, D. (2007). RACAI: Meaning affinity models. *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)* (pp. 282–287), Association for Computational Linguistics, Prague, Czech Republic, June 2007.

Irimia, E. (2009). *Metode de traducere automată prin analogie. Aplicaţii pentru limbile română şi engleză. (Methods for Analogy-based Machine Translation. Applications for Romanian and English)*. PhD thesis, March 2009.

Irvine, A., & Callison-Burch, Ch. (2013). Combining bilingual and comparable corpora for low resource machine translation. *Proceedings of the Eighth Workshop on Statistical Machine Translation* (pp. 262—270).

Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015). Montreal neural machine translation systems for WMT15. *Proceedings of the Tenth Workshop on Statistical Machine Translation* (pp. 134–140).

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics, 6*(1), 1–37.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. *Proceedings of Machine Translation Summit X*.

Koehn, P. (2010). *Statistical machine translation*. Cambridge University Press.

Koehn, P., & Knowles, R. (2017). Six challenges for neural machine translation. *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017* (pp. 28–39), Vancouver, Canada, August 4, 2017.

Li, B., & Gaussier, E. (2010). Improving corpus comparability for bilingual lexicon extraction from comparable corpora. *Proceedings of COLING 2010*, Beijing, China.

Lu, B., Jiang, T., Chow, K., & Tsou, B. K. (2010). Building a large English-Chinese parallel corpus from comparable patents and its experimental application to SMT. *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora: From Parallel to Non-parallel Corpora*, Valletta, Malta (pp. 42–48).

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1412–1421).

McEnery, A., & Xiao, Z. (2007). Parallel and comparable corpora? *Incorporating Corpora: Translation and the Linguist. Translating Europe*. Multilingual Matters, Clevedon.

Morin, E., Daille, B., Takeuchi, K., & Kageura, K. (2007). Bilingual terminology mining – Using brain, not brawn comparable corpora. *Proceedings, 45th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 664–671).

Munteanu, D., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Pinnis, M., Ion, R., Ştefănescu, D., Su, F., Skadiņa, I., Vasiļjevs, A. et al. (2012). ACCURAT toolkit for multi-level alignment and information extraction from comparable corpora. *Proceedings of the ACL 2012 System Demonstrations* (pp. 91–96). Association for Computational Linguistics, Jeju, South Korea.

Rapp, R. (1995). Identifying word translations in non-parallel texts. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* (pp. 320–322).

Rayson, P., & Garside, R. (2000) Comparing corpora using frequency profiling. *Proceedings of the Comparing Corpora Workshop at ACL'00* (pp. 1–6).

Rehm, G., & Uszkoreit, H. (Eds.). (2012). White paper series. Springer.

Sennrich, R., Haddow, B., & Birch, A. (2016a). Edinburgh neural machine translation systems for WMT 16. *Proceedings of the First Conference on Machine Translation, Vol. 2: Shared Task Papers* (pp. 368–373), Berlin, Germany.

Sennrich, R., Hadow, B., & Birch, A. (2016b). Improving neural machine translation models with monolingual data. *Proceedings of Annual Meeting of ACL* (pp. 86–96).

Sharoff, S. (2007). Classifying Web corpora into domain and genre using automatic feature identification. *Proceedings of 3rd Web as Corpus Workshop*, Louvain-la-Neuve, Belgium

Skadiņa, I., Aker, A., Mastropavlos, N., Su, F., Tufiş, D., Verlic, M. et al. (2012). Collecting and using comparable corpora for statistical machine translation. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (pp. 438–445).

Smith, J.R., Quirk, C., & Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. *NAACL-HLT 2010* (pp. 403–411).

Ştefănescu, D., Ion, R., & Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)* (pp. 137–144), Trento, Italy.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D. et al. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation: LREC'06*.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., & Schlüter, P. (2012). DGT-TM: A freely available translation memory in 22 languages. *Proceedings of LREC'2012* (pp. 454–459), Istanbul, Turkey.

Tiedemann, J. (2016). OPUS – Parallel corpora for everyone. *Baltic Journal of Modern Computing (BJMC), 4*(2). Special Issue: *Proceedings of the 19th Annual Conference of the European Association of Machine Translation (EAMT)*, 2016.

Tyers, F. M., & Alpren, M. S. (2010). South-East European Times: A parallel corpus of Balkan languages. *Proceedings of Workshop "Exploitation of Multilingual Resources and Tools for Central and (South) Eastern European Languages"*.

Utiyama, M., & Isahara, H. (2003). Reliable measures for aligning Japanese-English news articles and sentences. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 7–12).

Xu, J., Kennington, C., Przywara, C., & Wanzare, L. (2012). Comparable corpora in Wikipedia text for machine translation. *Proceedings of the 6th NIC Symposium 2012: 25 Years HLRZ/NIC (Book Section)*. ISBN: 9783893367580, Jülich, Germany, February 2012.

Zhao, B., & Vogel, S. (2002). Adaptive parallel sentences mining from web bilingual news collection. *Proceedings of the 2002 I.E. International Conference on Data Mining (ICDM'02)* (p. 74).

Zweigenbaum, P., Sharoff, S., & Rapp, R. (2018). Overview of the third BUCC shared task: Spotting parallel sentences in comparable corpora. *Proceedings of 11th Workshop on Building and Using Comparable Corpora* (pp. 39–42).

Chapter 2
Cross-Language Comparability and Its Applications for MT

Bogdan Babych, Fangzhong Su, Anthony Hartley, Ahmet Aker, Monica Lestari Paramita, Paul Clough, and Robert Gaizauskas

Abstract The concept of comparability, or linguistic relatedness, or closeness between textual units or corpora has many possible applications in computational linguistics. Consequently, the task of measuring comparability has increasingly become a core technological challenge in the field, and needs to be developed and evaluated systematically. Many practical applications require corpora with controlled levels of comparability, which are established by comparability metrics. From this perspective, it is important to understand the linguistic and technological mechanisms and implications of comparability and develop a systematic methodology for developing, evaluating and using comparability metrics. This chapter presents our approach to developing and using such metrics for machine translation (MT), especially for under-resourced languages. We address three core areas: (1) systematic *meta-evaluation* (or calibration) of the metrics on the basis of parallel corpora; (2) the development of *feature-selection techniques* for the metrics on the basis of aligned comparable texts, such as Wikipedia articles and (3) applying the developed metrics for the tasks of MT for under-resourced languages and measuring their effectiveness for corpora with unknown degrees of comparability. This has led to redefining the vague linguistic concept of comparability in terms of task-specific performance of the tools, which extract phrase-level translation equivalents from comparable texts.

Chapter editors: Bogdan Babych and Robert Gaizauskas

B. Babych (✉) · F. Su · A. Hartley
University of Leeds, Leeds, UK
e-mail: b.babych@leeds.ac.uk

A. Aker · M. L. Paramita · P. Clough · R. Gaizauskas
University of Sheffield, Sheffield, UK

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_2

2.1 Introduction: Definition and Use of the Concept of Comparability

In several areas of computational linguistics, there is a growing interest in measuring the degree of 'similarity', or 'comparability' between different linguistic units, such as corpora, sub-corpora, text, paragraphs, sentences or phrases. Interpretation of the concept of comparability varies according to the range of intended applications, but different areas share the need for using similar units in the same or different languages, and measuring the degree of similarity between the compared units, possibly of different granularity (paragraphs, texts or corpora). This need leads to the idea that it is useful to have an automated metric that ranks corpora, sub-corpora or documents according to the degree of their 'closeness' to each other. Typically, closeness is either measured by pre-defined formal parameters (such as lexical overlap) or intuitively described in terms of less formal linguistic categories (such as *genre* or *subject domain*). This section outlines our rationale for developing formal metrics based on combinations of measurable parameters that correlate with human intuitions about the linguistic categories in question.

The concept of comparability is relevant both in monolingual contexts (as similarity between corpora/texts written in the same language) and in cross-lingual contexts (as similarity of corpora/texts in different languages).

In the monolingual context, the concept of corpus comparability is used in computational lexicography for building translation dictionaries (e.g. Teubert 1996), and in corpus linguistics for identifying qualitative differences between language varieties (e.g. British vs. American English), domains, modalities (spoken vs. written language), in order, for example, to determine which words are particularly characteristic of a corpus or text (Kilgarriff 2001: 233; Rayson and Garside 2000). Another monolingual application of comparability is automatic identification of domains and genres for texts on the web (e.g. Kessler et al. 1998; Sharoff 2007; Vidulin et al. 2007; Kanaris and Stamatatos 2009; Wu et al. 2010), with the goal of developing domain-sensitive and genre-enabled information retrieval (IR) methods, which can restrict search according to automatically identified fine-grained text types (such as blogs, forum discussions, editorials, analytical articles, news, user manuals, etc.).

Cross-lingual comparable corpora are frequently used for identifying potential translation equivalents for words, phrases or terminological expressions (Rapp 1995, 1999; Fung 1998; Fung and Yee 1998; Daille and Morin 2005; Morin et al. 2007; Chiao and Zweigenbaum 2002), or supporting human translators in dealing with non-trivial translation problems (Sharoff et al. 2006; Babych et al. 2007). Multilingual comparable corpora are now becoming increasingly useful in training translation models for statistical machine translation (SMT) (Wu and Fung 2005; Munteanu et al. 2004; Munteanu and Marcu 2006), especially for under-resourced languages, where traditional parallel resources are not available, or are very small or in some other way unrepresentative (Skadiņa et al. 2010; Eisele and Xu 2010). There are several dimensions of comparability, which can be summarised as follows:

1. The granularity of comparability dimension: a measure of correspondence between units at different structural levels:
   (a) *Corpus-level* comparability—between corpus A and corpus B as a whole, or comparability between individual sections (sub-corpora) within the corpus;
   (b) *Document-level* comparability—between different documents within or across corpora, for example Lee et al. (2005);
   (c) *Paragraph- and sentence-level* comparability—between structural and communicative units within or across individual documents, for example Li et al. (2006);
   (d) *Comparability of sub-sentential units*—between clauses, phrases, multiword expressions, lexico-grammatical constructions;
   (e) *Cross-level comparability*—between units that are on different levels, for example a text and a corpus on the same topic, or a set of phrase queries and a document collection. This comparability is relevant for the tasks of bootstrapping larger collections of specialised linguistic resources from smaller samples, for example seed texts, terminological or phrase lists, etc.
2. The *degree of comparability* dimension, which characterises a level of closeness between the units of comparison, using some pre-defined scale. The values on the scale can be real numbers, for example ranging from 0 to 1, or a set of discrete values, for example for the case of multilingual texts ranging from close translations, through free translations of the same text, different texts about the same event, different texts on the same topic, texts within the same domain and finally to completely unrelated documents. For monolingual texts, the scale can define very close texts as those that were not produced independently, for example a plagiarised text; and texts with lower degrees of comparability as those sharing the same topic, subject domain, genre, style. Discrete labels on the comparability scale are interpretable in terms of task-related concepts, which makes them more informative compared to real-number values. However, in many cases, it is difficult to precisely define these concepts and devise a consistent classification method, since the label categories often have to rely on intuitive linguistic concepts (such as style, genre, subject domain, communicative intention) without an agreed definition or method of applying the scale classifications to real-world data.

Our research on MT for under-resourced languages focuses on cross-lingual comparability. For this application, we suggest a discrete comparability scale defined in terms of granularity of linguistic units that can be aligned to each other across different languages. The alignability criterion has an advantage of making clear-cut categorical distinctions between different types which are easy to apply for text classification and to check consistently. The suggested classification relies much less on vaguely defined linguistic categories.

In the cross-lingual context, we can distinguish the following broad categories:

- *Parallel corpora*, which can be aligned at the sentence and phrase levels (these consist of translated documents, e.g. corpora collected from multilingual news websites)

- *Strongly comparable corpora*, which can be aligned at the document level (these consist of texts describing the same event or subject, where alignment at the document level is still possible, e.g. linked Wikipedia articles in different languages, news stories on the same event)
- *Weakly comparable corpora*, which can be aligned on the level of sub-corpora according to domain, sub-domain or topic similarities (these consist of texts in the same domain or genre, but describing different events or areas; document alignment is usually not possible, e.g. a collection of British and German laws on immigration policy)
- *Non-comparable corpora*, which cannot be aligned across different languages

Other applications of comparable corpora will require different definitions and ways of identifying the intended degree of closeness between corpora, texts, etc.

Another task is identifying formal features in compared units that can be used for automatically predicting their levels of comparability. These features need to be derived in a systematic way and they can either be used for training classifiers, or for mapping comparability onto a numerical scale. Different metrics often use similar sets of features and similar methods of calculating the scores. For example, both monolingual and cross-lingual comparability in terms of subject domains typically rely on lexical features, weighted or filtered by frequency, textual salience of key terms, etc. Often, the only difference is that in the case of cross-lingual metrics, lexical features (words) are mapped to words in another language using bilingual dictionaries or Machine Translation (MT) systems, while in monolingual applications, lexical features are compared directly.

Due to its wide range of possible applications, the task of measuring comparability has increasingly become a core technological challenge in computational linguistics that needs to be developed and evaluated systematically.

Many applications now require not just *it-looks-good-to-me* comparable corpora, but corpora with controlled and benchmarked levels of comparability according to certain criteria. Comparability metrics are used not only for reporting scores of closeness between corpora, but also for collecting additional texts to make a corpus bigger, or to filter out unwanted texts from corpora to ensure the intended level of comparability. From this perspective, it is important to understand how reliable a particular metric is and to what extent it matches its specifications in its ability to *evaluate* comparability of corpora or individual texts.

In this chapter, we describe the application of our approach for three core tasks in measuring comparability: (1) systematic *meta-evaluation* (or calibration) of comparability metrics which use different sets or configurations of features on the basis of parallel corpora—Sect. 2.2; (2) development of *feature-selection techniques* for the metrics on the basis of strongly-comparable corpora—Sect. 2.3 and (3) application of the developed metrics to the tasks of MT for under-resourced languages and measurement of their effectiveness for corpora with unknown degree of comparability—Sect. 2.4.

This approach was implemented and used in the framework of the ACCURAT project for the scenario of enhancing the performance of MT systems for under-resourced languages. However, it potentially has wider applications, since it exemplifies core stages in the development, evaluation and use of comparability metrics.

2.2 Development and Calibration of Comparability Metrics on Parallel Corpora

In this section, we propose a method for meta-evaluating comparability metrics with different configurations of features. We show that our method gives consistent results for the same metrics on different datasets, which indicates that it is reliable and can be used for selecting a best-performing metric, or for finding the most efficient settings of parameters for parametrised metrics. We also describe our application domain which requires corpora with controlled levels of comparability.

2.2.1 Application of Corpus Comparability: Selecting Coherent Parallel Corpora for Domain-Specific MT Training

Traditionally, statistical and example-based MTs have relied on parallel corpora (collections of texts translated by human translators) to train statistical translation models and automatically extract translation equivalents. However, a serious limitation of this approach is that translation quality is impaired where parallel resources are not available in sufficient volume.

Firstly, it has been shown that improving translation quality at a constant rate requires an exponential increase in the training data (e.g. Och and Ney 2003: 43); that is if improving some MT evaluation score, for example BLEU, by one point requires doubling the size of an initial training corpus, then further improvement by one additional point requires a corpus four times bigger than the initial corpus, etc. This dependency imposes fundamental limitations on translation quality even for well-resourced languages, such as English, German or French, where only the huge datasets used by engines like Google Translate produce relatively good quality (and even then, only for certain text types). Smaller and less resourced languages do not have the benefit of such data repositories which results in a much lower MT quality.

Secondly, training translation models and language models for SMT has been shown to be domain-dependent to a much greater degree than rule-based MT (RBMT) (Eisele et al. 2008; Xu et al. 2007). If an SMT engine is trained on a corpus that doesn't match the domain of the translated text, then the quality for such out-of-domain translation becomes much lower. In practice, this means that for more narrow subject domains and text types, SMT cannot produce acceptable translation quality without domain adaptation which needs correspondingly highly specific parallel textual resources.

For translation to and from under-resourced languages in narrow domains and for specific text types, the two problems described above are combined. As a result, traditional ways of building SMT engines with acceptable translation quality are often not possible for many domain/language combinations.

There is, therefore, a need to develop a fine-grained monolingual domain selection and domain control mechanism for evaluating comparability of corpus sections

that can usefully be added to any SMT-training corpus (comparability here is measured monolingually—either on the source or on the target side). The methodology should allow MT developers to balance the size of the corpus to be built and its internal consistency, in terms of how newly added sections match the originally intended subject domain.

2.2.2 Methodology

Methodologies for computing the comparability of sections of an MT training corpus are usually based on calculating the degree of overlap between the two sections in terms of simple word tokens or at more advanced levels of linguistic annotations, such as lemmas (dictionary forms of words), combination of lemmas and part-of-speech codes, translation probabilities for each of the words, etc. Note that there are several major challenges for the efficient calculation of this overlap.

Firstly, calculated scores for comparability should be consistent with human intuition about closeness between the two sections, and what constitutes the subject domain at different levels of granularity: for example the broader domain of computer hardware vs. the more narrow domain of network technologies, documentation for different types of network cards, etc. This is required if user needs for finer- or coarser-grained domains are to be adequately addressed for most types of projects.

Secondly, for practical applications, the number of calculations between compared sections can be very large; so, the calculation method should be fast enough to produce the results in real time.

Thirdly, comparison often needs to be done between corpus sections of different sizes, so the calculation method should be minimally affected by the size of the compared sections or texts.

Ideally, comparison should also take into account both source and target parts of new additions to corpora, and evaluate not only monolingual distance, but also the 'translation' distance (which could mean that the same translation equivalents are used, and that terminology is translated consistently across the selected updates).

2.2.2.1 Description of Calculation Method

The method that we use in our experiments is based on the work of Kilgarriff (2001). This method was initially developed for purposes of linguistic analysis, that is to find words that are substantially different in two corpora: for example a corpus of spoken vs. a corpus of written English. But one of the side effects of this method is that it can produce a single numeric value that shows the 'distance' between the two compared monolingual corpora.

In our work, we focus not on identifying individual words which are used differently in different corpora, but on general quantitative measures of comparability between them. The method can be summarised as follows.

For each corpus (on the source or target side), we build a frequency list, and take the top 500 most frequent words (including function words).

Since corpora can be of different sizes, we use relative frequency (the absolute frequency, i.e. the number of times each word is found, divided by the total number of words in the corpus).

We compare corpora pairwise using a standard chi-square distance measure:

$$\text{Chi}^2 = \sum_{w_1}^{w_{500}} \frac{\text{Freq}(\text{corpus}^{\text{A}}) - \text{Freq}(\text{corpus}^{\text{B}})}{\text{Freq}(\text{corpus}^{\text{A}})}$$

2.2.2.2 Symmetric vs. Asymmetric Calculation of Distance

The challenge for this method is that some words that are in the top-500 list for CorpusA may be missing from the top-500 in CorpusB and vice versa. If the algorithm encounters the missing word, then it just adds its relative frequency to the value of the chi-square distance.

Obviously, exactly the same number of words is missing from the top-500 in CorpusA and in CorpusB. However, the sum of relative frequencies for these words can be different, for example it is possible that on average, more frequent words will be missing from CorpusA, and less frequent from CorpusB.

Therefore, if we compute the chi-square distance from CorpusA to CorpusB, and then from CorpusB to CorpusA, we will get different values which shows that the term 'distance' (used both in a mathematical and in everyday sense) is not exactly right for describing the values: our calculation method is asymmetric.

Instead, Kilgarriff (2001) uses a symmetric calculation: he takes into account only words which are common to both corpora, and goes down the frequency lists as far as is needed to collect the 500 most frequent common words. This method always returns the same value for distance regardless of the direction it is computed. However, the symmetric approach has its drawbacks: missing words do not contribute to the score directly (only by virtue of occupying 'someone else's place'); also it is harder to select the initial list for comparison: in a bad case, it could be necessary to start with, say, the top-1000 words for each of the corpora, and then to remove mismatches. It may take slightly longer to do these calculations, and in real time, this may result in unnecessary delays and increased waiting time for users.

In our approach, we make two independent asymmetric calculations, one in each direction: CorpusA → CorpusB and CorpusB → CorpusA, and get two chi-square scores.

However, now it is not obvious what is the best way to combine these two scores into a single measure of distances between the corpora: one approach would be to take the average of the distances; another is to take the minimum distance as the value.

We compare these two possibilities experimentally in later sections, and show that the minimum of the two chi-square scores computed for each of the two directions gives better and more meaningful results.

2.2.2.3 Calibrating the Distance Metric

We used corpora available from TAUS (Translation Automation User Society) in its TDA (TAUS Data Association) repository. The corpora there were initially annotated by data providers in terms of 'subject domains', which are identified manually at the upload stage. The idea is that the metric should be able to simulate identification of these domains automatically.

We calculated the distance between different sections of the TDA repository—individual uploads and collections of uploads grouped by the same data provider and domain. In order to tell whether the metric intuitively makes sense, we checked whether there is an agreement between the resulting values and the labels provided by the TDA members.

In our experiment, we focussed on the English (US and UK)–French (France) language pair. We selected the set of uploads in a way that covered different combinations of domains and data providers: some corpora are labelled as belonging to different domains, but were produced by the same company. Some were produced by different companies but were labelled with the same domain tag.

These labels were used as a benchmark for judging the quality of the lexical comparability metric. We aimed at giving the smallest distance score to corpora within the same subject domain.

The results of measuring comparability between sections of the corpus given by different data providers are presented in Fig. 2.1. Different shades of grey illustrate different ranges of distances: the closer the distance, the darker the colour.

It can be seen from Fig. 2.1 that the metric reliably identifies the following:

1. All corpora within the 'computer hardware' domain are reliably grouped together, and distinguished from other domains.

	compHardB	compHardA	compHardC	compSoftJ	compSoftK	compSoftC	compSoftG	compSoftF	consumerElectronicsD	financialE	legalEC	hansardEuroparl	legalUN	pharmBiotechH	profBusinessK	profBusinessG	profBusinessD
compHardB		0.18	0.21	0.42	0.47	0.5	0.41	0.28	0.24	0.26	0.44	0.8	0.91	0.38	0.51	0.43	0.25
compHardA	0.18		0.33	0.4	0.52	0.49	0.49	0.27	0.29	0.27	0.5	0.65	0.77	0.45	0.54	0.41	0.29
compHardC	0.21	0.33		0.45	0.44	0.5	0.48	0.32	0.35	0.37	0.44	0.74	1.01	0.45	0.47	0.55	0.36
compSoftJ	0.42	0.4	0.45		0.6	0.64	0.53	0.39	0.46	0.46	0.53	0.64	0.83	0.46	0.67	0.59	0.47
compSoftK	0.47	0.52	0.44	0.6		0.81	0.43	0.56	0.49	0.48	0.85	0.82	1.5	0.63	0.03	0.68	0.5
compSoftC	0.5	0.49	0.5	0.64	0.81		0.64	0.54	0.51	0.56	0.68	0.83	1.03	0.44	0.64	0.73	0.52
compSoftG	0.41	0.49	0.48	0.53	0.43	0.64		0.44	0.47	0.52	0.56	0.9	1.12	0.55	0.46	0.79	0.48
compSoftF	0.28	0.27	0.32	0.39	0.56	0.54	0.44		0.37	0.41	0.56	0.58	0.95	0.48	0.62	0.66	0.38
consumerElectronicsD	0.24	0.29	0.35	0.46	0.49	0.51	0.47	0.37		0.27	0.44	0.72	0.94	0.38	0.53	0.38	0.01
financialE	0.26	0.27	0.37	0.46	0.48	0.56	0.52	0.41	0.27		0.44	0.67	0.79	0.4	0.51	0.35	0.27
legalEC	0.44	0.5	0.44	0.53	0.85	0.68	0.56	0.56	0.44	0.44		0.74	0.73	0.43	0.93	0.7	0.45
hansardEuroparl	0.8	0.65	0.74	0.64	0.82	0.83	0.9	0.58	0.72	0.67	0.74		0.87	0.6	0.92	0.84	0.72
legalUN	0.91	0.77	1.01	0.83	1.5	1.03	1.12	0.95	0.94	0.79	0.73	0.87		0.88	1.48	0.8	0.91
pharmBiotechH	0.38	0.45	0.45	0.46	0.63	0.44	0.55	0.48	0.38	0.4	0.43	0.6	0.88		0.68	0.66	0.38
profBusinessK	0.51	0.54	0.47	0.67	0.03	0.64	0.46	0.62	0.53	0.51	0.93	0.92	1.48	0.68		0.72	0.53
profBusinessG	0.43	0.41	0.55	0.59	0.68	0.73	0.79	0.66	0.38	0.35	0.7	0.84	0.8	0.66	0.72		0.38
profBusinessD	0.25	0.29	0.36	0.47	0.5	0.52	0.48	0.38	0.01	0.27	0.45	0.72	0.91	0.38	0.53	0.38	

Fig. 2.1 Chi-square distances between different data providers (Labels indicate domain and owner, e.g. compSoftG is the label 'computer software' produced by the company G)

2. Some of the corpora which were produced by the same companies 'D' and 'K' were reliably grouped together, even though the corpora had received different human labels: company D—'consumer electronics' and 'professional business services'; company K—'computer software' and 'professional business services'. These instances can be explained by inconsistency in assigning labels to corpora that essentially represented the same domain.
3. Different domains which are intuitively close to each other were also grouped together: 'computer hardware' and 'consumer electronics', and then at some distance—several corpora on computer software.

However, there are several problems with the presented distances and grouping:

1. The 'computer software' and 'legal' domains are not coherently grouped. A possible reason is a greater variety of sub-domains within the 'computer software' domain (it may describe more 'products', and have a more diverse lexical profile).
2. The 'pharmaceutical and biotechnology' and 'financial' domains are not sufficiently distinct from the 'software' and 'hardware' domains.

Still these problems can be attributed to inconsistencies in human labelling, as well as to shortcomings of the metric itself. Symbolic labels are speculative in their nature, and do not capture the inner structure or diversity of the domain; at present, human annotation offers no way of dealing with mislabelled data.

2.2.3 Validation of the Scores: Cross-Language Agreement for Source vs. Target Sides of TMX Files

We validate our choice of metric by comparing different versions of Kilgarriff's metric for computing the distance between corpora. As we indicated, there are two possibilities for combining asymmetric chi-square distances: we can either take the average of the two different values or the minimum of the two values.

Sections of corpora in the TDA repository are uploaded in TMX (Translation Memory Exchange) format, which is an XML file with sentence-or segment-aligned parallel corpora.

The idea for comparison is the following: we use each of the possibilities on the source side and on the target side of the same TMX files and then compare how the scores 'agree' with each other. The agreement can be measured using a standard statistical method for calculating correlation, like Pearson's r correlation coefficient: if there is a good agreement, r is closer to 1 or to -1; if there is no agreement, r is closer to 0.

To get more data points for more reliable calculation of correlation, we further split sections of the corpora presented in Fig. 2.1 into individual uploads, for example for Hardware Company A, we had five individual TMX files. Distances were computed at this finer granularity between all individual uploads.

If one of the compared metrics produces a higher correlation, then this means that the results obtained on the source side are more consistent with the results obtained on the target side, and the metric is more meaningful. Essentially, we know from the start that the two texts come from the same TMX file, but the metric doesn't have that information. The better it can figure this out, the more reliable it is.

Table 2.1 compares the *r* correlation figures for individual uploads. It can be seen from the table that the minimum distance has the best correlation between source and target sides of TMX files: $r = 0.85$. Thus, minimum distance can be viewed as a more reliable metric than average distance or the third score we computed, one-direction distance.

Figures 2.2 and 2.3 further illustrate this difference: they compare the correspondence between the TMX distances for source text (horizontal axis) vs. distances for target texts for the same uploads (and illustrate the correlation figures presented above). Figure 2.2 indicates the distances in terms of minimum chi-square scores for TMX-A → TMX-B vs. TMX-B → TMX-A. Figure 2.3 indicates average chi-square scores for the same pairs of distances.

It can be seen that the minimum distances have a much better correlation between source and target, so they more reliably indicate whether the texts are indeed closer to each other.

This method offers a way to evaluate different comparability scores: the more the source and target agree with each other, the better the quality of the matching scores is. This evaluation method is based on the assumption that if the texts are close in

Table 2.1 Pearson's *r* correlation for distances computed for English vs. French

Metric	*r*-Correlation
Minimum distance	0.85
Average distance	0.67
One-direction distance (A→B and B→A)	0.61

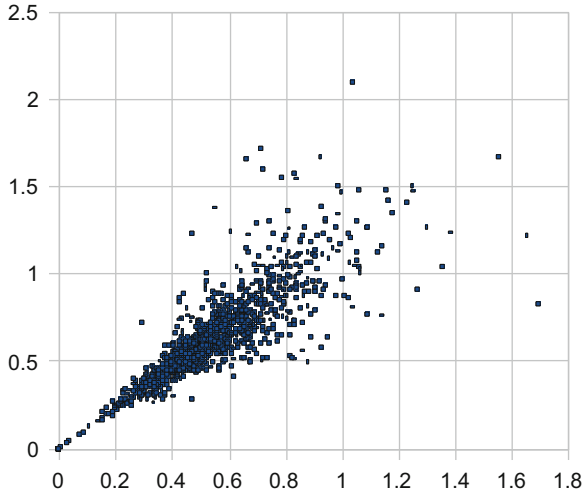


Fig. 2.2 Minimum chi-square distance (x = En; y = Fr)

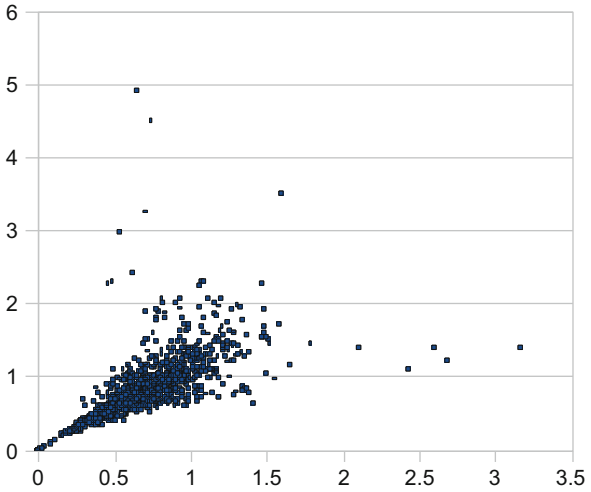


Fig. 2.3 Average chi-square distance (x = En; y = Fr)

Table 2.2 Pearson's r correlation for English vs. French on larger corpus sections

Metric	r-Correlation
Minimum distance	0.88
One-direction distance (A→B)	0.72
One-direction distance (B→A)	0.86

terms of the source side, the scores should also show that they are close in terms of the target side.

However, there is a question of how to interpret divergences of the dots from the diagonal line. One explanation is that the quality of the matching scores is not so good. But another explanation suggests that some inconsistent translation equivalents are used across the upload in the target, so even if the documents are genuinely close on the source side, they become divergent on the target side. These issues require a more careful look into the compared data.

To verify that our meta-evaluation method provides consistent results for different sizes of evaluated corpus, we repeated the experiment for the same language pair, but now, we used the original joined TMX files, where all uploads are grouped together for the same data provider. Correlation coefficients for this setting are presented in Table 2.2, which measure an agreement for minimum chi-square scores for TMX-A → TMX-B vs. TMX-B → TMX-A.

The highest correlation is again found between minimum scores across the two directions which confirms our choice of this metric as the most reliable.

These results show that data with a different number of data points obtained on sections of different sizes point to the same metric as the best one which indicates that our proposed method is internally coherent.

2.2.4 Discussion

In this section, we proposed a method for meta-evaluation of comparability metrics using correlation between source and target sides of parallel corpora. We used a collection of parallel corpora available from the TDA repository. Comparability metrics need to be calibrated on a diverse parallel corpus that includes sections with several distinct, annotated subject domains, genres, etc. However, after successful calibration, such comparability metrics can be further used to collect monolingual and bilingual comparable corpora, without the need to have expensive parallel resources. Pearson's r coefficient, which we calculate during calibration, gives an indication of how reliable the metric is and how much noise might occur in the data.

The metric which was found to perform best in our experiment (minimum Chi-Square distance between the compared top-500 words of frequency lists) has relatively high agreement for data generated on the source and target sides of TMX files ($r = 0.85$) which indicates the upper limit of the metric's reliability.

However, the applicability of the proposed meta-evaluation method is limited by the accuracy and completeness of translations in the parallel corpus used for calibration: gaps or excessively free translation can result in shifts in feature patterns, so distances between different domains calculated on source and target texts can become greater. Another potential limitation of the method is its reliance solely on those formal parameters that can be computed in a language-independent way and do not vary substantially across languages. Language-specific differences, for example variations in type-token ratio (due to different morphological structure of languages), can potentially lead to differences in feature patterns and, as a result, lower correlation figures.

2.3 Exploration of Comparability Features in Document-Aligned Comparable Corpora: Wikipedia

2.3.1 Overview: Wikipedia as a Source of Comparable Corpora

Wikipedia has been mined for various linguistic purposes because of the diversity and richness of information available in a variety of languages (Tomás et al. 2008). In addition, the presence of inter-language links, which connect documents from different languages describing the same topic, makes Wikipedia a useful multilingual resource, for example as a source of comparable documents. However, although articles written in different languages on the same topic could be considered comparable (Gamallo and López 2010), the degree of similarity may vary widely. Parts of the content could be translation equivalents (i.e. parallel); other parts may have been developed independently and share little thematic or lexical overlap. For tasks such as Cross-Language Information Retrieval (CLIR) or Statistical MT (SMT), the degree of similarity between texts will affect the quality of translation

resources subsequently created; using non-similar documents will introduce noise and reduce MT performance (Lu et al. 2007).

Different measures have been developed to measure the similarity between Wikipedia articles in different languages (see Sect. 2.3.2) which can be used to filter out non-similar documents. However, little past work has analysed whether or not these methods correlate with human assessments across multiple languages. In this work, we have collected manual judgements on Wikipedia articles in various language pairs which include seven under-resourced languages. We analyse the judgements gathered for inter-assessor agreement and compare the judgements with two machine-computed measures of document-level similarity: one based on language dependent features, the other on language-independent features. Being able to reliably measure the similarity of Wikipedia articles across languages is helpful if Wikipedia is to be used as a source of comparable data.

Section 2.3.2 provides a summary of past work on comparing the similarity between Wikipedia articles; Sect. 2.3.3 describes the methodology used in our experiments including the creation of human cross-language similarity judgements; Sect. 2.3.4 discusses results obtained from comparing Wikipedia articles across languages; Sect. 2.3.5 provides a discussion of results; Sect. 2.3.6 concludes this part of the chapter and provides directions for further research.

2.3.2 *Previous Work on Using Wikipedia as a Linguistic Resource*

Wikipedia is often viewed as a promising source of comparable documents as pairings of similar (or near similar) documents in different languages are provided through the inter-language links (Otero and López 2010). However, Wikipedia articles on the same topic are not necessarily equivalent to each other and, in some cases, the entry description may even contain information that is contradictory (Filatova 2009). Nevertheless, Wikipedia does contain a rich seam of information that can be mined. For example titles from articles connected by inter-language links have been extracted and used as the source of bilingual lexicons, enabling parallel sentences within connected articles to be identified without the use of any other linguistic resources (Adafre and de Rijke 2006; Erdmann et al. 2008; Tomás et al. 2008). Smith et al. (2010) also used Wikipedia as a source of similar or comparable sentences but instead used the image captions. Lin et al. (2011) mined information found in the infoboxes to gather named entities (NEs) and other information in different languages.

Adafre and de Rijke (2006) developed a method to retrieve parallel sentences from Wikipedia documents by using information about the overlap of anchors. Smith et al. (2010) further developed this idea by using additional features, such as sentence length and longest aligned/unaligned words to produce a binary classifier trained on parallel corpora. Bharadwaj and Varma (2011) also developed a binary sentence classifier for English–Hindi that does not require parallel corpora or other

linguistic resources. They first indexed the content of documents for each language separately, treating each sentence as a bag-of-words and creating separate indexes for each language. To identify whether a sentence pair was parallel or not, they performed retrieval for each sentence from the appropriate index, that is English sentences queried on the English index; Hindi sentences queried on the Hindi index. Different features were then extracted, such as the intersection and union of retrieved articles and sentence lengths. They report that the binary sentence classifier is able to identify parallel sentences with an accuracy of 78%.

Several methods have also been used to assess the accuracy of extracted information from Wikipedia. For example, Yu and Tsujii (2009) conducted human evaluation to assess the accuracy of extracted parallel phrases; whilst Smith et al. (2010) and Adafre and de Rijke (2006) conducted similar evaluations at the sentence level. Comparable corpora are mostly evaluated by calculating the improvement of MT performance (Munteanu and Marcu 2005).

However, despite the continued interest in Wikipedia, there seems to be little work on comparing similarity at the document level. One paper that does consider document-level similarity attempts to identify parallel documents from Wikipedia (Patry and Langlais 2011). The method first retrieves candidate document pairs using an IR system. Parallel documents are identified using lightweight content-based features extracted from the documents, such as numbers, words only occurring once (hapax legomena) and punctuation marks. They report that the resulting classifier can correctly identify parallel and noisy parallel documents with an accuracy of 80%.

Much of the previous work has been conducted based on English Wikipedia. However, given the variation in size and interconnectivity of Wikipedia in different languages, the performance of similarity measures is likely to vary (particularly for languages where there exist limited translation resources). Our work aims to address this and provide empirical evidence, demonstrating the success of measuring cross-language similarity between different language pairs. In addition, to the best of our knowledge, there has been little or no research on comparing automatically derived similarity scores and human judgements.

2.3.3 Methodology

2.3.3.1 Document Pre-processing

Articles from dumps of Wikipedia[1] were downloaded for seven under-resourced language pairs[2] and articles linked through inter-language links were extracted using JWPL (Zesch et al. 2008). In these experiments, we used the following language pairs:

[1]Data downloaded March 2010: http://dumps.wikimedia.org/

[2]Providing translation resources for under-resourced languages is the goal of the ACCURAT (http://www.accurat-project.eu/) project within which this study was carried out.

Table 2.3 Size of initial Wikipedia datasets

Lang	Number of documents		Number of entries in bilingual lexicon
	Total	Linked to EN	
DE	1,036,144	637,382	181,408
EL	49,275	36,752	28,294
ET	72,231	42,008	22,645
HR	81,366	51,432	26,804
LT	102,407	57,954	41,497
LV	26,297	21,302	15,511
RO	141,284	97,815	35,774
SL	85,709	51,332	25,101

Croatian–English (HR–EN), Estonian–English (ET–EN), Greek–English (EL–EN), Latvian–English (LV–EN), Lithuanian–English (LT–EN), Romanian–English (RO–EN), and Slovenian–English (SL–EN). All of these language pairs have limited translation resources available and would benefit from language-independent methods of assessing cross-language similarity. We also included one additional pair, German–English (DE–EN), to compare performance against a language pair that is well-resourced and for which high-quality translation resources are available.

Wikipedia articles were pre-processed with information such as infoboxes, images, tables, etc. filtered out. Plaintext only from the main body of Wikipedia articles was extracted and used as the basis for human cross-language similarity judgements.

Table 2.3 shows the statistics of the Wikipedia dumps used in this study. The second column shows the total number of articles in each language. The third column shows for each language the number of articles that are linked to an English article on the same topic by an inter-language link. The last column shows the number of entries in the bilingual lexicon used in the similarity measures described in Sect. 2.3.3.2.

2.3.3.2 Similarity Measures

Two approaches for assessing document-level similarity between Wikipedia articles written in different languages were investigated: a language-independent approach based on using a bilingual lexicon derived from Wikipedia (referred to as *Anchor+word overlap*); a second approach that involved translating all non-English documents into English using available MT systems[3] (referred to as *Translation*). The latter approach enabled comparison with machine translation; however, in practice, it is not viable due to the limited availability of translation resources.

The first approach, similar to Adafre and de Rijke (2006), determines sentence similarity by measuring overlap of anchor texts and cognates (e.g. numbers, dates

[3]Bing Translate was used to translate all document pairs apart from HR–EN, which was translated using Google Translate.

and named entities), which appear as the same text string in different language versions of the text (see example in Fig. 2.4). To translate the anchors, we start by extracting all document titles (typically nouns, named entities or phrases) that are connected using inter-language links and using them to build a bilingual title lexicon for each language pair (e.g. 'asteroidov' ⇔ 'asteroid' for Slovenian and English). We then use the lexicon to translate all anchor texts in the non-English Wikipedia article into English. We measure the proportion of overlapping terms using Jaccard coefficient; each sentence is treated as a binary vectors (or set) such that only token types are counted. Figure 2.4 shows an original non-English article (in Slovenian) where anchor texts are shown in bold.[4] Using the bilingual lexicon, the anchor texts are replaced with their English equivalent. The Slovenian text is then compared for the overlap of terms with the equivalent English article where cognates (e.g. numbers in Fig. 2.4) are also compared. The second approach also measures overlap of terms at sentence level, but instead of using anchor+word overlap, it measures term overlap between the original English text and the English translation of the non-English text.

In both approaches, we perform a pairwise comparison between all sentence pairs, allowing a *1:1* and *M:1* correspondence between sentences in both articles. We first split documents into sentences, and for each sentence in the shorter Wikipedia article, we calculate its similarity with all sentences in the longer document. The sentence is paired with the sentence(s) receiving the highest similarity score(s). Multiple sentences in the shorter article may be paired to the same sentence in the longer document. This accommodates cases in which simpler sentences are combined into an equivalent complex sentence (an example of this is also shown in Fig. 2.4). A minimum similarity threshold is set, below which sentence pairs are ignored. This threshold was set based on manual inspection of aligned sentence pairs. The local similarity scores (the similarity scores between sentence pairs) are combined into a global (or document-level) score by computing the mean value of all aligned sentence pair scores normalised by the number of sentences in the shorter document.

2.3.3.3 Eliciting Human Judgements

To select articles for human inspection, we first sorted all Wikipedia articles for a given language pair by their overall anchor and word overlap similarity score (Sect. 2.3.3.2). The scores were divided into 10 bins, and we randomly selected 10 document pairs from each bin,[5] resulting in a total of 100 documents per language pair. This initial selection process was undertaken to provide a range of article pairs with

[4]Note: the text in bold that appears with a 'l' character separating terms represents the referred article title and the document text as it appears to the user.

[5]When this was not possible (i.e. fewer than 10 document pairs were found in a bin), the maximum number of document pairs in that bin was chosen for the evaluation set and a higher number of documents were chosen from the lower bins to achieve the total number of 100 document pairs.

Original Slovenian text (anchor texts in bold)

*Večinajih je v bližini**[[družinaVesta|asteroidnedružineVesta]]**.*

*Imajopodobne**[[izsrednost|izsrednosti]]**, todanjihova**[[elipsa|velikapolos]]**leži v območju od 2,18**[[astronomskaenota|a. .e.]]** do 2,50 a. e. (kjer je **[[Kirkwoodo-vavrzel|Kirkwoodovavrzel]]** 3 : 1).*

Slovenian text with anchors replaced with English (bold)

*Večinajih je v bližini**[[vesta family]]**.*

*Imajopodobne**[[eccentricity]]**, todanjihova**[[ellipsis]]**leži v območju od 2,18**[[astronomical unit]]** do 2,50 a. e. (kjer je **[[kirkwood gap]]** 3 : 1)*

Equivalent English article (matches in bold)

*A large proportion have orbital elements similar to those of 4 Vesta, either close enough to be part of the **[[vesta family]]**, or having similar **[[eccentricity (orbit)]]** and **[[inclination]]**s but with a **[[semi-major axis]]** lying between about **2.18[[astronomical unit]]** and the **3:1[[kirkwood gap]]** at **2.50** AU.*

Fig. 2.4 Example anchor text translation

different similarity scores to include in the test data. In total, 97% of articles included in the dataset contain fewer than 1000 tokens to ensure judges were able to read and digest the articles in a reasonable time and to limit assessor fatigue. Table 2.4 provides a summary of the documents used for human similarity judgement.

Given a pair of Wikipedia articles in a language pair, we asked assessors to read the articles and answer the four questions[6] shown in Fig. 2.5. Assessors (16 in total) were all native speakers of the non-English language and fluent speakers of English. We used a 5-point Likert scale for all questions. For the questions regarding an assessment of document-level similarity (and comparability), we did not provide descriptions for each level. A general definition of similarity is complex (Hatzivassiloglou et al. 1999); therefore, by using a scale and asking assessors to comment on characteristics they felt contributed to their judgement of similarity (see Q1), we can better understand what characterises cross-language similarity between Wikipedia articles (see Sect. 2.3.5). All the judges were from partner sites in the ACCURAT project and therefore had a reasonable degree of knowledge about

[6]The questions were based on a prior pilot study in which 10 assessors assessed 5 document pairs and gave comments on the evaluation scheme and decisions regarding their assigned similarity score.

Table 2.4 Summary of documents used for human similarity judgements

Total number of documents	1600 (800 pairs)
Number of languages	9 (DE, EL, EN, ET, HR, LT, LV, RO & SL)
Average number of words per document	450.59 (min: 107, max: 1546)
Average number of sentences per document	51.31 (min: 22, max: 1028)

Q1. How similar are these two documents?
○ 1 (very different) ○ 2 ○ 3 ○ 4 ○ 5 (very similar)
Why did you give this similarity score (please tick all relevant ones):
☑ Documents contain similar structure or main sections
☐ Documents contain overlapping named entities
☐ Fragments (e.g. sentences) of one document can be aligned to the other
☐ Content in one document seems to be derived or translated from the other
☐ Documents contain different information (e.g. different perspective, aspects, areas)
☐ Others, please mention: .....

Q2. What proportion of overall document contents is shared between the documents?
○ 1 (none) ○ 2 ○ 3 ○ 4 ○ 5 (all)

Q3. Of the shared content (if there is any), on average how similar are the matching sentences?
○ 1 (very different) ○ 2 ○ 3 ○ 4 ○ 5 (very similar)

Q4. Overall, what is the comparability level between these two documents?
○ 1 (very different) ○ 2 ○ 3 ○ 4 ○ 5 (very similar)

Fig. 2.5 Evaluation sheet completed by human assessors for each document pair

comparability and similarity. The documents and human judgements are available for public download.[7]

2.3.4 Results and Analysis

2.3.4.1 Responses to the Questionnaire

There is a significant correlation between the similarity level (Q1) assigned by the assessor and the level of comparability (Q4) (rho $= 0.873$; $p < 0.01$) and the similarity level (Q1) and the overall proportion of shared content (Q2) (rho $= 0.900$; $p < 0.01$), suggesting that the more overlap between information in article pairs, the greater is the perceived degree of similarity. There is also a significant correlation between the overall similarity score (Q1) and the similarity between matching sentences of the shared content (Q3) (rho $= 0.727$; $p < 0.01$).

Figure 2.6 shows the distribution of the similarity scores assigned to document pairs where multiple scores are averaged across the scores of the two assessors.

[7]Data and judgements are available for download at the website: ir.shef.ac.uk/cloughie/resources/similarity_corpus.html

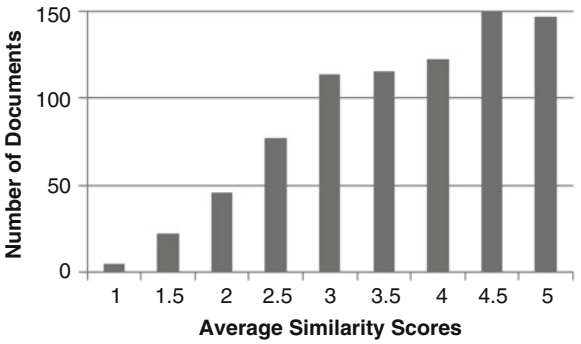


Fig. 2.6 Distribution of document-level similarity scores averaged across both assessors ($N = 800$)

For the 800 document pairs, 52.5% of document pairs are judged to exhibit a high degree of similarity (average score equals 4 or above), 28.8% judged to be moderately similar (average score between 2.5 and 3.5), and 18.8% judged to be different (average score of 2 or less). This confirms that articles in different languages on the same topic are not necessarily similar, and therefore, a suitable method to identify cross-language similarity is required. We explore in more detail what features judges use to derive their judgement of similarity in Sect. 2.3.5.

2.3.4.2 Inter-assessor Agreement

We report the agreement between assessors for each question in the evaluation task (shown in Fig. 2.5). Scores are calculated over the original 5-point scale[8] and also for a 2-point scale created by aggregating the results for scores 1–3 (low similarity) and 4–5 (high similarity). The values in parentheses represent the proportion of cases where assessors' scores are the same.

As shown in Table 2.5, assessors chose the same similarity score to represent document pairs 41% of the time. However, upon further inspection, we found that in 44% of cases, scores assigned to the document pairs differ only by 1 (14% by 2, 2% by 3 and 0.4% by 4). This is confirmed by the high level of agreement when considering scores combined into two classes. Here, the proportion of cases in which assessors give the same value rises to 73%.

Table 2.6 shows agreement between assessors for each of the questions on a 5-point scale broken down by language pairs. There is considerable variance across language pairs from 25% agreement (DE–EN) to 70% (SL–EN) for assigned similarity scores. Table 2.7 shows inter-assessor agreement when scores are combined into two classes. In all the cases, the agreement is markedly improved.

[8]Agreement for the five similarity levels is calculated using a weighted version of Cohen's Kappa, in which the order of classes is taken into account, e.g. similarity scores of 1 and 2 are in better agreement than scores 1 and 5.

Table 2.5 Inter-assessor agreement (% indicates the proportion of times assessors agree on the same value)

Question	Weighted Cohen's Kappa (5 classes)	Cohen's Kappa (2 classes)
Q1) Similarity	0.38 (41%)	0.43 (73%)
Q2) Proportion	0.47 (48%)	0.52 (77%)
Q3) Similar sentences	0.39 (50%)	0.42 (81%)
Q4) Comparability level	0.37 (48%)	0.46 (80%)

Table 2.6 Inter-assessor agreement (weighted Cohen's kappa) for each language pair (five classes)

Lang	Similarity	Proportion	Similar sentences	Comparability level
DE–EN	0.34 (25%)	0.45 (46%)	0.45 (52%)	0.33 (42%)
ET–EN	0.49 (57%)	0.49 (58%)	0.36 (45%)	0.44 (69%)
EL–EN	0.25 (43%)	0.36 (50%)	0.37 (56%)	0.41 (59%)
HR–EN	0.34 (28%)	0.38 (34%)	0.43 (51%)	0.25 (24%)
LT–EN	0.14 (19%)	0.31 (43%)	0.14 (27%)	0.08 (23%)
LV–EN	0.43 (45%)	0.36 (39%)	0.45 (51%)	0.31 (43%)
RO–EN	0.37 (37%)	0.33 (38%)	0.39 (48%)	0.52 (59%)
SL–EN	0.36 (70%)	0.62 (79%)	0.20 (72%)	0.30 (65%)

Table 2.7 Inter-assessor agreement (Cohen's kappa) for each language pair (two classes)

Lang	Similarity	Proportion	Similar sentences	Comparability level
DE–EN	0.58 (79%)	0.6 (80%)	0.44 (80%)	0.12 (70%)
ET–EN	0.71 (86%)	0.67 (84%)	0.42 (75%)	0.49 (98%)
EL–EN	0.14 (64%)	0.21 (64%)	0.27 (75%)	0.68 (88%)
HR–EN	0.22 (53%)	0.45 (70%)	0.35 (82%)	0.42 (82%)
LT–EN	0.16 (53%)	0.43 (71%)	0.35 (75%)	0.00 (61%)
LV–EN	0.55 (78%)	0.53 (78%)	0.49 (81%)	0.27 (83%)
RO–EN	0.39 (72%)	0.34 (69%)	0.57 (84%)	0.58 (85%)
SL–EN	0.71 (97%)	0.75 (97%)	0.21 (93%)	0.31 (71%)

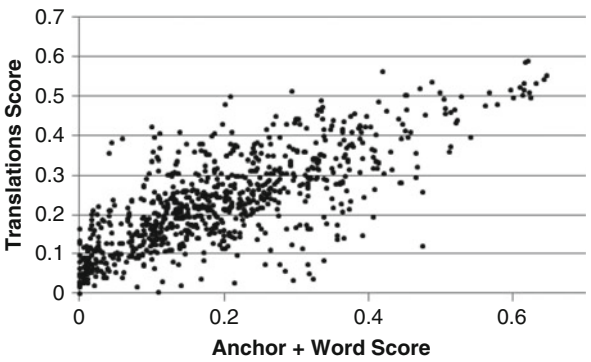


Fig. 2.7 Correlation between anchor+word overlap and similarity based on document translation

Table 2.8 Correlation (Spearman Rank, rho) between human judgements and similarity measures for five classes

Judgement set	Anchor+word overlap	Translation
Judgement 1	0.290	0.228
Judgement 2	0.321	0.323
Combined	0.353	0.325

Table 2.9 Correlation (Spearman Rank, rho) between human judgements and similarity measures and between similarity measures for five classes and across languages

	Correlation with human judgements		
Lang	Anchor+word overlap	Translation	Correlation between similarity measures
DE-EN	0.631	**0.703**	0.897
EL-EN	**0.124**	0.077	0.441
ET-EN	−0.045	**−0.001**	0.741
HR-EN	**0.495**	0.408	0.683
LT-EN	0.376	**0.512**	0.791
LV-EN	0.362	**0.497**	0.593
RO-EN	**0.279**	0.250	0.680
SL-EN	**0.417**	0.385	0.576

2.3.4.3 Correlation of Similarity Measures to Human Judgements

Section 2.3.3.2 described two approaches to compute cross-language similarity between document pairs: the first a language-independent approach; the second based on translation of non-English articles into English and computing monolingual similarity. Similarity values between the two approaches are highly correlated ($r = 0.744$, $p < 0.01$), showing that results obtained using language-independent features are comparable to results relying on translation resources. A scatter plot of the scores obtained from these two approaches for all document pairs is shown in Fig. 2.7.

Table 2.8 shows the correlation between similarity measures and sets of judgements: those from each assessor separately and combined (the mean score of each judgement rounded up to the nearest whole number). In these results, we use human judgements based on a 5-point Likert scale. From the results for the combined judgements, the anchor and word overlap approach shows a higher correlation (rho $= 0.353$, $p < 0.01$) than the approach based on translating non-English articles into English and computing word overlap (rho $= 0.325$, $p < 0.01$).

Table 2.9 shows the correlation of similarity scores with human judgements for each language pair. We observe that the correlation varies widely based on the language pair. For example human judgements for the DE–EN language pair correlate highly with both measures of similarity; however, the correlation for Estonian–English (ET–EN) is very poor. The anchor+word overlap measure of similarity has higher correlation than the translation approach with the human similarity judgements for 4/8 of the language pairs. This is a positive result, given

that the result is obtained using a language-independent approach making use of only a bilingual lexicon derived from Wikipedia. From Table 2.9, we also find that the correlation between the similarity scores overall is good but, again, varies depending on each language pair. For example correlation is lowest for EL–EN, which may suggest poorer MT results for Greek to English.

2.3.4.4 Classification Task

In this section, we compare the two approaches for measuring similarity based on using the scores from each approach as features in a classification task. For each document pair, we round up the average assigned similarity scores to the nearest class: for example a document pair with average score of 4.5 is included in class 5. Using a Naïve Bayes classifier[9] and threefold cross-validation, we are able to classify 40% of the 800 cases correctly using the scores from anchor+word overlap method (similar performance was achieved using the translation method).

By taking the Most Common Class (class 5, $N = 297$) as a baseline and then simply assigning all cases to this results in an accuracy of 37.1%, we find that many (36%) of the mis-classified cases are between classes 4 and 5. The classifier correctly classified 52.5%, 37.2% and 38.2% of document pairs in classes 5, 4 and 3, respectively. None of the document pairs in classes 1 and 2 were correctly classified, most probably due to the small number of available training documents (5 and 61 cases, respectively). These document pairs were incorrectly classified as class 3 instead. By combining human judgements into two classes, as described in Sect. 2.3.4.2, we can correctly classify 58% of cases using either similarity score (this represents 50.2% of similar documents and 66.8% of non-similar documents). Accuracy for the Most Common Class baseline is 52.5% (for the class 'similar').

2.3.5 Discussion

2.3.5.1 Features of 'Similar' Articles

As stated in the introduction, one of the goals of this study was to better understand what makes two Wikipedia articles written in different languages similar. The evaluation scheme has enabled us to analyse the characteristics of document pairs assigned each similarity score in more detail (Fig. 2.8).

When judging cross-language similarity, the judges were asked to provide input on what led them to make their decision. The options included whether the two articles contained a similar structure or ordering of the content (similar structure), whether documents contained overlapping NEs, whether fragments of text (e.g. sentences) from one document could be aligned to the other (overlapping

[9]In these experiments, we used the Weka Toolkit (version 3.4.13).

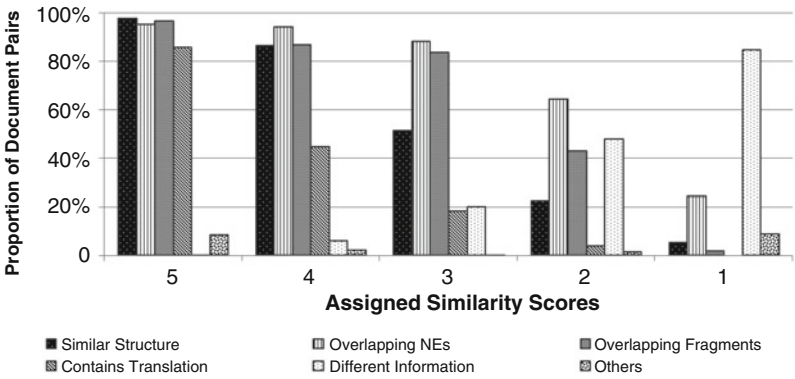


Fig. 2.8 Characteristics that capture various levels of similarity

fragments), whether content in one article appeared with equivalent translations in the other (contains translation), whether articles contained different information or were written from a different perspective (different information) and any other reasons. The results suggest that the majority of document pairs judged as highly similar (either a score of 4 or 5) in Wikipedia have the following characteristics: they contain similar structure, overlapping named entities, overlapping fragments and over 80% of these document pairs contain what appear to be translations of the content, that is translation equivalents.

Interestingly, the results also show that simply sharing named entities or having aligned segments of text does not guarantee that the overall document pair is similar. The latter could be the result of judges making document-level similarity assessments: a document pair may contain a number of aligned sentences, but at a document (or global) level, the degree of similarity is low. As expected, the number of articles containing different information increases for cases of little or no similarity (1–2). A distinguishing feature between text pairs which exhibit high similarity vs. those exhibiting little or no similarity would seems to be whether the content in the articles follows a similar structure and whether the document pairs contain translation equivalents of each other. To verify this, we created a binary feature vector for each of the human similarity judgements (1600) and the comments (5 features in total) and performed feature selection using threefold cross-validation on the 5 classes of similarity judgement and a measure of information gain.[10] The features are ranked for their discriminative power in the following order: contains translation, similar structure, different information, overlapping fragments and overlapping NEs.

Using binary feature vectors based on the comments to classify all 1600 cases, we achieve an accuracy of 57% (the Most Common Class baseline accuracy is 31.4%).

[10]We used weka.attributeSelection.InfoGainAttributeEval for feature selection and weka.attributeSelection.Ranker to rank the features from the Weka Toolkit.

When considering a 2-class problem (high/low similarity), we obtain an accuracy of 81%. In this case, the Most Common Class accuracy baseline is 60.3%. This suggests that capturing these features of similarity could improve our measure of cross-language similarity.

Judges were also able to provide 'other' comments and several highlighted a number of non-English articles containing duplicate English sentences. It would appear that in these cases, the assessors ignored the content during comparison; however, the computed measure of similarity would incorrectly count these cases as similar and thereby inflate the similarity scores. A solution to this would be to include a maximum threshold, above which sentences are filtered out (similar to using a minimum threshold) or to use language detection to detect such cases and then to ignore them during the sentence alignment stage.

2.3.5.2 Measuring Cross-Language Similarity

A further goal of this study was to compare an adapted version of an existing method for cross-language similarity (Adafre and de Rijke 2006) with an approach based on using freely available MT systems. In contrast to prior work, we compare the computed similarity scores with human judgements to identify their degree of correlation. We also compare results obtained across a range of language pairs to determine the success of exploiting inter-language links in Wikipedia to develop a bilingual lexicon. A similarity score based on this approach seems to capture some essence of cross-language document similarity as judged manually. There are obvious weaknesses to our approach for some language pairs (e.g. ET–EN) that require further development. However, the issue is not resolved by using an MT system which may simply reflect the difficulty faced when dealing with under-resourced languages that results in lower translation quality.

Through manual inspection, we identified two cases where assessors disagree with the assigned anchor+word overlap scores: (1) assessors assigning a low similarity score to pairs which have high anchor+word overlap score; (2) assessors assigning a high similarity score to pairs which have low anchor+word overlap score. We found that the most common reason for the first case is that the shorter document (normally the non-English one) is a subset of the longer document. In these cases, documents are scored highly using the anchor+word overlap approach as the length of the smaller document is used to normalise the similarity score. Assessors, on the other hand, identified different information in the longer document not included in the shorter document. They, therefore, gave a lower similarity score.

In the second case, when similar documents are scored poorly using the anchor+word overlap approach, we find that one reason is due to the existence (or lack of) overlapping cognates. This results in better performance on languages with a similar written form to English, such as German. For other languages, such as Greek, the alphabets are very different and subsequently, the number of matching cognates drops significantly. This then causes the approach to rely on the availability of links which in some cases is not enough. There are similar documents that simply

do not contain enough links for the language-independent method to identify parallel sentences accurately.

The findings also suggest that a lack of correlation in results arises, because the similarity of document pairs is assessed in different ways. The anchor+word overlap approach was initially developed to identify similar documents in Wikipedia for the purpose of building comparable corpora for MT. Therefore, the method is intended to identify Wikipedia documents that contain similar fragments. In situations described previously for case 1, documents are considered useful, because they contain overlapping fragments and, therefore, are scored higher. Whether or not the longer document contains a large amount of new information is irrelevant for the anchor+word overlap method. However, this does not relate very well to assessors' judgement, as they base their scores on the overall content of the Wikipedia articles. Therefore, further work is needed to better capture human similarity judgements.

2.3.6 Section Conclusions

In this section, we have analysed the performance of two similarity measures in identifying cross-language similarity between Wikipedia articles on the same topic but written in different languages. In this initial study, we evaluated 800 document pairs and found that similarity measures using machine translation and language-independent features based on mining anchor texts from inter-language links in Wikipedia correlate with each other (rho $= 0.744$, $p < 0.01$) and to a lesser degree with human judgements (rho $= 0.353$, $p < 0.01$ and rho $= 0.325$, $p < 0.01$). We have shown that our measure of similarity varies widely across language pairs with, for example, German–English results correlating better with human judgements than Estonian–English.

The performance of the language-independent method is comparable to an approach based on translating articles into English and determining similarity monolingually in several language pairs. This demonstrates the potential benefit of mining inter-language links from Wikipedia for under-resourced languages. We have also shown that the similarity measures can be used to perform classification with an accuracy of 40% for 5 levels or classes of similarity. We also analysed features that judges have identified to capture cross-language similarity between articles and have used this analysis to uncover distinguishing features of the various levels of similarity.

There are several avenues to explore in future work. These include improving the cross-language similarity measures by incorporating term weighting rather than using binary feature vectors, automatically capturing the features identified by the judges to distinguish similar and non-similar document pairs and improving the sentence alignment algorithm to include cases where sentences are split up between the source and target documents.

2.4 Metrics for Identifying Comparability Levels in Non-aligned Documents

2.4.1 Using Parallel and Comparable Corpora for MT

Parallel corpora have been extensively exploited in different ways in machine translation (MT)—both in Statistical (SMT) and more recently, in Rule-Based (RBMT) architectures: in SMT, aligned parallel resources are used for building translation phrase tables and calculating translation probabilities, and in RBMT—for automatically building bilingual dictionaries of translation equivalents and deriving bilingual mappings for frequent structural patterns. However, large parallel resources are not always available for this task, especially for under-resourced languages or narrow domains. For this reason, in recent years, the use of cross-lingual comparable corpora has attracted considerable attention in the MT community (Sharoff et al. 2006; Fung and Cheung 2004a; Munteanu and Marcu 2005; Babych et al. 2008).

Most of the applications of comparable corpora focus on discovering translation equivalents to support machine translation, such as bilingual lexicon extraction (Rapp 1995, 1999; Morin et al. 2007; Yu and Tsujii 2009; Li and Gaussier 2010; Prochasson and Fung 2011; Chiao and Zweigenbaum 2002), parallel phrase extraction (PPE) (Munteanu and Marcu 2006) and parallel sentence extraction (Fung and Cheung 2004b; Munteanu and Marcu 2005; Munteanu et al. 2004; Smith et al. 2010).

Comparability between documents is often understood to be when two texts belong to the same subject domain, genre or text type and so, its definition relies on these vague linguistic concepts. The problem with this definition is that it cannot be exactly benchmarked, since it is hard to relate automated measures of comparability to such inexact and non-measurable linguistic concepts. Research on comparable corpora needs not only good measures for comparability but also a clearer, technologically grounded and quantifiable definition of comparability in the first place.

In this section, we relate comparability to usefulness of comparable texts for MT. In particular, we propose a performance-based definition of comparability, as the possibility to extract parallel or quasi-parallel translation equivalents—words, phrases and sentences that are translations of each other. This definition relates comparability to texts' potential to improve the quality of MT by adding extracted phrases to phrase tables, training corpora or dictionaries. It also can be quantified as the rate of successful extraction of translation equivalents by automated tools, as proposed in Munteanu and Marcu (2006).

Still, successful detection of translation equivalents in comparable corpora very much depends on the number of potential translation equivalents that can be found there. This property can be characterised by (1) the degree of textual equivalence (i.e. comparability) of document pairs within a comparable corpus and (2) robustness of algorithm for alignment of textual units. The goal of this chapter is to address the first problem—to provide comparability metrics that can reliably identify cross-

lingual comparable documents from raw corpora crawled from the web and characterise the degree of their similarity. Information about the degree of comparability between different documents across languages enriches non-aligned comparable corpora by indicating potential document alignments and allowing developers to filter out documents pairs that are not useful. This, in turn, leads to extraction of good-quality translation equivalents from the corpora—the second problem, which is addressed in Chap. 5.

In this section, we present three different approaches to measure the comparability of cross-lingual comparable documents: one approach based on lexical mapping, one on keywords and one on machine translation mapping. Experimental results show that all of them can effectively predict the comparability levels of the compared document pairs. We then further investigate the applicability of the proposed metrics by measuring their impact on the task of parallel phrase extraction from comparable corpora. It turns out that a higher comparability level predicted by the metrics consistently leads to a higher number of parallel phrases being extracted from comparable documents. Thus, the metrics can help to select more comparable document pairs to enhance efficiency of parallel phrase extraction.

The remainder of this section is organised as follows: Section 2.4.2 discusses previous work. Section 2.4.3 introduces our comparability metrics. Section 2.4.4 presents the experimental results and evaluation. Section 2.4.5 describes the application of the metrics. Section 2.4.6 discusses the pros and cons of the proposed metrics, followed by conclusions and future work in Sect. 2.4.7.

2.4.2 Related Work

The term ‘comparability’, which is the key concept in this work, applies at the level of corpora, documents and sub-document units. However, so far, there is no widely accepted definition of comparability. For example there is no agreement on the degree of similarity that documents in comparable corpora should have or on the criteria for measuring comparability. Also, most of the work on algorithms that extract translation equivalents from comparable corpora assumes that the corpora used are reliably comparable and focuses on the design of efficient extraction algorithms. Therefore, there has been very little literature discussing the characteristics of comparable corpora (Maia 2003). In this section, we introduce some representative work which deals with different understandings of comparability metrics.

Some studies (Sharoff 2007; Maia 2003; McEnery and Xiao 2007) analyse comparability by assessing corpus composition, using structural criteria (e.g. format and size) and linguistic criteria (e.g. topic, domain, and genre). Kilgarriff and Rose (1998) measure similarity and homogeneity between monolingual corpora. They generate word frequency lists from each corpus and then apply statistical measures to the most frequent N words (e.g. top 500) in the compared corpora.

Our work is closer to the paradigm that deals with comparability measures in cross-lingual comparable corpora. Saralegi et al. (2008) measure the degree of

comparability of corpora (English and Basque) according to the distribution of topics and publication dates of documents. They compute content similarity for all the document pairs between two corpora. These similarity scores are then given as parameters for the Earth Mover's Distance (EMD) measure which is employed to calculate the global compatibility of the corpora. Munteanu and Marcu (2005, 2006) select comparable document pairs in a cross-lingual information retrieval-based manner by using the Lemur toolkit.[11] The retrieved document pairs then serve as input for the tasks of parallel sentence and sub-sentence extraction. Smith et al. (2010) treat Wikipedia as a comparable corpus and use 'interwiki' links to identify aligned comparable document pairs for the task of parallel sentence extraction. Li and Gaussier (2010) propose a comparability metric which can be applied at both document level and corpus level and use it as a measure to select more comparable texts from other external sources to add into the original corpora for bilingual lexicon extraction. The metric measures the proportion of overlapping words translated from the source language corpus into the target language using a bilingual dictionary. They evaluate the metric on the richly resourced English–French language pair, where high-quality dictionary resources are available. However, for under-resourced languages, such dictionaries are often not available, or may be less reliable, or have small coverage, lack resources for lemmatisation, are not publically available, even if they exist in commercial applications, etc.

2.4.3 Comparability Metrics

To measure the degree of comparability of document pairs in different languages, we need to translate the texts or map lexical items from a source language into target languages, so that we can compare them within the same language. Usually, this is done by using bilingual dictionaries (Rapp 1999; Li and Gaussier 2010; Prochasson and Fung 2011) or available machine translation tools. We present three different approaches that are based on this mapping process and measure the comparability of comparable documents: a dictionary-based metric (or lexical mapping), a keyword-based metric and an MT-based metric.

2.4.3.1 Lexical Mapping Based Metric

The core, straightforward approach to map compared textual features across languages is to use a bilingual dictionary to produce a lexical mapping within a specific language pair. However, for under-resourced languages, the situation is complicated by reduced availability and quality of bilingual lexical resources. Unlike language pairs in which both languages are richly resourced (e.g. English–French, or

[11] Available at http://www.lemurproject.org/

English–Spanish) and dictionary resources are relatively easy to obtain, it is likely that bilingual dictionaries with good word coverage either do not exist or are not publicly available (e.g. for English–Slovenian, or English–Lithuanian language pairs).

In order to address this problem, we automatically construct statistical alignment dictionaries by using word alignment on large-scale parallel corpora (e.g. Europarl and JRC-Acquis[12]).

Specifically, the GIZA++ toolkit (Och and Ney 2000) with default settings is used for word alignment on the JRC-Acquis parallel corpora (Steinberger et al. 2006). The aligned word pairs together with the alignment probabilities are then converted into dictionary entries. For example, in the Estonian–English language pair, the alignment example '*kompanii ~ company 0.625*' in the word alignment table means the Estonian word '*kompanii*' can be translated as (or aligned with) the English candidate word '*company*' with a probability of 0.625. We prepared a statistical alignment dictionary in a specific format, where translation candidates are ranked by translation probability in descending order. Note that without corpus lemmatisation in a preparatory stage, the statistical alignment dictionary contains inflected forms of words, not only lemmas.

Using the resulting dictionary, we then perform lexical mapping (word-for-word). We scan each word in the source language text to check if it occurs among the dictionary entries. Europarl and JRC corpora often do not contain domain-specific terminology or low-frequency words, but they mostly cover frequent words from the general lexicon. If the word is found in the dictionary, the first and second (if available) translation candidates are recorded as the corresponding word in the mapping. The second candidate is kept in the mapping only if its translation probability is higher than 0.3.[13]

For non-English+English language pairs, the non-English texts are normally mapped into English. If both languages are non-English (e.g. Greek-Romanian), we use English as a pivot language and map both the source and target language texts into English.[14]This allows us to exploit richer open access monolingual annotation resources that are available for English. Monolingual processing tasks that benefit from these resources include stop-word filtering and word lemmatisation.[15] Finally, cosine similarity measure is applied to compute the comparability strength of the compared document pairs.

[12]The JRC-Acquis covers 22 European languages and provides large-scale parallel corpora for all the 231 language pairs.

[13]From manual inspection on the word alignment results, we find that if the alignment probability is higher than 0.3, it is more reliable.

[14]Generally, in JRC-Acquis, the size of parallel corpora for most non-English language pairs is much smaller than that of language pairs that contain English. Therefore, the resulting bilingual dictionaries that contain English have better word coverage, as they have many more dictionary entries.

[15]We use WordNet (Fellbaum 1998) for word lemmatization.

2.4.3.2 Keyword-Based Metric

The lexical mapping-based metric takes all the words in the text into account for measuring comparability. However, an alternative approach is to retain only a small number of representative words (keywords) and discard all the other less informative words in each document, using only the retained words to measure document comparability. Our intuition is that, if two documents share more keywords, they should be more comparable.

To compute a keyword-based comparability metric, we perform keyword extraction by using a simple TF.IDF-based approach which has been shown to be effective for keyword or keyphrase extraction (Frank et al. 1999; Hulth 2003; Liu et al. 2009).

Our keyword-based metric can be described as follows. Firstly, similar to the lexical mapping-based metric, bilingual dictionaries are used to map non-English texts into English. Thus, only the English resources are applied for stop-word filtering and word lemmatisation. These steps are useful for keyword extraction, since data sparseness for keywords is higher than for general usage words. We then use TF.IDF to measure the weight of words in the document and rank the words by their TF.IDF weights in descending order. The top N (e.g. top 30) words are extracted as keywords to represent the document. Finally, the comparability of each document pair is determined by applying cosine similarity to their keyword lists.

2.4.3.3 Machine Translation (MT)–Based Metrics

Since lexical and keyword-based metrics use bilingual dictionaries, the major shortcoming for both approaches is that words that do not occur in the dictionary are omitted. Also, the mapping result is a list of isolated words and information such as word order, syntactic structure and named entities cannot be preserved. Therefore, in order to improve the text translation quality and preserve a richer set of textual features during cross-lingual mapping, we use state-of-the-art SMT systems.

For our experiment, we use the Microsoft translation API[16] to translate texts in under-resourced languages (e.g. Lithuanian and Slovenian) into English and then explore several features for comparability metric design which are listed as below.

- **Lexical features**: Lemmatised bag-of-word representation of each document after stop-word filtering. Lexical similarity (denoted by W_L) of each document pair is then obtained by applying the cosine measure to the lexical features.
- **Structure features**: We approximate the morphosyntactic structure of the document as the number of content words (adjectives, adverbs, nouns, verbs and proper nouns) and the number of sentences in each document, denoted by C_D and S_D, respectively. The intuition is that, if two documents are highly comparable, their number of content words and their document length should be similar.

[16] Available at http://code.google.com/p/microsoft-translator-java-api/

The structure similarity (denoted by W_S of two documents $D1$ and $D2$) is defined as below:

$$W_S = 0.5^*(C_{D1}/C_{D2}) + 0.5^*(S_{D1}/S_{D2}), \text{supposing that}$$
$$C_{D1} \leq C_{D2} C\ S_{D1} \leq S_{D2}.$$

- **Keyword features**: Top 20 words (ranked by TF.IDF weight) of each document. The keyword similarity (denoted by W_K) of two documents is also measured by the cosine of the angle between the corresponding feature vectors.
- **NE features**: Named entities in each document. If more named entities are shared in two documents, the documents are very likely to talk about the same event or subject and thus should be more comparable. We use the Stanford named entity recogniser[17] to extract named entities from the texts (Finkel et al. 2005). Again, cosine similarity is then applied to calculate the named entity feature (denoted by W_N) in a document pair.

We then combine these four different types of score in an ensemble manner. Specifically, a weighted average strategy is applied: each individual score is associated with a constant weight, indicating the relative confidence (importance) of the corresponding type of score. The overall comparability score (denoted by SC) of a document pair is thus computed as below:

$$\text{SC} = \alpha * W_L + \beta * W_S + \gamma * W_K + \delta * W_N,$$

where α, β, γ and $\delta \in [0, 1]$ and $\alpha + \beta + \gamma + \delta = 1$.SC should be a value between 0 and 1, and larger SC values indicate higher comparability.

2.4.4 Experiments and Evaluation

2.4.4.1 Data Sources

To investigate the reliability of the proposed comparability metrics, we performed experiments for six language pairs which contain under-resourced languages: German–English (DE–EN), Estonian–English (ET–EN), Lithuanian–English (LT–EN), Latvian–English (LV–EN), Slovenian–English (SL–EN) and Greek–Romanian (EL–RO). A comparable corpus was collected for each language pair. Based on the definition of comparability levels (see Sect. 2.1), human annotators fluent in both languages then manually annotated the comparability degree (parallel, strongly comparable, and weakly comparable) at the document level. Hence, these bilingual comparable corpora were used as gold standards for our experiments. The data distribution for each language pair, that is, number of document pairs in each comparability level, is given in Table 2.10.

[17] Available at http://nlp.stanford.edu/software/CRF-NER.shtml

Table 2.10 Data distribution of gold standard corpora

Language pair	# Document pairs	Parallel	Strongly comparable	Weakly comparable
DE–EN	1286	531	715	40
ET–EN	1648	182	987	479
LT–EN	1177	347	509	321
LV–EN	1252	184	558	510
SL–EN	1795	532	302	961
EL–RO	485	38	365	82

2.4.4.2 Experimental Results

For evaluation of the methodology, we used the following procedure. For each language pair, we compute average scores for all the document pairs at the same comparability level (annotated by human judges) and compare them to the gold-standard comparability labels. In addition, in order to better reveal the relation between the scores obtained from the proposed metrics and comparability levels, we also measure the Pearson correlation between them.[18] For the keyword-based metric, the top 30 keywords are extracted from each text for experiment. For the machine translation-based metric, we empirically set

$$\alpha = 0.5; \beta = \gamma = 0.2 \text{ and } \delta = 0.1.$$

This is based on the assumption that lexical features can best characterise the comparability given the good translation quality provided by the powerful MT system, while keyword and named entity features are also better indicators of comparability than simple document length information.

The results for the lexical mapping-based metric, the keyword-based metric and the machine translation-based metric are listed in Tables 2.11, 2.12 and 2.13, respectively.

Overall, from the average scores for each comparability level presented in Tables 2.11, 2.12 and 2.13, we can see that the scores obtained from the three comparability metrics can reliably reflect the comparability levels across different language pairs, as the average scores for higher comparable levels are always significantly larger than those of lower comparable levels, namely:

$$\text{SC(parallel)} > \text{SC(strongly comparable)} > \text{SC(weakly comparable)}$$

In addition, for all three metrics, the Pearson correlation scores are very high (over 0.93) across different language pairs, which indicates that there is a strong

[18]For the correlation measure, we use numerical calibration to different comparability degrees: 'Parallel', 'strongly-comparable' and 'weakly-comparable' are converted to 3, 2 and 1, respectively. The correlation is then computed between the numerical comparability levels and the corresponding average comparability scores automatically derived from the metrics.

Table 2.11 Average comparability scores for the lexical mapping-based metric

Language pair	Parallel	Strongly comparable	Weakly comparable	*r* Correlation
DE–EN	0.545	0.476	0.182	0.941
ET–EN	0.553	0.381	0.228	0.999
LT–EN	0.545	0.461	0.225	0.964
LV–EN	0.625	0.494	0.179	0.973
SL–EN	0.535	0.456	0.314	0.987
EL–RO	0.342	0.131	0.090	0.932

Table 2.12 Average comparability scores for the keyword-based metric

Language pair	Parallel	Strongly comparable	Weakly comparable	*r* Correlation
DE–EN	0.526	0.486	0.084	0.941
ET–EN	0.502	0.345	0.184	0.990
LT–EN	0.485	0.420	0.202	0.954
LV–EN	0.590	0.448	0.124	0.975
SL–EN	0.551	0.505	0.292	0.937
EL–RO	0.210	0.110	0.031	0.997

Table 2.13 Average comparability scores for the MT-based metric

Language pair	Parallel	Strongly comparable	Weakly comparable	*r* Correlation
DE–EN	0.912	0.622	0.326	0.999
ET–EN	0.765	0.547	0.310	0.999
LT–EN	0.755	0.613	0.308	0.984
LV–EN	0.770	0.627	0.236	0.966
SL–EN	0.779	0.582	0.373	0.988
EL–RO	0.863	0.446	0.214	0.988

correlation between the comparability scores obtained from the metrics and the corresponding comparability level.

Moreover, from comparison of Tables 2.11, 2.12 and 2.13, we have several other findings. Firstly, the performance of the keyword-based metric (seeTable 2.12) is comparable to the lexical mapping-based metric (Table 2.11) as their comparability scores for the corresponding comparability levels are similar. This means it is reasonable to determine the comparability level by only comparing a small number of keywords from the texts. Secondly, the scores obtained for the MT-based metric (Table 2.13) are significantly higher than those in both the lexical mapping-based metric and the keyword-based metric. This result may be explained by the advantages of using a state-of-the-art MT system. In comparison to the approach of using a dictionary for word-for-word mapping, such systems can provide much better text translation, which allows a greater proportion of lexical overlapping to be detected and more useful features to be mined in the translated texts. Thirdly, in the lexical mapping-based metric and keyword-based metric, we can also see that, although the average scores for EL–RO (both under-resourced languages) conform to the

comparability levels, they are much lower than those of the other five language pairs. The reason is that the sizes of the parallel corpora in JRC-Acquis for these five language pairs are significantly larger (over 1 million parallel sentences) than for EL–EN, RO–EN,[19] and EL–RO. Thus the resulting dictionaries for these five language pairs also contain many more dictionary entries.

2.4.5 Metric Application to Equivalent Extraction

The experiments in Sect. 2.4.4 confirm the reliability of the proposed metrics. The comparability metrics are thus useful for collecting high-quality comparable corpora, as they can help filter out weakly comparable or non-comparable document pairs from the raw crawled corpora. But are they also useful for other Natural Language Processing (NLP) tasks, such as translation equivalent detection from comparable corpora? In this section, we further measure the impact of the metrics on parallel phrase extraction (PPE) from comparable corpora. Our intuition is that, if document pairs are assigned higher comparability scores by the metrics, they should be more comparable and, thus, more parallel phrases can be extracted from them.

The algorithm for parallel phrase extraction, which further develops the approach presented in Munteanu and Marcu (2006), uses lexical overlap and structural matching measures (Ion 2012). Taking a list of bilingual comparable document pairs as input, the extraction algorithm involves the following steps:

1. Split the source and target language documents into phrases.
2. Compute the degree of parallelism for each candidate pair of phrases by using the bilingual dictionary generated from GIZA++ (base dictionary), and retain all the phrase pairs with a score larger than a predefined parallelism threshold.
3. Apply GIZA++ to the retained phrase pairs to detect new dictionary entries and add them to the base dictionary.
4. Repeat Steps 2 and 3 several times (empirically set at 5) by using the augmented dictionary, and output the detected phrase pairs.

Phrases extracted by this algorithm are frequently not exact translation equivalents. Below we give some English–German examples of extracted equivalents with their corresponding alignment scores:

1. *But a successful mission—seiner ueberaus erfolgreichen Mission abgebremst—0.815501989333333*
2. *Former President Jimmy Carter—Der ehemalige US-Praesident Jimmy Carter—0.69708324976825*
3. *on the Korean Peninsula—auf der koreanischen Halbinsel—0.8677432145*

[19]Remember that in our experiment, English is used as the pivot language for non-English language pairs.

4. *across the Muslim world—mit der muslim- ischen Welt ermoeglichen—0.893330864*
5. *to join the United Nations—der Weg in die Vereinten Nationen offensteht—*0.397418711927629

Even though some of the extracted phrases are not exact translation equivalents, they may still be useful resources both for SMT and RBMT if these phrases are passed through an extra pre-processing stage, or if the engines are modified specifically to work with semi-parallel translation equivalents extracted from comparable texts. We address this issue in the discussion section (see Sect. 2.4.6).

For evaluation, we measure how the metrics affect the performance of the parallel phrase extraction algorithm on five language pairs (DE–EN, ET–EN, LT–EN, LV–EN and SL–EN). A large raw comparable corpus for each language pair was crawled from the web, and the metrics were then applied to assign comparability scores to all the document pairs in each corpus. For each language pair, we set three different intervals based on the comparability score (SC) and randomly select 500 document pairs in each interval for evaluation. For the MT-based metric, the three intervals are (1) $0.1 < SC < 0.3$, (2) $0.3 < SC < 0.5$ and (3) $SC > 0.5$. For the lexical mapping-based metric and keyword-based metric, since their scores are lower than those of the MT-based metric for each comparability level, we set three lower intervals at (1) $0.1 < SC < 0.2$, (2) $0.2 < SC < 0.4$ and (3) $SC > 0.4$. The experiment focuses on counting the number of extracted parallel phrases with parallelism score $SC \geq 0.4$[20] and computes the average number of extracted phrases per 1,00,000 words (the sum of words in the source and target language documents) for each interval. In addition, the Pearson correlation measure is also applied to measure the correlation between the interval[21] of comparability scores and the number of extracted parallel phrases. The results, which summarise the impact of the three metrics on the performance of parallel phrase extraction, are listed in Tables 2.14, 2.15 and 2.16, respectively.

From Tables 2.14, 2.15 and 2.16, we can clearly see that for all the five language pairs, based on the average number of extracted aligned phrases, we have interval (3) > (2) > (1). In other words, for any of the three metrics, a higher comparability level always leads to significantly more aligned phrases extracted from the comparable documents. Moreover, although the lexical mapping-based metric and the keyword-based metric produce lower comparability scores than the MT-based metric (see Sect. 2.4.4), they have similar impact on the task of parallel phrase extraction. This means, the comparability score itself does not matter too much, as long as the metrics are reliable and proper thresholds are set for different metrics.

For all three metrics, the Pearson correlation scores are very close to 1 for all the language pairs which indicates that the intervals of comparability scores obtained from the metrics are in line with the performance of the equivalent extraction

[20] A manual evaluation of a small set of extracted data shows that parallel phrases with parallelism score $SC \geq 0.4$ are more reliable.

[21] For the purpose of correlation measure, the three intervals are numerically calibrated as '1', '2' and '3', respectively.

Table 2.14 Impact of the lexical mapping-based metric to parallel phrase extraction (PPE)

Language pair	$0.1 < SC < 0.2$	$0.2 < SC < 0.4$	$SC > 0.4$	*r* Correlation
DE–EN	728	1434	2510	0.993
ET–EN	313	631	1166	0.989
LT–EN	258	419	894	0.962
LV–EN	470	859	1900	0.967
SL–EN	393	946	2220	0.975

Table 2.15 Impact of the keyword based metric to parallel phrase extraction (PPE)

Language pair	$0.1 < SC < 0.2$	$0.2 < SC < 0.4$	$SC > 0.4$	*r* Correlation
DE–EN	1007	1340	2151	0.972
ET–EN	438	650	1050	0.984
LT–EN	306	442	765	0.973
LV–EN	600	966	1722	0.980
SL–EN	715	1026	1854	0.967

Table 2.16 Impact of the machine translation (MT)-based metric to parallel phrase extraction (PPE)

Language pair	$0.1 < SC < 0.3$	$0.3 < SC < 0.5$	$SC > 0.5$	*r* Correlation
DE–EN	861	1547	2552	0.996
ET–EN	448	883	1251	0.999
LT–EN	293	483	1070	0.959
LV–EN	589	1072	2037	0.982
SL–EN	560	1151	2421	0.979

algorithm. Therefore, in order to extract more parallel phrases (or other translation equivalents) from comparable corpora, we can try to improve the corpus comparability by applying the comparability metrics beforehand to add highly comparable document pairs to the corpora.

2.4.6 Discussion

We have presented three different approaches to measure comparability at the document level. In this section, we will analyse the advantages and limitations of the proposed metrics, and the feasibility of using semi-parallel equivalents in MT.

2.4.6.1 Advantages and Disadvantages of the Metrics

Using a bilingual dictionary for lexical mapping is simple and fast. However, as it adopts a word-for-word mapping strategy and out-of-vocabulary (OOV) words are

omitted, the linguistic structure of the original texts is badly hurt after mapping. Thus, apart from lexical information, it is difficult to explore other useful features for comparability metrics. The TF.IDF-based keyword extraction approach allows us to select more representative words and prune a large number of less informative words from the texts. The keywords are usually relevant to subject and domain terms which is quite useful in judging the comparability of two documents. Both the lexical mapping-based approach and the keyword-based approach use a dictionary for lexical translation, thus relying on the availability and completeness of dictionary resources or large-scale parallel corpora.

The machine translation-based metric provides much better text translation than the dictionary-based approach, with the consequence that the comparability of two documents can be better revealed from the richer lexical information and other useful features, such as named entities. However, the text translation process is expensive, as it depends on the availability of the powerful MT systems[22] and takes substantially longer than the simple dictionary-based translation.

In addition, we use a translation strategy of translating texts from under-resourced (or less-resourced) languages into rich-resourced language. When both languages are under-resourced languages, English is used as the pivot language for translation. This can compensate for the shortage of the linguistic resources in the under-resourced languages and allows us to take advantage of various resources in more richly resourced languages.

2.4.6.2 Using Semi-parallel Equivalents in MT Systems

We note that modern SMT and RBMT systems take maximal advantage of strictly parallel phrases, but they still do not exploit the full potential of semi-parallel translation equivalents of the type illustrated in the application chapter (see Chap. 6). Such resources, even though they are not exact equivalents, contain useful information which is not used by the systems.

In particular, the modern decoders do not work with under-specified phrases in phrase tables, and do not work with factored semantic features. For example, consider the phrase:

But a successful mission—seiner ueberaus erfolgreichen Mission abgebremst

The English side contains the word *but*, which pre-supposes contrast, and the German side contains the words *ueberaus erfolgreichen* (‘*generally successful’)* and *abgebremst (‘slowed down’)*—which taken together exemplify a contrast, since they have different semantic prosodies. In this example, the semantic feature of contrast can be extracted and re-used in other contexts. However, this would require the

[22]Alternatively, we can also train MT systems for text translation by using the available SMT toolkits (e.g. Moses) on large-scale parallel corpora.

development of a new generation of decoders or rule-based systems that can successfully identify and re-use such subtle semantic features.

2.4.7 Conclusion

In this section, we investigated the impact of the metrics on the task of parallel phrase extraction from comparable corpora. It turns out that higher comparability scores always lead to significantly more parallel phrases extracted from comparable documents. Since higher quality comparable corpora are more fruitful, our metrics can be applied to select highly comparable document pairs for the tasks of translation equivalent extraction.

References

Adafre, S. F., & de Rijke, M. (2006). Finding similar sentences across multiple languages in Wikipedia. *Proceedings of the EACL Workshop on New Text*, Trento, Italy.

Babych, B., Hartley, A., Sharoff, S., & Mudraya, O. (2007). Assisting translators in indirect lexical transfer. *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (pp. 136–143).

Babych, B., Sharoff S., & Hartley, A. (2008). Generalising lexical translation strategies for MT using comparable corpora. *Proceedings of LREC 2008*, Marrakech, Morocco.

Bharadwaj, R. G., & Varma, V. (2011). Language independent identification of parallel sentences using Wikipedia. *Proceedings of the 20th International Conference Companion on World Wide Web, WWW '11* (pp. 11–12).

Chiao, Y.-Ch., & Zweigenbaum, P. (2002). Looking for candidate translational equivalents in specialized, comparable corpora. *Proceedings of COLING 2002*, Taipei, Taiwan.

Daille, B., & Morin, E. (2005). French-English terminology extraction from comparable corpora. *IJCNLP* (pp. 707–718).

Eisele, A., & Xu, J. (2010). improving machine translation performance using comparable corpora. *Proceedings of the LREC Workshop on Building and Using Comparable Corpora*, Malta, May 2010.

Eisele, A., Federmann, C., Saint-Amand, H., Jellinghaus, M., Herrmann, T., & Chen, Y. (2008). Using Moses to integrate multiple rule-based machine translation engines into a hybrid system. *Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 179–182).

Erdmann, M., Nakayama, K., Hara, T., & Nishio, S. (2008). Extraction of bilingual terminology from a multilingual web-based encyclopedia. *Journal of Information Processing, 16*, 67–79.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Filatova, E. (2009). Directions for exploiting asymmetries in multilingual Wikipedia. *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3 '09)*.

Finkel, J., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. *Proceedings of ACL 2005*, University of Michigan, Ann Arbor, MI.

Frank, E., Paynter, G, & Witten, I. (1999). Domain-specific keyphrase extraction. *Proceedings of IJCAI 1999*, Stockholm, Sweden.

Fung, P. (1998). A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. *Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA'98)* (pp. 1–16). Springer.

Fung, P., & Cheung, P. (2004a). Mining very non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. *Proceedings of EMNLP 2004*, Barcelona, Spain.

Fung, P., & Cheung, P. (2004b). Multi-level bootstrapping for extracting parallel sentences from a quasicomparable corpus. *Proceedings of COL- ING 2004*, Geneva, Switzerland.

Fung, P., & Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. *COLING '98: Proceedings of the 17th International Conference on Computational Linguistics* (pp. 414–420).

Gamallo, P. O., & López, I. G. (2010). Wikipedia as multilingual source of comparable corpora. *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora, LREC* (pp. 21–25). http://www.fb06.unimainz.de/lk/bucc2010/documents/Proceedings-BUCC-2010.pdf#page=29

Hatzivassiloglou, V., Klavans, J. L., & Eskin, E. (1999). Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 203–212).

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of EMNLP 2003*, Sapporo, Japan.

Ion, R. (2012). PEXACC: A parallel data mining algorithm from comparable corpora. *Proceedings of LREC 2012*, Istanbul, Turkey.

Kanaris, I., & Stamatatos, E. (2009). Learning to recognize webpage genres. *Information Processing and Management, 45*, 499–512.

Kessler, B., Numberg, G., & Schuetze, H. (1998). Automatic detection of text genre. *ACL '98: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics* (pp. 32–38).

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics, 6*(1), 1–37 (Reprinted in Teubert & Krishnamurthy (Eds.), *Corpus linguistics: Critical concepts in linguistics*. Routledge. 2007.) Retrieved from http://www.kilgarriff.co.uk/Publications/2001-K-CompCorpIJCL.pdf.

Kilgarriff, A., & Rose, T. (1998). Measures for corpus similarity and homogeneity. *Proceedings of EMNLP 1998*, Granada, Spain.

Lee, M. D., Pincombe, B., & Welsh, M. (2005). An empirical evaluation of models of text document similarity. *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 1254–1259).

Li, B., & Gaussier, E. (2010). Improving corpus comparability for bilingual lexicon extraction from comparable corpora. *Proceedings of COLING 2010*, Beijing, China.

Li, Y., McLean, D., Bandar, Z., O'Shea, J., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering, 18*(8), 1138–1150.

Lin, W., Snover, M., & Ji, H. (2011). Unsupervised language-independent name translation mining from Wikipedia infoboxes. *Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing* (pp. 43–52). Edinburgh, Scotland (pp. 27–31).

Liu, F., Pennell, D., Liu, F., & Liu, Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. *Proceedings of NAACL 2009*, Boulder, Colorado.

Lu, Y., Huang, J., & Liu, Q. (2007). Improving statistical machine translation performance by training data selection and optimization. *Proceedings of the 2007 EMNLP-CoNLL* (pp. 343–350).

Maia, B. (2003). What are comparable corpora? *Proceedings of the Corpus Linguistics Workshop on Multilingual Corpora: Linguistic Requirements and Technical Perspectives*, Lancaster.

McEnery, A., & Xiao, Z. (2007). Parallel and comparable corpora? *Incorporating Corpora: Translation and the Linguist. Translating Europe. Multilingual Matters*, Clevedon.

Morin, E., Daille, B., Takeuchi, K., & Kageura, K. (2007). Bilingual terminology mining – using brain, not brawn comparable corpora. *Proceedings of ACL 2007* (pp. 664–671), Prague, Czech Republic.

Munteanu, D., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Munteanu, D. S., & Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. *ACL-2006: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 81–88), Sydney, Australia.

Munteanu, D. S., Fraser, A., Marcu, D. (2004). Improved machine translation performance via parallel sentence extraction from comparable corpora. In: *HLT-NAACL 2004: Main Proceedings* (pp. 265–272).

Och, F., & Ney, H. (2000). Improved statistical alignment models. *Proceedings of ACL 2000*, Hongkong, China.

Och, F., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics, 29*(1), 19–51.

Otero, P. G., & López, I. G. (2010). Wikipedia as multilingual source of comparable corpora. *Proceedings of the LREC Workshop on BUCC* (pp. 30–37).

Patry, A., & Langlais, P. (2011). Identifying parallel documents from a large bilingual collection of texts: Application to parallel article extraction in Wikipedia. *Proceedings of the 4th Workshop on Building and Using Comparable Corpora* (pp. 87–95).

Prochasson, E., & Fung, P. (2011). Rare word translation extraction from aligned comparable documents. *Proceedings of ACL-HLT 2011*, Portland, OR.

Rapp, R. (1995). Identifying word translations in non-parallel texts. *ACL '95: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics* (pp. 320–322), Cambridge, MA.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. *ACL '99: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 519–526). College Park, MA.

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. *WCC '00: Proceedings of the Workshop on Comparing Corpora* (pp. 1–6).

Saralegi, X., Vicente, I., & Gurrutxaga, A. (2008). Automatic extraction of bilingual terms from comparable corpora in a popular science domain. *Proceedings of the Workshop on Comparable Corpora, LREC 2008*, Marrakech, Morocco.

Sharoff, S. (2007). Classifying Web corpora into domain and genre using automatic feature identification. *Proceedings of 3rd Web as Corpus Workshop*, Louvain-la-Neuve, Belgium.

Sharoff, S., Babych, B., & Hartley, A. (2006). Using comparable corpora to solve problems difficult for human translators. *COLING/ACL 2006 Main Conference Poster Sessions* (pp. 739–746).

Skadiņa, I., Vasiļjevs, A., Skadiņš, R., Gaizauskas, R., Tufiş, D., & Gornostay, T. (2010). Analysis and evaluation of comparable corpora for under resourced areas of machine translation. *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora. Applications of Parallel and Comparable Corpora in Natural Language Engineering and the Humanities* (pp. 6–14), Valletta, Malta.

Smith, J., Quirk, C., & Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. *Proceedings of NAACL 2010*, Los Angeles, CA.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., & Tufiş, D. (2006). The JRC- Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of LREC 2006*, Genoa, Italy.

Teubert, W. (1996). Comparable or parallel corpora? *International Journal of Lexicography, 9*, 238–264.

Tomás, J., Bataller, J., Casacuberta, F., & Lloret, J. (2008). Mining Wikipedia as a parallel and comparable corpus. *Language Forum, 1*, 34.

Vidulin, V., Lustrek, M., & Gams, M. (2007). Using genres to improve search engines. *Proceedings of the International Workshop Towards Genre-Enable Search Engines: The Impact of Natural Language Processing* (pp. 45–51).

Wu, D., & Fung, P. (2005). Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. *Natural Language Processing IJCNLP 2005, 3651*, 257–268.

Wu, Z., Markert, K., & Sharoff, S. (2010). Fine-grained genre classification using structural learning algorithms. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 749–759).

Xu, J., Deng, Y., Gao, Y., & Ney, H. (2007). Domain dependent machine translation. *Proceedings of the Machine Translation Summit XI*, Copenhagen, Denmark.

Yu, K., & Tsujii, J. (2009). Extracting bilingual dictionary from comparable corpora with dependency heterogeneity. *Proceedings of HLT-NAACL 2009*, Boulder, CO.

Zesch, T., Műller, C., & Gurevych, I. (2008). Extracting lexical semantic knowledge from Wikipedia and Wikictionary. *Proceedings of the LREC 2008*, Marrakech, Morocco.

Chapter 3
Collecting Comparable Corpora

Monica Lestari Paramita, Ahmet Aker, Paul Clough, Robert Gaizauskas, Nikos Glaros, Nikos Mastropavlos, Olga Yannoutsou, Radu Ion, Dan Ştefănescu, Alexandru Ceauşu, Dan Tufiş, and Judita Preiss

Abstract The availability of parallel corpora is limited, especially for under-resourced languages and narrow domains. On the other hand, the number of comparable documents in these areas that are freely available on the Web is continuously increasing. Algorithmic approaches to identify these documents from the Web are needed for the purpose of automatically building comparable corpora for these under-resourced languages and domains. How do we identify these comparable documents? What approaches should be used in collecting these comparable documents from different Web sources? In this chapter, we firstly present a review of previous techniques that have been developed for collecting comparable documents from the Web. Then we describe in detail three new techniques to gather comparable documents from three different types of Web sources: Wikipedia, news articles, and narrow domains.

3.1 Introduction

Statistical machine translation (SMT) relies on the availability of rich parallel corpus resources. Often, however, parallel resources are not readily available for under-resourced languages or specific narrow domains. This leads to underperforming machine translation systems. To overcome the low availability of parallel resources, the machine translation community has recognized the potential of using comparable

Chapter editors: Robert Gaizauskas and Monica Lestari Paramita

M. L. Paramita · A. Aker · P. Clough · R. Gaizauskas (✉) · J. Preiss
University of Sheffield, Sheffield, UK
e-mail: R.Gaizauskas@sheffield.ac.uk

N. Glaros · N. Mastropavlos · O. Yannoutsou
Institute for Language and Speech Processing (ILSP), Athens, Greece

R. Ion · D. Ştefănescu · A. Ceauşu · D. Tufiş
Research Institute for Artificial Intelligence, Romanian Academy Center for Artificial Intelligence (RACAI), Bucharest, Romania

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_3

resources as training data (Rapp 1999; Munteanu and Marcu 2002, 2006; Sharoff et al. 2006; Kumano et al. 2007; Barzilay and McKeown 2001; Kauchak and Barzilay 2006; Callison-Burch et al. 2006; Nakov 2008; Zhao et al. 2008; Marton et al. 2009).

Various attempts at gathering comparable corpora from the Web have been made (Braschler 1998; Resnik 1999; Huang et al. 2010; Talvensaari et al. 2008). The process of obtaining such a corpus involves: (1) downloading for each language a separate set of documents, (2) aligning comparable documents between these two document sets by comparing document contents. Extraction of useful units, such as parallel sentences, for SMT is then performed on aligned document pairs from the comparable corpus. This process is described in Chap. 4.

ACCURAT has investigated efficient methods and developed tools for identifying and gathering large amounts of comparable textual data from the Web for under-resourced languages and narrow domains. Many different corpus collection techniques were explored, developed, and tested extensively. Approaches considered to be the most suitable have been implemented and are described in this chapter. In particular, Sect. 3.2 reviews all significant methods and tools for harvesting comparable documents from the Web that existed before the ACCURAT project. Some of them served as the technological ground, upon which ACCURAT comparable corpora collection techniques were built. In Sect. 3.3, we describe the methods and tools for building comparable corpora from the Web developed within ACCURAT. These tools are available for download from the project's website.

Using these tools, we have successfully built comparable corpora for the following language pairs: English–German, English–Greek, English–Croatian, English–Estonian, English–Latvian, English–Romanian, English–Lithuanian, and English–Slovenian from three different Web sources: Wikipedia, news sites, and narrow domains.

3.2 Previous Work in Collecting Comparable Corpora

The process of collecting comparable documents from the Web consists mainly of two phases. First, Web crawling techniques are employed to collect documents from the Web. These crawlers proceed based on a seed website or a list of keywords, which serve as a means to focus the crawler. Once a collection of monolingual texts has been retrieved and indexed, different methods are applied to identify and align comparable document pairs. This section reports previous work in the two areas: Web crawling is described in Sect. 3.2.1 and comparable document identification is described in Sect. 3.2.2.

3.2.1 Web Crawling

Many tools and approaches have been developed to build comparable corpora using retrieval techniques. For example, BootCat (Baroni and Bernardini 2004) retrieves documents using a list of seed words as queries. Outputs are then used to bootstrap

the process by inserting more seed words extracted from the retrieved results to improve the recall of the retrieval process. This approach assumes that the retrieved results are relevant and satisfy the requirements of the query. Other approaches, on the other hand, evaluate the relevance of results. If relevant, the results are used to enhance some underlying language model, or included in the collection to generate a query; otherwise, the results are not considered. This approach is referred to as *focused crawling* (Chakrabarti et al. 2002) and has been shown to retrieve narrow domains more effectively than general-purpose crawlers.

Talvensaari et al. (2008) implemented focused crawling using keywords as the input seeds. In an approach that differed from BootCat, they did not specifically look for relevant documents; rather, they used the retrieval results to look for websites that consistently produced top results over the majority of these queries. These websites were seen as good resources for that particular domain and were crawled to retrieve all documents within them. Language was detected using a simple *n*-gram-based algorithm (Cavnar and Trenkle 1994).

Instead of using a list of keywords as query seeds, Ghani et al. (2005) used a set of documents previously judged as relevant and nonrelevant to a given query(s) to focus the crawl, in an approach referred to as CorpusBuilder. To focus retrieval on documents of under-resourced languages, in this case Slovenian, they used Slovenian documents as the relevant documents and documents from other languages as the nonrelevant documents. They investigated the performance of several query generation methods; an approach based on odds-ratio resulted in the highest performance, compared to term frequency or random sampling baselines. The odds-ratio of each word was calculated using the probability of the word occurring in a relevant vs. nonrelevant document. A further difference of this method compared to others is that the query uses both inclusion and exclusion of terms. Highest performance was obtained by using three positive and three negative keywords, each chosen based on the highest odds-ratio score of the sets of relevant and nonrelevant documents. After each retrieval operation, the first document was passed to a language filter. If this document was identified to be in Slovenian, the set of documents was updated, and query generation was performed again. In case the new document did not change the query, the next ranking document in the result was taken as a result, and this process was performed iteratively. This method managed to retrieve general corpora from minority languages effectively.

Dimalen and Roxas (2007) adapted the methods used in CorpusBuilder to collect documents from similar languages instead of just one minority language. In this case, they retrieved comparable documents in Tagalog and other similar languages from the Philippines, by using as input a set of relevant and nonrelevant documents. They implemented a novel approach to prune all words occurring in both relevant and nonrelevant documents, as these words would not be helpful in identifying the language of the new documents. The odds-ratio was then calculated using this pruned collection. To classify the documents into the correct Philippine language, Dimalen and Roxas (ibid.) calculated the edit distance of character *n*-grams of the retrieved document to the previously collected corpora in each language.

A different approach was investigated by Fung and Cheung (2004), in which parallel sentences from bilingual corpora were used to retrieve new documents from nonparallel corpora. These documents were likely to have different topics and therefore would not be found by standard keyword searching on topic, named entities, or dates. However, as they contained similar sentences, they could still share similar terms that could be used as parallel resources.

3.2.2 Identifying Comparable Text

A range of approaches have been used to automatically identify comparable texts or text fragments in a large collection of unaligned multi-lingual texts. For example, Resnik (1999) and Zhang et al. (2006) analyzed the HTML structures and URL paths of Web documents in order to find parallel texts. Other approaches have relied less on structure and more on language resources, such as existing parallel corpora, bilingual dictionaries, or machine translation systems to aid the translation process prior to searching for comparable or parallel text. For example, given a document in the source language, dictionaries can be used in a straightforward manner to translate the words (and phrases) in the document; these translated terms can then be used to build a query to retrieve comparable documents in the target language. Ambiguity, however, can be an issue when using dictionaries if a word (or phrase) has multiple interpretations, and, therefore, translations.[1] Problems also occur when a word does not exist in the dictionary. Such words are referred to as out-of-vocabulary or OOV terms. To solve this problem, cognate matching can be used to identify the translation of a word (e.g., 'colour' in English and 'couleur' in French) if the languages share the same etymological roots and use the same writing system (Simard et al. 1993). Alternatively, if multi-lingual parallel corpora in the required languages exist, document alignments can be computed and the resulting aligned texts can be used to build a statistical machine translation system (Koehn 2009). However, this approach is computationally expensive and it can also be difficult to gather enough resources for machine translation due to the limited amount of parallel corpora available and accessible on the Web, particularly for under-resourced languages. Moreover, most parallel corpora only cover a specific domain, such as law; an MT system built using these domain-specific corpora may perform poorly when used to translate documents from a different domain (see, e.g., Talvensaari et al. 2008). In aligning news documents, metadata in the article, such as date or location, have been used in identifying comparable texts. Argaw and Asker (2005) aligned news articles in Amharic–English that were published on the same date and occurred in the same place. No lexical resources were used to translate the words. Argaw and Asker instead performed transliteration on the titles and calculated the edit

[1]For example, "letter of credit" in English may be translated in Dutch as "accreditief" or "kredietbrief" (based on using Eurowordnet).

distance between words in the titles. Pairs of documents that scored above a certain threshold were deemed comparable.

Other approaches to align documents have been based on identifying overlapping contents, such as named entities (Hassan et al. 2007) and document clustering (Steinberger et al. 2005). Uszkoreit et al. (2010) approached the problem of automatically identifying parallel documents from the Web as a form of cross-language near-duplicate detection using word *n*-grams. In this work, Uszkoreit et al. used a baseline machine translation system to translate all documents into a single language, in their case English. An index was created for the entire corpus to map each unique *n*-gram in the corpus to the documents in which it occurred. *N*-grams whose document frequency was very large were discarded as this gave a drastic reduction in runtime performance with only a small reduction in recall. This is because *n*-grams with high document frequency have relatively little discriminatory power. Document pairs containing matching *n*-grams whose original languages were different were taken as candidate parallel documents and compared more rigorously using their lower order *n*-grams. The method was shown to identify approximately 65% of all parallel documents with almost no documents incorrectly identified as parallel (65% recall with 97% precision). Uszkoreit et al. (ibid) also showed that this method has linear complexity and could be run in distributed way. The approach is therefore very scalable over extremely large collections of documents, such as the Web. The authors reported that by using 2000 state-of-the-art CPUs, this method could be run over 2.5 billion Web pages in less than 24 h. Crucially, however, this computation time does not take into account the initial time required to translate all documents into English, which is likely to be an extremely computationally intensive task and a possible obstacle for institutions or individuals with access to limited language resources.

While many previous studies aimed to identify comparability at the document level (i.e., in aligning comparable or parallel texts), other works focused on identifying parallel sentences from large collections of multilingual comparable corpora (MCC). Munteanu and Marcu (2005) aligned sentences from Arabic and English news corpora by first translating every word in the Arabic document to English using a bilingual dictionary and then building a query for each Arabic document to retrieve comparable documents in English. The 100 highest ranked English documents were retrieved for each Arabic document and those published outside of a specified time window based on the time of publication of the Arabic document were filtered out. Each sentence in the Arabic document and those in each of the English documents were then paired, and various features were evaluated in each sentence pair to identify which sentences were parallel.

Identification of parallel sentences has also been performed between Wikipedia documents in different languages by utilizing Wikipedia inter-language link information.[2] Adafre and de Rijke (2006) retrieved parallel sentences from Wikipedia inter-language-linked article pairs by identifying overlap of links (or anchor texts)

[2]Wikipedia inter-language links connect documents from different languages that describe the same topic.

between sentence pairs. This approach also makes use of inter-language link information from Wikipedia in order to create a bilingual lexicon, therefore avoiding the need to use any external linguistic resources, such as a dictionary or MT system. Smith et al. (2010) developed this idea by using additional features, such as sentence length and longest aligned/unaligned words, to develop a binary sentence classifier trained on parallel corpora.

Bharadwaj and Varma (2011) also developed a binary sentence classifier for English–Hindi, which does not require parallel corpora or other linguistic resources. First, they indexed contents of English–Hindi inter-language-linked articles by treating each sentence as a bag-of-words and creating separate indexes for each language. To identify whether a sentence pair was parallel or not, they performed retrieval for each sentence from the appropriate index, that is, English sentences were queried on the English index and Hindi sentences were queried on the Hindi index. The features vectors were built using the following features: the difference in sentence lengths, the number of retrieved articles in each language whose corresponding articles in the other language (based on the inter-language links) are also retrieved, the number of articles whose corresponding articles in the other language are not retrieved, and the total number of retrieved documents for each language. They report that the binary sentence classifier is able to identify parallel sentences with an accuracy of 78%.

3.3 ACCURAT Techniques to Collect Comparable Documents

In this section, we describe various techniques to collect comparable documents from the Web. Our approaches were specifically developed to retrieve documents from different areas in the Web. First, we describe techniques to collect comparable documents from Wikipedia in Sect. 3.3.1. Techniques to collect comparable news articles are described in Sect. 3.3.2. Lastly, a technique to collect documents from narrow domains is described in Sect. 3.3.3.

3.3.1 Comparable Corpora Collection from Wikipedia

Wikipedia has been mined for various linguistic purposes due to the diversity and richness of information available in a variety of languages (Tomás et al. 2001). In addition, the presence of inter-language links, which connect documents from different languages describing the same topic, makes Wikipedia a useful multilingual resource (e.g., as a source of comparable documents). Wikipedia as a comparable corpus has been studied and used by Yu and Tsujii (2009), who sketched a simple mining algorithm for multilingual comparable corpora (MCC) that exploits

the existence of inter-language links between articles. In Sect. 3.3.1.1, we describe methods developed within ACCURAT project for extracting pairs of comparable articles from Wikipedia. In Sect. 3.3.1.2, we describe a method to measure similarity of pairs of Wikipedia articles deemed comparable by methods such as those of Sect. 3.3.1.1.

3.3.1.1 Extracting Comparable Articles

In this work, our goal is to extract good-quality MCC in three languages: Romanian, English, and German (Ion et al. 2010). We employed two different methods for gathering MCC from Wikipedia:

1. The first one considers an input list of good-quality Romanian articles[3] from Romanian Wikipedia (http://ro.wikipedia.org/) and for each such article, it searches for the equivalent in English Wikipedia.
2. The second one uses Princeton WordNet and extracts all the capitalized nouns (single-word or multi-word expressions) from all the synsets. Then, it looks for Wikipedia page names formed with these nouns, extracts them, and their corresponding Wikipedia pages in Romanian and German (if these exist).

We describe these two methods in more detail below.

The first method for MCC compilation uses three different heuristics for identifying the English equivalent of a given Romanian article (they are tried in the order listed):

1. It searches for *an English page with exactly the same name as the Romanian page*. For instance, we have found the following exact-match English pages (starting from the Romanian equivalents): "Alicia Keys", "Hollaback Girl", etc.
2. It searches for *the English link from the Romanian page* that would lead to the same article in those languages. The Romanian version of the page may or may not be a complete translation from English; we noticed that the narrative order of the English page was rarely kept and the content of the Romanian page often reflected the translator's beliefs with regard to the content of the English page.
3. It automatically *transforms the Romanian page name into an English Wikipedia search query* by using a translation dictionary that has scores for each translation pair. Thus, for each content word in the Romanian page name, we generated the first k translations ($k = 2$ in our experiments) and with this query, we retrieved the first 10 documents from English Wikipedia. We manually chose the right English candidate, but an automatic pairing method based on document clustering is described below.

[3]Good-quality articles are those that senior Wikipedia moderators and the Romanian Wikipedia community think to be complete, well written, with good references, etc.

Using these heuristics, we managed to compile a very good Romanian–English comparable corpus that consists of 128 paired Romanian and English documents of approx. 502K words in English and 602K words in Romanian.

The second method for MCC compilation uses Princeton Wordnet for extracting a list of named entities. These named entities are then transformed into Wikipedia links by replacing the white spaces with an underscore and adding the string "http://en.wikipedia.org/wiki/" in front of them. Then, an application performs the following steps:

1. It goes to every link and downloads the Wikipedia page if it exists.
2. Every downloaded Wiki page is searched for links to corresponding Romanian and German Wiki pages; if such links exist, those pages are also downloaded.
3. All the HTML tags of every EN–RO or EN–DE pair of Wiki documents are stripped so that only the plain text remains (there is also the possibility of preserving some mark-up for important terms highlighted in Wikipedia articles). The categories of the documents are kept in a simple database.

Using the categories of the documents, one can select documents referring to specific subjects. However, due to the fact that we searched only for named entities, confusions may occur. For example, Wiki articles about Paris, Rome, or London might be considered to be about sports as they were categorized, among other categories, as "Host cities of the Summer Olympic Games." In reality, these articles contain very little information about such topics.

Pairing Documents Using Clustering Technique

Clustering is an unsupervised machine learning technique that groups together objects based on a measure of similarity between them. This technique is appropriate for pairing documents in a comparable corpus according to their topical similarity. Classical document similarity measures rely on the supposition that the documents have common elements (words). But similar documents in different languages have actually very few common elements (numbers, formulae, punctuation marks, etc.). In order to make documents in different languages similar, one approach is to replace document terms with their equivalent translation pairs. In this approach, each document term is replaced with its translation equivalent pairs from a translation equivalents list. The document vectors for both source and target language documents are collections of translation equivalents pairs.

There are two difficulties in this approach that have to be surpassed:

1. **Translation Equivalents Selection.** Not all the translation equivalent pairs have the same discriminative power in differentiating between comparable documents.
2. **Clustering Algorithm Modifications.** The algorithm should consider pairing only different language documents.

Translation Equivalents Table

The accuracy of the comparable document selection depends directly on the quality of the translation equivalents table.[4] The translation equivalents table contains only

[4]We used a clean dictionary containing more than 1.5 million entries for RO-EN; for other language pairs, dictionaries were built in the ACCURAT project using GIZA++.

content-word translations of lemmas with *N*-gram maximum lengths. Considering the fact that not all the translation equivalents have the same discriminative power for selecting comparable documents, the translation equivalents table was filtered using a maximum translation equivalents entropy threshold (0.5 in our case). Using this filtering method, light verbs, nouns with many synonyms, and other spurious translation equivalents were removed.

Document Collection
The documents were tagged and lemmatized. Considering only the content words, for each *n*-gram from the document collection, a set of translation equivalents was selected from the translation equivalents table. For example, the translation equivalents for "acetic acid" in both English and Romanian are: "acetic–acetic," "acetic acid–acid acetic," "acid–acid."

Clustering for Comparable Documents Identification
This technique relies on the supposition that translation equivalents can be used as common elements that would make documents in different languages similar. We chose an agglomerative clustering algorithm. We tested several simple distance measures like Euclidean distance, squared Euclidean distance, Manhattan distance, and percent disagreement. We found that percentage disagreement better differentiated comparable and noncomparable documents. Considering the document vectors $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$, in which elements are 1 or 0 depending on whether the corresponding vocabulary term belongs to the document or not, the percentage disagreement is computed as

$$d(x, y) = \frac{\sum_{i=1}^{n} x_i \neq y_i}{n},$$

where n represents the size of translation equivalent table. The distance measure has the restriction that the compared documents have to be in different languages. This simple distance measure gave us a precision of 72% (with a maximum translation equivalent entropy threshold of 0.5 and a maximum of 3 translation equivalents per document term) on the collection of 128 English and Romanian Wikipedia documents described.

3.3.1.2 Measuring Similarity in Inter-language Linked Documents

Using methods such as those described in the last section, we are able to extract and align Wikipedia articles describing the same topic. However, Wikipedia articles on the same topic may still be of varying degrees of comparability (Filatova 2009; Gamallo and Garcia 2012). Parts of the content may be translation equivalents (i.e., parallel), while other parts may have been developed independently and share little thematic or lexical overlap. For tasks, such as cross-language information retrieval (CLIR) or SMT, the degree of similarity between texts will affect the quality of translation resources subsequently created; usage of nonsimilar documents will introduce noise and reduce MT performance (Lü et al. 2007). Therefore, we

developed an approach to measure content similarity within inter-language-linked or otherwise paired Wikipedia articles in order to filter out noncomparable article pairs and therefore obtain higher quality comparable corpora. Document similarity is analyzed by aligning sentences within the document pairs that contain overlaps of links (Adafre and de Rijke 2006) and cognates. Documents with a low number of aligned sentences were regarded as noncomparable and filtered out.

While the method described below will work for measuring similarity between any pair of Wikipedia articles, we only tested the method on pairs of inter-language-linked articles. First, we used the Wikipedia Extractor Tool[5] to extract the contents of all inter-language-linked documents from the Wikipedia dumps of March 2010. These documents were used as the dataset as they are on the same topic and therefore have a higher probability of containing comparable segments than those not paired by Wikipedia. Of course, documents paired by the methods described in Sect. 3.3.1.1 could also be examined. The method contains five main processes (as shown in Fig. 3.1):

1. **Bilingual Lexicon Extraction**

We built a bilingual lexicon by extracting titles of inter-language-linked articles in Wikipedia. For example, the English Wikipedia document entitled "`asteroid`" is paired with the Slovenian document with the title "`asteroidov`"; therefore, the pair "asteroid–asteroidov" is added into the English–Slovenian bilingual lexicon. While the resulting bilingual lexicon did not contain translations of all possible words in a regular dictionary, it contained important terms such as named entities. This bilingual lexicon was then used as a "dictionary" to aid translation; this eliminated the need of using any other linguistic resources for the translation process.

2. **Document Filtering**

Most Wikipedia documents contain descriptions of specific topics in the form of paragraphs and lists. We regarded this information as the main document content and filtered out additional information in the document, such as images, image captions, tables, and infoboxes (shown in Italic in Fig. 3.2). After the filtering step, the resulting documents contained sentences and link information only (shown in [[anchors texts]]), as shown in Fig. 3.3.

3. **Sentence Splitter**

As discussed previously, this retrieval method identifies comparable documents by analyzing similarity between the sentences, that is, comparable documents are those containing parallel or comparable sentences. Therefore, each document was split into sentences to enable further analysis, resulting in documents that contain one sentence per line.

[5]Wikipedia Extractor tool is available for download in the ACCURAT project website (Paramita et al. 2012).

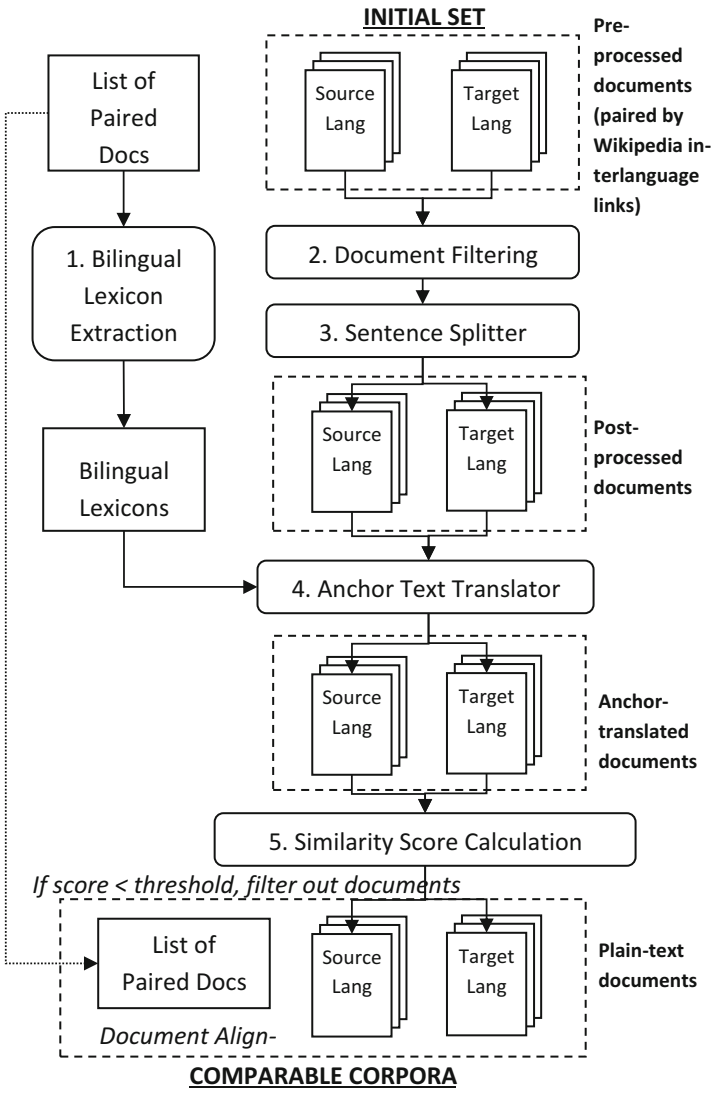


Fig. 3.1 Wikipedia retrieval processes

4. **Anchor Text Translator**

In this step, we utilized the extracted bilingual lexicon to translate all anchor texts from the source language into the target language. In the previous example, after replacing all anchor texts into English, we obtained the document shown in Fig. 3.4 (all the replaced anchors are shown in bold). If an anchor text did not exist in the bilingual lexicon, no translation was performed and the original text was used instead.

```
{{atsauces+}}
{{Warbox
|conflict=Desmitdienu karš <br>([[Dienvidslāvijas kari]])
|campaign=
|colour_scheme=background:#bbcccc
|image=[[Attēls:Radenci, July 28, 1991.jpg|300px]]
|caption= Sadursmes vieta Radencos 28. jūnijā.
|date=[[27. jūnijs]] - [[6. jūlijs]], [[1991]]
...
|}}'''Desmitdienu karš''', arī '''Slovēnijas neatkarības karš'''
([[slovēņu valoda|slovēņu]] - ''Slovenska osamosvojitvena vojna''
(''Slovēnijas neatkarības karš''), [[serbu valoda|serbu]] - ''Рат
у Словенији'' (''Karš Slovēnijā'')) - bruņots konflikts no 1991.
gada 27. jūnija līdz 6. jūlijam starp [[Slovēnija|Slovēniju]] un
[[Dienvidslāvija|Dienvidslāviju]], kura rezultātā nodibinājās
neatkarīgā Slovēnijas valsts.

Desmitdienu karš bija iesākums daudz asiņainākiem kariem citās
Dienvidslāvijas republikās. Tas bija arī gandrīz vienīgais no
Dienvidslāvijas kariem, pēc kura neviena no pusēm netika
apsūdzēta [[kara noziegumi|kara noziegumos]].
...
```

Fig. 3.2 An example pre-processed Latvian document

```
Desmitdienu karš, arī Slovēnijas neatkarības karš ([[slovēņu
valoda|slovēņu]] - Slovenska osamosvojitvena vojna (Slovēnijas
neatkarības karš), [[serbu valoda|serbu]] - Рат у Словенији (Karš
Slovēnijā)) - bruņots konflikts no 1991. gada 27. jūnija līdz 6.
jūlijam starp [[Slovēnija|Slovēniju]] un [[Dienvidslāvija|Dien-
vidslāviju]], kura rezultātā nodibinājās neatkarīgā Slovēnijas
valsts.

Desmitdienu karš bija iesākums daudz asiņainākiem kariem citās
Dienvidslāvijas republikās. Tas bija arī gandrīz vienīgais no
Dienvidslāvijas kariem, pēc kura neviena no pusēm netika
apsūdzēta [[kara noziegumi|kara noziegumos]].
...
```

Fig. 3.3 Postprocessed Latvian document

5. Calculation of Similarity Score

In this step, we calculated the similarity score of each document pair by pairing sentences that contained the highest word overlap. No other translation was performed in the text. Therefore, sentences were paired if they shared the same anchor texts. Moreover, sentences that shared overlapping words, such as named entities or numbers, were also taken into account. For each sentence from the smaller document of the pair (regardless of the language), we aimed to find the best matching

```
Desmitdienu karš, arī Slovēnijas neatkarības karš ([[slovene lan-
guage]] - Slovenska osamosvojitvena vojna (Slovēnijas neatkarības
karš), [[serbian language]] - Рат у Словенији (Karš Slovēnijā)) -
bruņots konflikts no 1991. gada 27. jūnija līdz 6. jūlijam starp
[[slovenia]] un [[yugoslavia]], kura rezultātā nodibinājās neat-
karīgā Slovēnijas valsts.

Desmitdienu karš bija iesākums daudz asiņainākiem kariem citās
Dienvidslāvijas republikās.

Tas bija arī gandrīz vienīgais no Dienvidslāvijas kariem, pēc
kura neviena no pusēm netika apsūdzēta [[war crime]].

...
```

Fig. 3.4 Documents with the anchor texts translated

```
Desmitdienu karš, arī Slovēnijas neatkarības karš (slovēņu —
Slovenska osamosvojitvena vojna (Slovēnijas neatkarības karš),
serbu — Рат у Словенији (Karš Slovēnijā)) — bruņots konflikts no
1991. gada 27. jūnija līdz 6. jūlijam starp Slovēniju un Dienvid-
slāviju, kura rezultātā nodibinājās neatkarīgā Slovēnijas valsts.

Desmitdienu karš bija iesākums daudz asiņainākiem kariem citās
Dienvidslāvijas republikās. Tas bija arī gandrīz vienīgais no
Dienvidslāvijas kariem, pēc kura neviena no pusēm netika
apsūdzēta kara noziegumos.

…
```

Fig. 3.5 Plain-text documents

sentence from the bigger document. The comparability score for a document pair is computed as the average score of all paired sentences:

$$\text{comparability score} = \frac{\sum_{i=0}^{n} S_i}{n},$$

where S_i is the Jaccard similarity score of word overlap between sentence i and the best matching sentence (or 0 if unpaired), and n is the number of sentences in the shorter document. Based on preliminary experiments, we selected a threshold value of 0.1; all document pairs scoring below the threshold were filtered out, and the plain-text versions of the remainder were included in the final set of comparable corpora (as shown in Fig. 3.5).

Evaluation

To evaluate the performance of the proposed similarity measure, we first created an evaluation set by gathering human judgments on similarity of 800 document pairs (Paramita et al. 2012). Afterward, we compared the performance of this anchor-based method to one using freely available MT systems (further described in Paramita et al. 2012) and found that they correlated well with each other (rho = 0.744, $p < 0.01$). Our approach correlated to a lesser degree with human judgments

(rho = 0.353, $p < 0.01$); nevertheless, it captured some aspects of cross-language document similarity as judged manually. More detailed evaluation results are reported in Paramita et al. (2012).

These results demonstrate the potential benefit of mining inter-language links from Wikipedia for under-resourced languages. Using this approach, we were also able to create comparable corpora for all ACCURAT languages without the need for any linguistic resources. This method is applicable for any languages contained in Wikipedia.

3.3.2 Comparable Corpora Collection from News Articles

Events occurring around the world are reported in various languages. If an event is reported in two (or more) different languages, it is highly likely that the reporting news articles will share textual units, such as sentences or phrases, that are translations of each other. News articles, therefore, are promising sources for building comparable corpora. We describe the technique used to collect these articles in this section. For further details, see Aker et al. (2012).

First, in order to collect comparable news articles for building comparable corpora, we collected news article titles through Google News Search and RSS News feeds. We only downloaded titles of current news articles and did not search for articles in news archives or on the entire Web in order to reduce the noise in the pairing process.

This collection of news article titles should be high recall, that is, contain as many potentially useful titles as possible. Thus, during this initial process of collecting our "working material," we focused on high recall and ignored precision, that is, the proportion of the collected titles that were actually comparable with each other. After collecting these titles, we applied different heuristics to pair them.

To collect the title corpora we adopted the following processes:

1. We first collected initial corpora of titles from news article monolingually using Google News. For each language, we iteratively downloaded titles from news in different topic categories, such as economics, world news, politics, etc. We set the iteration time to 15 min. Apart from the title for each search result, we also gathered information about the date and time of publication, the URL to the actual article and another URL used by Google News to show all related articles about the same topic in a cluster—we refer to this URL as a *cluster URL*. We refer to the titles obtained by this first step as the *initial corpora* (Fig. 3.6).
2. For each title in the initial corpora, we utilized the clustering information of similar news articles in Google News to collect other similar articles. More precisely, we followed the cluster URL and downloaded the first 30 articles from the cluster. We refer to these corpora as *news corpora 1*. Clearly, following these two steps, one can collect as many titles as one wants spanning a period of time. In our case, this period was a week, that is, since our method was set to run for 1 week, the first downloaded news article was 1 week old.

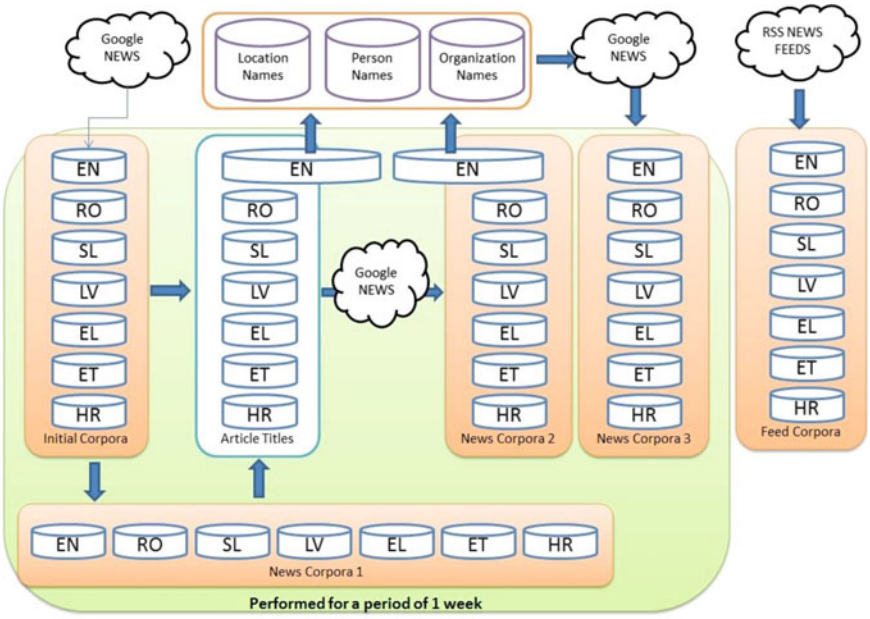


Fig. 3.6 News retrieval processes

3. We then used the titles from the *initial corpora* and *news corpora 1* as queries and performed a monolingual Google News search. We extracted the titles from the search results and these constituted *news corpora 2*. When performing this search, we restricted the date of the search to a maximum of one week from the moment the search is performed. Furthermore, we collected *news corpora 2* in parallel with the *initial corpora* and *news corpora 1*. As shown in Fig. 3.6, we ran these processes for a week.
4. Next, we further expanded the collection of article titles to create *news corpora 3*. For this, we took the article titles from the *initial corpora*, *news corpora 1* and *news corpora 2* for the English collection only. We parsed them for named entities such as person, location, and organization names.[6] For each named entity type, we translated the entities into the language in which the search will be performed (using Google Translate) and performed a Google News Search using the translated entity as a query. The search was restricted to a maximum of one week prior to the publication date of the article.
5. Finally, Google News does not support all languages equally. Languages such as German or Greek are well supported by Google News, that is, articles of different news agencies are pre-processed and listed by Google News. However, this is not the case for languages such as Latvian, Lithuanian, Estonian, etc. Due to this fact,

[6]For named entity parsing, we use OpenNLP tools: http://incubator.apache.org/opennlp/

there is a data scarcity problem in those languages. To overcome this problem, we manually identified a good number of RSS News feeds for each language from which we extracted similar information to that found in Google News Search.

The processes above were performed iteratively once a week until sufficient data had been gathered for the corpora.

Document Alignment

In the alignment phase, the goal is to identify comparable article pairs by matching the article titles from the collected corpora; contents of these paired articles are then downloaded to create a comparable corpus. This is hugely more efficient than downloading and comparing full document contents. Of course it is also noisier, but recall problems can be tolerated, given the volume of available data, and precision problems can be addressed by comparing full document contents in a postprocessing stage.

Matching news by title similarity (TS) is performed by computing the cosine similarity across the titles' term frequency vectors. Thus, each title pair is scored between 0 and 1. Before computing the cosine measure, we ensure that both titles (after removing the stop words) have at least 5 content words on both sides. We have experimentally observed that a news title with at least 5 content words is best to represent the actual document content. We translate the foreign title into English using Google Translate. We also combine TS with the following heuristics to investigate their impact on the quality of the produced pairs:

- HS: Each article title pair is scored by $1/(h+1)$, where h is the time difference in hours, with $h \in [0, \ldots, 23]$. Articles published within the same hour get a score of 1. If the time difference is greater than 23h, then HS is set to 0.
- DS: We score each article title pair by the publishing date difference between the two articles,$1/(d+1)$, where d is the date difference, with $d \in [0, \ldots, 7]$). Articles published on the same date get a score of 1. We set DS to 0 when the publishing date is greater than 7 days.
- TLD: We score each article title pair by $1/(w+1)$, where w is the difference in content word count (starting from 0). Article titles with the same length get a score of 1.

Evaluation

We created different combinations of the heuristics and evaluated the quality of the results. We use a linear combination of each heuristic with equal weight. Each combination produces a ranked list of article title pairs. The following list summarizes the different heuristic combinations:

- TS: Title cosine similarity
- TS_HS: Title cosine similarity and time difference
- TS_DS: Title cosine similarity and date difference
- TS_TLD: Title cosine similarity and title length difference
- TS_TLD_HS: Title cosine similarity, title length difference, and time difference
- TS_TLD_DS: Title cosine similarity, title length difference, and date difference

Table 3.1 Ranking correlation between the different heuristic combinations for the English–German pairs

	TS	TS_HS	TS_DS	TS_TLD	TS_TLD_HS	TS_TLD_DS
TS	–	1	0.94	1	0.99	0.73
CS	0.23	0.19	0.15	0.09	0.16	0.17

We performed a ranking comparison between the different ranked lists of title pairs and human assessment on the aligned articles.

Evaluation: Ranking Order

We compared the quality of the pairs produced by the different heuristic combinations with the ones obtained when the article content is used. To compute the content similarity, first we consider the union of the top 1K pairs of titles ranked by each one of the six aforementioned methods. The maximum number of pairs in the union is 6K. Following the corresponding URLs, we downloaded content (text) of the article pairs and computed the cosine similarity over term frequency vectors of the entire article. We use an HTML parser[7] to extract text from the HTML documents. Before comparing the article contents, each foreign article text is translated into English using Google Translate. The comparison of article texts produced another ranked list of article pairs, which we refer as the CS list.

We compared the rankings of each similarity heuristic using Kendall's τ. Kendall's τ values close to 1 reflect rankings very similar to each other, while values very close to 0 reflect independent rankings. The results are shown in Tables 3.1 and 3.2. As one can observe, in both German and Greek, the results in the first row show that the rankings produced by different heuristic combinations correlate very highly with the original title similarity. Thus, date, time, and title length do not dramatically change the matching process. On the other hand, the correlation between CS and the other heuristic combinations is rather low as shown in the second row of both Tables 3.1 and 3.2. Thus, using the title (along with other meta-data) does not produce the same matches as when using the entire article. The next step is to investigate how humans judge the different rankings produced for the two cases (title similarity and meta-data versus content similarity).

Evaluation: Human Judgment

To select the evaluation documents, we employed a "pooling" approach similar to the one used in TREC3 and ImageCLEF4, and constructed a depth-30 pool by considering the union of the top 30 document pairs coming from each one of the approaches under consideration: TS, TS_HS, TS_DS, TS_TLD, TS_TLD_HS, TS_TLD_DS, and CS.

The document pairs in the pools for the two languages were shown to two native German and eight native Greek speakers, respectively. All assessors were fluent in English. Both German participants judged all the pairs in the English–German pool.

[7]Boilerpipe—http://code.google.com/p/boilerpipe/—is used to extract the textual content from the URL.

Table 3.2 Ranking correlation between the different heuristic combinations for the English–Greek pairs

	TS	TS_HS	TS_DS	TS_TLD	TS_TLD_HS	TS_TLD_DS
TS	–	1	0.82	1	0.95	0.78
CS	0.11	0.21	0.14	0.18	0.25	0.25

Fig. 3.7 Evaluation tool

In case of the Greek experiment, each quarter of the pool was shown to two different assessors.

We asked assessors to judge the comparability of each document pair using five comparability classes proposed by Braschler (1998): *same story*, *related story*, *shared aspect*, *common terminology*, and *unrelated*. We hypothesized that if two news articles were about the "same story," then it was more likely that they contained useful fragments for SMT than "unrelated" articles. The document contents were shown to the assessors side-by-side. The design of the assessment implementation is shown in Fig. 3.7.

From the results shown in Tables 3.3 and 3.4, we can see that the documents aligned with the title and meta-data information were mainly judged as being "same story" and "related story." For English–German the best performance was achieved when the title similarity was combined with the publishing time (TS_HS). In case of the English–Greek pairs, we can see that HS also plays an important role. The reason for the positive impact of HS may be that it reflects the way news events emerge. Two news articles published very close to each other in time are likely to report the same news event in the same way. However, over time a news event develops and changes, so any new report about it will differ from the first reports. Although the

Table 3.3 English–German document pair evaluation results

	Same story	Related story	Shared aspect	Common terminology	Unrelated
TS	74	24	2	0	0
TS_HS	88	12	0	0	0
TS_DS	76	18	6	0	0
TS_TLD	74	24	2	0	0
TS_TLD_HS	86	12	2	0	0
TS_TLD_DS	72	22	6	0	0
CS	75	21	4	0	0

Results of both assessors are taken together. The numbers are percentage values

Table 3.4 English–Greek document pair evaluation results

	Same story	Related story	Shared aspect	Common terminology	Unrelated
TS	50	12	24	7	7
TS_HS	56	15	20	5	4
TS_DS	62	8	30	0	0
TS_TLD	50	8	25	11	6
TS_TLD_HS	70	8	20	2	0
TS_TLD_DS	42	18	32	8	0
CS	29	19	32	6	14

Results of four assessors are taken together. The numbers are percentage values

new reports are also about the same general event, the contents differ from the first reports and become reports of related stories or reports that share only some aspects with the first ones.

This fact is supported by the results that any combination of heuristics without HS has higher "shared aspect" than the combinations with HS. The heuristic DS is also meant to capture news articles about the same story. However, since DS uses day level difference in scoring, it can only achieve similar performance to HS for stories that do not emerge very quickly. For English–Greek, the best performance is achieved when TS and HS are combined with TLD (TS_TLD_HS)—note that adding TLD to TS_HS in English–German leads to almost as good results as those obtained with TS_HS only. In general the heuristic TLD plays also a role in the title method. It ensures that titles with no length difference are scored higher than those that vary a lot in length. We computed the average title length difference for each language.[8] The English titles contain on average 6.8 content words, the German titles 6.5, and the Greek titles 5.8. These figures show that the English and Greek titles vary from each other more than the English and German ones. We think that this may explain why TLD has more impact on the English–Greek results than it has on the English–German ones.

[8]Titles, which have less than five content words, are not taken into consideration.

In the ranking results shown in Tables 3.1 and 3.2, we see that there is no correlation between the ranked list of article pairs produced by CS and the article rankings of the other heuristics. However, from the results shown in Tables 3.3 and 3.4, we see a different picture. In the case of the German–English pairs, the title similarity heuristics performed as well or better than the CS measure, while for the English–Greek pairs, title similarity heuristics performed significantly better than the CS method. However, note that this comparison is not exactly fair, since CS was tested on data pre-selected using the other heuristics. A nonbiased selection of data could lead to different CS performance. Finally, we also think that the poor performance of the CS method for English–Greek is due at least in part to the performance of the machine translation system. For German, the machine translation system is much better than for Greek, which is an under-resourced language, and this difference may influence the results significantly.

3.3.3 Comparable Corpora Collection from Narrow Domains

Automatic acquisition of comparable corpora from the Web is a challenging task on its own. Adding the additional constraint of acquiring comparable corpora for narrow domains significantly increases the challenge of the task, especially when dealing with less resourced languages.

Domain-specific multilingual comparable corpora can play an important role for improving the performance of MT systems in narrow domains, since they can potentially provide topic-specific parallel sentences/phrases, bilingual terms, and lexical resources, which are all most needed parallel data for training SMT and RBMT systems, respectively. In this manner, this type of language resource can compensate for the poor performance that the MT systems usually demonstrate when they are given narrow domain-specific texts to translate.

In this section, we discuss the general methodology we used for automatically acquiring comparable corpora in specific narrow domains from the Web and describe in detail each step of the underlying workflow.

Since this task involves collecting texts from a wide range of topical areas, we could not rely on a specific set of Web portals for providing the required volume of data. Instead, an open Web crawler had to be deployed in order to crawl the Web and collect useful HTML documents, without being limited by a Web domain restriction.

3.3.3.1 Acquiring Comparable Documents

To perform this task, two approaches were considered. The first approach was based on a crawling engine that focused on retrieving documents for each language separately, either (1) using a “domain-specific” crawler and therefore pre-classifying each document with a domain-type, or (2) using a general monolingual crawler and afterward using a domain-classifier on the collected texts. This

approach would be faster and easier to implement (given the fact that many good-quality monolingual crawlers already exist), but on the other hand, the task of aligning the texts (at the document level) would be significantly more complicated.

The second approach, on the other hand, would retrieve bilingual (or multilingual) documents with a good possibility that these documents are either parallel or topic-related. As a starting point, a parallel crawler could be used and later on expanded to enable the acquisition of not only parallel, but also comparable texts. This approach would be harder to implement, since it is not yet supported by one of the readily available crawlers. However, it would provide a significant boost to the alignment process, especially if we use a focused crawler (a crawler that implements a domain-specific filtering).

An initial pursuit of the second approach showed that although there are a small number of tools available for automatic detection of parallel HTML documents, they mostly exploit URL and HTML structure similarities (Resnik 1998; Esplà-Gomis and Forcada 2010). The main assumption is that two parallel HTML documents in a Web domain tend to have very similar URLs (sometimes the only differences are the two letters denoting the document's language) and/or the same HTML structure (most bilingual websites use the same template for displaying the same article in different languages). However, this is not the case with comparable documents. In reality, two comparable documents (especially when in different Web domains) will probably have completely different URLs and HTML structures and therefore, it is impossible to find them using such techniques. In addition, comparable documents will rarely have link connections to each other, which is another characteristic of parallel Web pages often used for finding parallel pairs. This led us to the conclusion that the easiest way to find and pair comparable pages would be to analyze the contents and classify whether or not they are topically related.

Due to this finding, we pursued the first approach by designing and implementing a focused crawling system, a Web crawler with a built-in lightweight topic classifier able to decide whether or not a given Web page is relevant to the desired topic. This approach is similar to Talvensaari et al. (2008) who used a focused crawling system to produce comparable corpora in the genomics domain for the English, Spanish, and German languages. Using such a system, we were able to produce narrow domain comparable corpora in all ACCURAT languages.

Focused Crawling

The implemented crawling system incorporates several subtasks ranging from bootstrapping the crawler to Web page parsing, classifying, and processing in order to extract the required information. The essential steps of this system are illustrated in Fig. 3.8.

Two critical resources were required to bootstrap the focused crawler:

1. **Topic definition:** a list of weighted terms that are considered representative of a specific topic. Such a list can be constructed by manually selecting a representative set of terms and assigning weights. Online resources (e.g., EuroVoc) provide sets of words in different languages assigned in specific thematic categories and

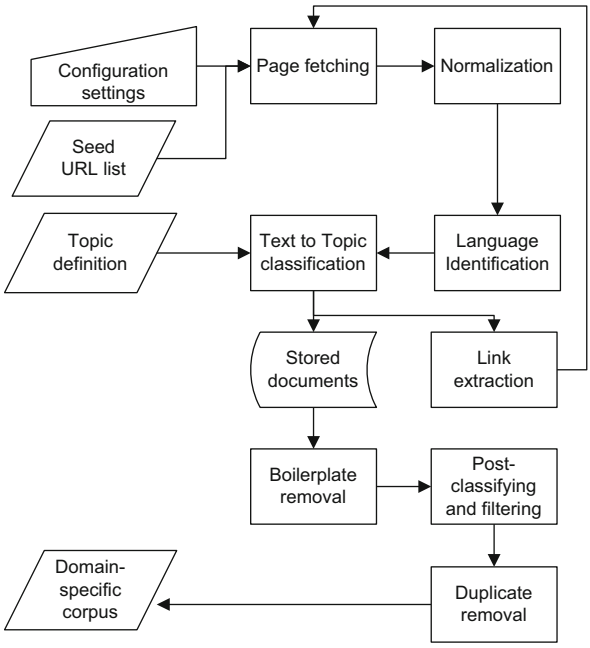


Fig. 3.8 Essential steps of a focused crawler system

therefore can greatly assist in this process. Alternatively, these lists can be automatically extracted by small topic-specific corpora using *tf-idf* and term extraction algorithms. The topic definition consists of triplets representing the weight, term, and the domain or subdomain it belongs to. A sample of the topic definition from the renewable energy domain is presented below:

100: natural resources=RenewableEN
100: natural processes=RenewableEN
100: biogas=RenewableEN
100: renewable power generators=RenewableEN

2. **Seed URL list construction:** a list of URLs that are considered highly relevant to the topic. This list can be semiautomatically assembled using search engines (e.g., Yahoo, Google). BootCaT's (Baroni and Bernardini 2004) tuple generation algorithm can also be used as follows: using the topic definition, a number of combinations of the topic terms are generated and then used as search queries in Google. The top 5 or 10 URL results from each search are selected as candidates for the final seed URL list.

In the next step, we reviewed several crawling algorithms:

1. The simplest and most common algorithm is Breadth-First (Pinkerton 1994). Each visited page has its links extracted and inserted in the crawler's schedule, known as the frontier. The frontier is filled in a First-In First-Out (FIFO) manner, meaning that the crawler visits links in the order they are found.

2. Fish-search (De Bra and Post 1994) attempts to slightly improve the Breadth-First algorithm. The main difference is that links, which were extracted from Web pages that were classified as irrelevant to the topic, are disregarded.
3. Best-First (Cho et al. 1998) was the logical evolution of Breadth-First. Instead of visiting pages in the order they are discovered, a classifier is used to estimate a Web page's relevance to the topic. This relevance score is then used to sort the frontier and therefore ensuring that Best pages will be visited First.
4. Shark-search (Hersovici et al. 1998) is the first algorithm that attempts to measure the relevance of anchor texts (the text that is visible as an HTML link) and uses it to score each link. This was considered an improved version of Fish-Search.
5. PageRank (Brin and Page 1998) introduced the concept of Web page popularity. Instead of scoring each page based on its content, this algorithm attempts to implement a ranking system by scoring a Web page depending on the number of other Web pages that have links to it. Therefore, a page is deemed popular when there is a high number of links that lead back to it. This approach is commonly used for indexing and ranking retrieved results.
6. InfoSpiders (Menczer and Belew 2000) is another algorithm that scores Web pages according to their relevance to the topic, but this time, the user may assess the relevance of the documents visited by InfoSpiders up to a certain point. A distinct population of agents attempts to "sense" their local neighborhood by analyzing the text of the document where they are currently situated. The behavior of these agents can be subsequently altered by user's feedback, therefore resulting to an adaptive environment.
7. Path algorithm (Passerini et al. 2001) again ranks pages based on their topic relevance, but also considers each page's distance from another relevant page, that is, starting from a relevant page, how many links must be followed before reaching the current one.

Since we were seeking a content-based solution, an algorithm that prioritizes most-relevant Web pages, a Best-First type of algorithm, seemed the obvious choice. However, anchor texts can often indicate if a link will lead to a relevant Web page as well; therefore, a hybrid solution was used by employing the basic idea of a Best-First algorithm and the anchor text scoring introduced by Shark-Search.

Normalization and Language Identification

The text normalization phase involves detection of the formats and text encodings of the downloaded Web pages as well as conversion of these pages into a unified format (plain text) and text encoding (UTF-8).

Meanwhile, in the language identification phase, each downloaded Web page is analyzed and its language is identified. Documents that are not in the language of interest are discarded. Lingua::Identify,[9] an open-source and flexible language identifier based on *n*-grams, is used for this task. Lingua::Identify did not originally support the Greek language; however, we provided a small corpus of Greek texts

[9] http://search.cpan.org/~ambs/Lingua-Identify-0.56/lib/Lingua/Identify.pm

(taken from JRC Acquis) to the developer of the tool, who released a new version of the identifier, which we used throughout the subsequent work.

Text Classification

Our goal was to implement a "lightweight" text classifier, so that it could be used during the crawling cycle without crippling the crawler's performance. In order to achieve a good compromise between crawling speed (larger number of Web pages visited) and classification quality (less irrelevant pages actually fetched), we used a simple string-matching algorithm for the comparison of each crawled and normalized Web page to the topic definition. By adopting the method described in Ardö and Golub (2007), the score of relevance c for each Web page is calculated as follows:

$$c = \sum_{j=1}^{4} \sum_{i=1}^{N} \frac{w_j^l \cdot w_i^t \cdot n_{ij}}{l_j},$$

where N is the number of terms in the topic definition, $w^l{}_j$ denotes the weight assigned to each location j of the HTML page (i.e., 10 for title, 4 for metadata, 2 for keywords, and 1 for main text), $w^t{}_i$ is the weight of term i, n_{ij} denotes the number of occurrences of term i in location j, and l_j is the number of words in location j.

The calculated score models the likelihood that the page under consideration contains text relevant to the target domain. Therefore, if the relevance score is under a predefined threshold, the page is classified as irrelevant and discarded. Otherwise, the page is stored and its links are extracted and added to the list of links scheduled to be visited.

In order to rank each link regarding the likelihood that the link points to a relevant Web page, we adopt an extension of the Shark-search (Hersovici et al. 1998) algorithm. Specifically, the potential score of each link is influenced by the estimated relevance of its anchor text (i.e., the visible, clickable text in a link), the text surrounding the link, and the source Web page.

Analytically, the score $s_{l_{i,j}}$ of the ith link of the jth Web page ($l_{i,j}$) is calculated by the following formula:

$$s_{l_{i,j}} = c_j + \sum_{m=1}^{N} w_m \cdot n_m,$$

where w_m denotes the weight of the mth term of the topic definition, n_m is the number of occurrences of the mth term in the anchor text and the text surrounding the link, and c_j denotes the score of the source Web page (i.e., this value is constant for every link found in the jth Web page).

Boilerplate Removal

Web pages often need to be cleaned of elements that are irrelevant to the content, such as navigation links, advertisements, disclaimers, etc. (often called "boilerplate"). Since we aim to collect comparable corpora useful for training MT systems, such parts of the HTML source are irrelevant. The algorithm we adapted for

boilerplate removal (Kohlschütter et al. 2010) used a set of shallow text features (link density, number of words in text blocks, etc.) for classifying individual text elements in a Web page as boilerplate.

Duplicate Detection
In (near-) duplicate detection, each new candidate document is checked against all other documents appearing in the corpus (e.g., by document similarity measures) before being added to the collection. An efficient algorithm for de-duplication was used for this task (Theobald et al. 2008). The algorithm represents each document as a set of spot signatures, that is, chains of words that follow frequent words as these are attested in a corpus. Each document is classified with respect to the cardinality of their set of spot signatures, and this significantly reduces the time complexity of the task.

Postfiltering
The crawling engine was able to score each visited page by using the topic definition and a text-to-topic classifier. However, the crawler's scoring strategy examined the whole HTML document (including title and metadata) and assigned high scores to pages with a high probability of leading to other related pages, regardless of whether or not the initial pages contained topically related content on their own. Therefore, in the final stage of the crawling framework, clean texts were examined and ranked in respect to the occurrences of words from the topic definition. This score, along with thresholds regarding file size and type, was used to filter out unwanted documents from the final collection.

The FMC Tool
The method described above has led to the development of the *Focused Monolingual Crawler (FMC) tool*, which has been extensively used within ACCURAT project in order to collect narrow domain bi(multi)lingual comparable corpora from the Web. The FMC tool is based on the *Bixo*[10] open-source Web mining toolkit. Given a narrow domain (topic) and a language, FMC has to be fed with two input datasets: (1) a list of topic definition multi-word term expressions and (2) a list of topic-related seed URLs. The user can configure FMC in a variety of ways, for example, set file types to download, domain-filtering options, self-terminating conditions, crawling politeness parameters, etc.

Crawling starts from the seed URLs and expands dynamically to other URLs, while lightweight text classification is performed on the Web pages being visited, so as to retrieve only those Web documents that are relevant to the chosen topic. Operations such as boilerplate removal, text normalization and cleaning, language identification, etc. are done at runtime; postcrawling processing steps (including deduplication, postclassification, and filtering) are also implemented.

The FMC output consists of the collected Web documents in both HTML and text format as well as their metadata. The metadata are stored in XML using a cesDOC format that can be validated against XCES standard schemas.

[10] http://bixo.101tec.com

To collect a pair of bilingual comparable corpora, two separate crawls are required (one per language). The comparability of the bi(multi)lingual documents retrieved is achieved by ensuring that, for each language, the FMC tool is made to return Web documents that are close to the same topic. By using FMC, a series of more than 35 comparable corpora on various narrow domains and in 6 language pairs (EN–LV, EN–LT, EN–HR, EN–RO, EN–EL, EN–DE) amounting to a total of more than 198M tokens have been constructed.

It should be noted that, overall, FMC is a multi-parametric, multi-thread tool featuring multi-faceted crawling functionality. Besides its standard functionality, that is, crawl the Web for a given topic (focused crawl) starting from seed URLs, FMC has additional operational modes that can prove useful in alternative crawling applications. For example, FMC can crawl for no topic (unfocused crawl) in user defined web domain(s) only (selective crawl), which provides an easy way to harvest significant amounts of parallel text in the case of multi-lingual websites. Also, FMC can collect the Web pages of given Websites without looking for a particular topic, which again is useful in case the targeted sites are known a priori to be focused on a specific topic. Furthermore, an FMC-based workflow for bilingual focused crawling can be easily established by (1) configuring FMC to crawl only within user-specified web domain(s), (2) providing two parallel lists of topic definition expressions, and (3) launching two FMC crawls, one for each language of a user-defined language pair. The comparability degree of the returned corpora in that case strongly depends on the comparability/parallelism of the given web domain(s).

For the formal evaluation of FMC tool's performance, the following two criteria have been used:

I. Check the relevance of each monolingual topic-specific corpus (collected by FMC) to the targeted topic.
II. Check the comparability degree of the bilingual topic-specific corpora collected by FMC.

Criterion II has been tested by using the ACCURAT comparability assessment tools described in Chap. 2 as well as the ACCURAT document alignment tools described in Chap. 2.

In order to assess FMC capacity for producing in-domain corpora from the Web (Criterion I), that is, for harvesting Web pages that are relevant to a target domain, we evaluated the "domainess" of the harvested documents by calculating the following relevance score:

$$s = \frac{1}{L} \cdot \sum_{i=1}^{N} w_i \cdot n_i,$$

where s score is calculated for every document in a narrow domain corpus collected by FMC, N is the number of terms in the corresponding topic definition file, w_i denotes the weight assigned to ith term in the topic definition file, n_i denotes the number of occurrences of the ith term in the document under consideration, and L is

the total count of tokens in this document. The *s* score reflects the multitude of terms in a given Web document, the occurrence frequency of the terms, etc. In fact, the *s* score models the "likelihood" that the harvested document contains text relevant to the target domain, in such a way that if the relevance score is greater than a pre-defined threshold, the document can be classified as relevant (Mastropavlos and Papavassiliou 2011). Extensive experimentation showed that an appropriate threshold for classifying documents according to this relevance score is 0.4.

For the automatic evaluation of the relevance to the topic (Criterion I), a method has been implemented that analyzes the probability density functions (PDFs) of the *s* score values, which FMC automatically calculates for every Web page it downloads. Once the PDF graph has been generated for a collected monolingual topic-specific corpus, then a high relevance of that corpus to the topic selected is indicated by a high lobe in the high score area ($s > 1$) of the horizontal axis of the graph. This is exemplified in Figs. 3.9 and 3.10, which depict the PDF of relevance scores for two corpora collected by using FMC (topic: "Wind Energy"; languages: EN and EL). So, a high lobe is observed around relevance score values of 12 for the EN corpus (Fig. 3.9); and again, a high lobe is observed around relevance score values of 12 for the EL corpus (Fig. 3.10). These two particular corpora admittedly demonstrate very good "domainess," as it has been found that only a very small subset (0.87%) of the EN collection of documents includes documents with relevance score lower than 1, whereas a nearly 5% of the EL collection of documents have relevance score lower than 1. Besides facilitating the evaluation of "domainess" of the collected corpora, plots of the PDF of relevance score provide a valuable feedback in

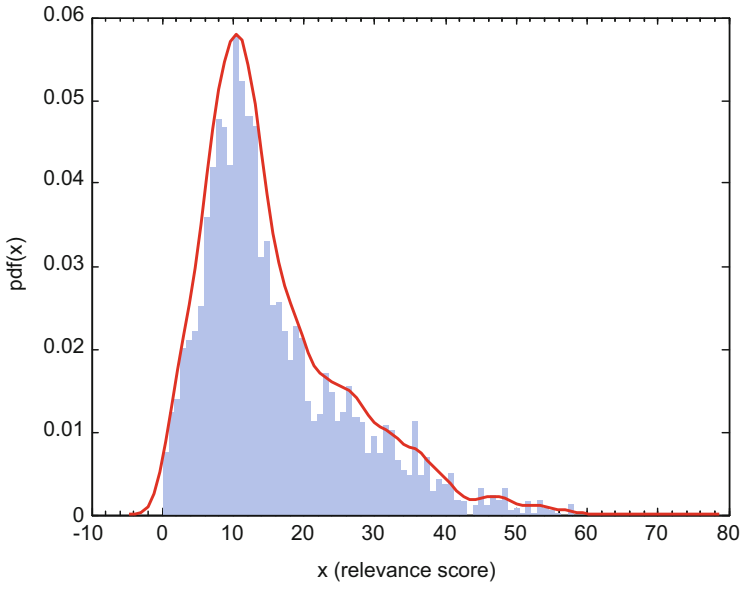


Fig. 3.9 Estimated PDF of the relevance score for the EN corpus in "Wind Energy" domain

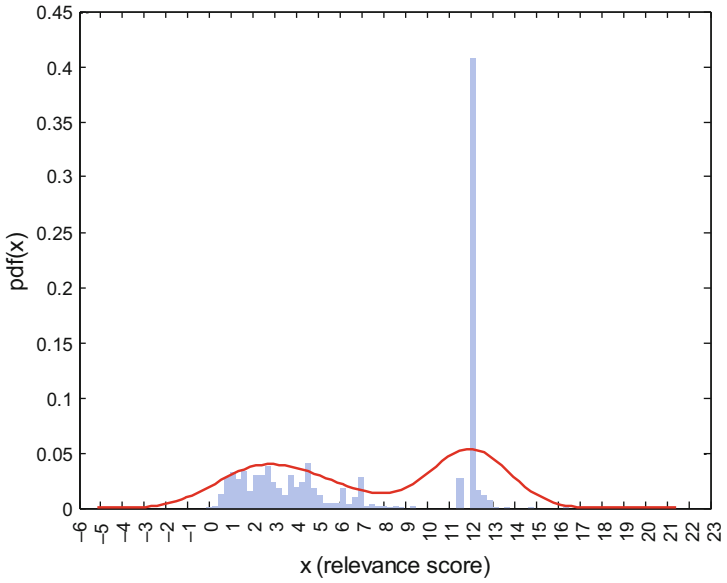


Fig. 3.10 Estimated PDF of the relevance score for the EL corpus in "Wind Energy" domain

re-defining a relevance score threshold in order to meet more effectively the requirements of a given focused crawling task.

The general conclusions on FMC's performance can be summarized as follows:

- The amount of parallel segments contained in the narrow domain comparable corpora collected by FMC significantly increases when FMC starts crawling from or crawls only in multi-lingual websites.
- Narrow domain comparable corpora collected by FMC seem to be more or less similar to general comparable corpora collected by other tools (Chap. 3) in terms of the amount of parallel segments they contain.
- Provided that Web does contain sufficient amount of MT exploitable comparable documents in the targeted topic and languages, the comparability degree of bilingual narrow texts gathered by the FMC significantly depends on
  - The quality[11] of the topic definition file
  - The quality[12] of seed URLs list file
  - The crawling duration[13]

[11]Have all topic-core-terms been included? Have other terms effectively pointing to the topic also been included? Does the topic definition file contain multi-word strong topic indicators? Have all terms been ranked consistently?

[12]Do seed URLs in the source language and seed URLs in the target language address highly comparable Web documents? Have multilingual sites, if any, been included?

[13]The longer the better, especially in cases where there are not too many Web documents relevant to the topic selected.

Moreover, there is always the option to run FMC with a user-customized configuration file, in which classifier parameters have been set to less loose[14] values than the default ones; this will very likely lead to performance gains, primarily in terms of "domainess" and secondarily in terms of comparability of the collected topic-specific corpora.

The following list provides general guidelines for FMC best usage scenarios:

- *Topic definition*: include all topic-core-terms; also, other topic indicators, preferably multi-word expressions; rank them consistently; align topic terms across languages.
- *Seed URLs list*: include multilingual sites, if any; avoid URLs pointing to texts in one language that are unlikely to match Web documents in the other language.
- *Crawl duration*: longer crawls are generally needed on under-resourced topics and languages.
- Combine *focused* or *unfocused* crawl and *selective* crawl modes to extract parallel data from multilingual Websites, when possible.

More information on FMC tool, including installation and how to use instructions, is given in Chap. 8 (Appendix).

3.3.3.2 Aligning Comparable Document Pairs

The approach described above has been developed to collect bilingual comparable documents on a specified topic. However, further processing is needed to align them at the document level. To perform this task, DictMetric (described in Chap. 2) can be used to calculate similarity between document pairs in order to pair those whose scores are above a specific threshold. Alternatively, latent Dirichlet allocation (LDA) can be used as an approach to align comparable documents (Preiss 2012).

LDA (Blei et al. 2003) is a generative probabilistic model where documents are viewed as mixtures over underlying topics, and each topic is a distribution over words. Both the document-topic and the topic-word distributions are assumed to have a Dirichlet prior. Given a set of documents and a number of topics, the model returns θ_i, the topic distribution for each document i, and φ_{ik}, the word distribution for topic k. We employed the publicly available implementation of LDA, JGibbLDA2 (Phan et al. 2008), which has two main execution methods: parameter estimation (model building) and inference for new data (classification of a new document). Both invocations produce the following:

$\varphi_{ij} : p(\text{word}_i \mid \text{topic}_j)$
$\theta_{jk} : p(\text{topic}_j \mid \text{document}_k)$
tassign: a deterministic topic-word assignment for each word in every document.

[14]For example, increase "*Minimum unique terms that must exist in clean content*" from default value of 3–5.

The LDA topic models are created from a randomly selected tenth of the Reuters corpus (Rose et al. 2002).[15]

Cross-Language Topic Identification

Being nondeterministic, multiple executions of the LDA algorithm are not guaranteed to (and do not) give rise to identical topics (even within one language). It is therefore not possible to build a topic model in the source language and the target language separately, as there is no clear alignment between their respective topics. Traditionally, parallel corpora are used to generate a language-independent topic-document distribution, from which polylingual topic models can be created so that the underlying topics are shared.

We translate each word from the source language topic model using the BING API and substitute the new wordmap, thus creating a target language topic model. While word distributions are clearly different across languages, and building a shared topic-document distribution to sample words from allows words to retain their language-specific distributions, our technique completely avoids the need for parallel corpora and merely requires the translation of the words in the LDA model (which can be performed using dictionary lookup, or NE lists instead of the BING API).

Identifying Comparable Documents

In order to identify comparable documents, given a source language document and multiple candidates of target language document, target language candidate documents are classified using the translated target language LDA model. Meanwhile, the source language document is classified using the source language LDA model. The candidate documents are then ranked according to the cosine similarity between the two vectors; the higher the similarity, the higher we rank the document. For every document in the source language, we selected the highest ranked target document as its pair.

We evaluated the performance of the LDA-based algorithm as described in Preiss (2012). Tested on a set of 100 randomly selected Czech–English Wikipedia inter-language-linked articles, LDA was able to align the documents with 83% precision although recall was low (35%). Finally, since this approach does not need parallel data for training, it can be applied to large document collections to find comparable documents across multi-lingual corpora.

References

ACCURAT Deliverable: D3.3, D3.4, D3.5.

Adafre, S. F., & de Rijke, M. (2006). Finding similar sentences across multiple languages in Wikipedia. *Proceedings of the EACL Workshop on New Text*, Trento, Italy.

Aker, A., Kanoulas, E., & Gaizauskas, R. (2012). A light way to collect comparable corpora from the Web. *Proceedings of LREC 2012*, 21–27 May, Istanbul, Turkey.

[15]LDA modeling can abstract a model from a relatively small corpus and a tenth of the original Reuters corpus is much more manageable in terms of memory and requirements.

Ardö, A., & Golub, K. (2007). *Documentation for the Combine (Focused) Crawling System*. http://combine.it.lth.se/documentation/DocMain/

Argaw, A. A., & Asker, L. (2005). Web mining for an amharic-english bilingual corpus. *Proceedings of the 1st International Conference on Web Information Systems and Technologies, WEBIST '05* (pp. 239–246). INSTICC Press.

Baroni, M., & Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the Web. *Proceedings of LREC 2004* (pp. 1313–1316).

Barzilay, R., & McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 50–57). Association for Computational Linguistics, Morristown, NJ.

Bharadwaj, R. G., & Varma, V. (2011). Language independent identification of parallel sentences using Wikipedia. *Proceedings of the 20th International Conference Companion on World Wide Web, WWW '11* (pp. 11–12), ACM, New York, NY.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research, 3*, 993–1022.

Braschler, P. S. (1998). Multilingual information retrieval based on document alignment techniques. *Research and Advanced Technology for Digital Libraries: Second European Conference, ECDL'98*, Heraklion, Crete, Cyprus, September 21–23, 1998: Proceedings, 183. Springer.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems, 30*(1–7), 107–117.

Callison-Burch, C., Koehn, P., & Osborne, M. (2006). Improved statistical machine translation using paraphrases. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 17–24). Association for Computational Linguistics, Morristown, NJ.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. *Ann Arbor MI, 48113* (2), 161–175.

Chakrabarti, S., Punera, K., & Subramanyam, M. (2002, May). Accelerated focused crawling through online relevance feedback. *Proceedings of the 11th International Conference on World Wide Web* (pp. 148–159). ACM.

Cho, J., Garcia-Molina, H., & Page, L. (1998). Efficient crawling through URL ordering. *Computer Networks and ISDN Systems, 30*(1–7), 161–172.

De Bra, P. M. E., & Post, R. D. J. (1994). Information retrieval in the World-Wide Web: Making client-based searching feasible. *Computer Networks and ISDN Systems, 27*(2), 183–192.

Dimalen, D. M. D., & Roxas, R. (2007). AutoCor: A query based automatic acquisition of corpora of closely-related languages. *Proceedings of the 21st PACLIC* (pp. 146–154).

Esplà-Gomis, M., & Forcada, M. L. (2010). Combining content-based and URL-based heuristics to harvest aligned bitexts from multilingual sites with bitextor. *The Prague Bulletin of Mathematical Linguistics, 93*, 77–86.

Filatova, E. (2009). Directions for exploiting asymmetries in multilingual Wikipedia. *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3 '09)*.

Fung, P., & Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP '04* (pp. 57–63), Citeseer.

Gamallo, P., & Garcia, M. (2012). Extraction of bilingual cognates from Wikipedia. *Computational Processing of the Portuguese Language* (pp. 63–72). Springer.

Ghani, R., Jones, R., & Mladenic, D. (2005). Building minority language corpora by learning to generate web search queries. *Knowledge and Information Systems, 7*(1), 56–83.

Hassan, A., Fahmy, H., & Hassan, H. (2007). Improving named entity translation by exploiting comparable and parallel corpora. *Proceedings of the 2007 Conference on Recent Advances in Natural Language Processing (RANLP)*, AMML Workshop.

Hersovici, M., Jacovi, M., Maarek, Y. S., Pelleg, D., Shtalhaim, M., & Ur, S. (1998). The sharksearch algorithm—An application: Tailored Web site mapping. *Computer Networks and ISDN Systems, 30*(1–7), 317–326.

Huang, D., Zhao, L., Li, L., & Yu, H. (2010). Mining large-scale comparable corpora from Chinese-English news collections. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 472–480). Association for Computational Linguistics.

Ion, R., Tufiş, D., Boroş, T., Ceauşu, A., & Ştefănescu, D. (2010). On-line compilation of comparable corpora and their evaluation. *Proceedings of the 7th International Conference Formal Approaches to South Slavic and Balkan Languages (FASSBL7)* (pp. 29–34). Croatian Language Technologies Society – Faculty of Humanities and Social Sciences, University of Zagreb, Dubrovnik, Croatia, October 2010.

Kauchak, D., & Barzilay, R. (2006). Paraphrasing for automatic evaluation. *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (pp. 455–462). Association for Computational Linguistics, Morristown, NJ.

Koehn, P. (2009). *Statistical machine translation*. Cambridge University Press.

Kohlschütter, C., Fankhauser, P., & Nejdl, W. (2010). Boilerplate detection using shallow text features. *The Third ACM International Conference on Web Search and Data Mining*.

Kumano, T., Tanaka, H., & Tokunaga, T. (2007). Extracting phrasal alignments from comparable corpora by using joint probability SMT model. *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)* (pp. 95–103).

Lü, Y., Huang, J., & Liu, Q. (2007, June). Improving statistical machine translation performance by training data selection and optimization. *EMNLP-CoNLL* (Vol. 34, pp. 3–350).

Marton, Y., Callison-Burch, C., Resnik, P. (2009). Improved statistical machine translation using monolingually-derived paraphrases. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 381–390). Association for Computational Linguistics.

Mastropavlos, N., & Papavassiliou, V. (2011). Automatic acquisition of bilingual language resources. *Proceedings of the 10th International Conference on Greek Linguistics*, Komotini, Greece

Menczer, F., & Belew, R. (2000). Adaptive retrieval agents: Internalizing local context and scaling up to the Web. *Machine Learning, 39*(2–3), 203–242.

Munteanu, D. S., & Marcu, D. (2002). Processing comparable corpora with bilingual suffix trees. *EMNLP '02: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing* (pp. 289–295). Association for Computational Linguistics, Morristown, NJ.

Munteanu, D. S., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Munteanu, D. S., & Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics* (pp. 81–88). Association for Computational Linguistics, Morristown, NJ.

Nakov, P. (2008). Paraphrasing verbs for noun compound interpretation. *Proceedings of the Workshop on Multiword Expressions, LREC-2008*.

Paramita, M., Clough, P., Aker, A., & Gaizauskas, R. (2012). Correlation between similarity measures for inter-language linked Wikipedia articles. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)* (pp. 790–797), Istanbul, Turkey.

Passerini, A., Frasconi, P., & Soda, G. (2001). Evaluation methods for focused crawling, Lecture Notes in Computer Science 2175, pp. 33–45.

Phan, X. H., Nguyen, L. M., & Horiguchi, S. (2008, April). Learning to classify short and sparse text and web with hidden topics from large-scale data collections. *Proceedings of the 17th International Conference on World Wide Web* (pp. 91–100). ACM.

Pinkerton, B. (1994). Finding what people want: Experiences with the Web Crawler. *Proceedings of the 2nd International World Wide Web Conference*.

Preiss, J. (2012). Identifying comparable corpora using LDA. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT '12)* (pp. 558–562). Association for Computational Linguistics, Stroudsburg, PA.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 519–526). Association for Computational Linguistics.

Resnik, P. (1998). Parallel strands: A preliminary investigation into mining the web for bilingual text. In D. Farwell, L. Gerber, & E. Hovy (Eds.), *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas (AMTA-98)*, Langhorne, PA, Lecture Notes in Artificial Intelligence 1529, Springer, October, 1998.

Resnik, P. (1999). Mining the web for bilingual text. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 527–534). Association for Computational Linguistics.

Rose, T. G., Stevenson, M., & Whitehead, M. (2002). The Reuters corpus volume 1 – from yesterday's news to tomorrow's language resources. *Proceedings of the Third International Conference on Language Resources and Evaluation* (pp. 827–832).

Sharoff, S., Babych, B., & Hartley, A. (2006). Using comparable corpora to solve problems difficult for human translators. *Proceedings of the COLING/ACL on Main Conference Poster Sessions* (pp. 739–746). Association for Computational Linguistics, Morristown, NJ.

Simard, M., Foster, G. F., & Isabelle, P. (1993). Using cognates to align sentences in bilingual corpora. In A. Gawman, E. Kidd, & P-Å. Larson (Eds.), *Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research: Distributed Computing (CASCON '93)* (Vol. 2, pp. 1071–1082). IBM Press.

Smith, J. R., Quirk, C., & Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *NAACL-HLT* (pp. 403–411).

Steinberger, R., Pouliquen, B., & Ignat, C. (2005). Navigating multilingual news collections using automatically extracted information. *Journal of Computing and Information Technology, 13*(4), 257–264.

Talvensaari, T., Pirkola, A., Järvelin, K., Juhola, M., & Laurikkala, J. (2008). Focused web crawling in the acquisition of comparable corpora. *Information Retrieval, 11*(5), 427–445.

Theobald, M., Siddharth, J., & Paepcke, A. (2008). SpotSigs: Robust and efficient near duplicate detection in large web collections. *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*.

Tomás, J., Bataller, J., Casacuberta, F., & Lloret, J., (2001). Mining Wikipedia as a parallel and comparable corpus. *Language Forum* (Vol. 34, No. 1, pp. 123–137). Bahri Publications.

Uszkoreit, J., Ponte, J. M., Popat, A. C., & Dubiner, M. (2010, August). Large scale parallel document mining for machine translation. *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 1101–1109). Association for Computational Linguistics.

Yu, K., & Tsujii, J. (2009). Extracting bilingual dictionary from comparable corpora with dependency heterogeneity. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (pp. 121–124). Association for Computational Linguistics, Stroudsburg, PA.

Zhao, S., Niu, C., Zhou, M., Liu, T., & Li, S. (2008, June). Combining multiple resources to improve SMT-based paraphrasing model. *Proceedings of ACL-08: HLT* (pp. 1021–1029). Association for Computational Linguistics, Columbus, OH.

Zhang, Y., Wu, K., Gao, J., & Vines, P. (2006). Automatic acquisition of Chinese-English parallel corpus from the web. *Proceedings of 28th European Conference on Information Retrieval ECIR 2006*, April 10–12, 2006, London.

Chapter 4
Extracting Data from Comparable Corpora

Mārcis Pinnis, Nikola Ljubešić, Dan Ştefănescu, Inguna Skadiņa, Marko Tadić, Tatjana Gornostaja, Špela Vintar, and Darja Fišer

Abstract Comparable corpora may comprise different types of single-word and multi-word phrases that can be considered as reciprocal translations, which may be beneficial for many different natural language processing tasks. This chapter describes methods and tools developed within the ACCURAT project that allow utilising comparable corpora in order to (1) identify terms, named entities (NEs), and other lexical units in comparable corpora, and (2) to cross-lingually map the identified single-word and multi-word phrases in order to create automatically extracted bilingual dictionaries that can be further utilised in machine translation, question answering, indexing, and other areas where bilingual dictionaries can be useful.

4.1 Introduction

Comparable corpora are sources of several different types of multi-lingual sub-sentential single-word and multi-word units that can be beneficial for machine translation system development and machine translation system adaptation to narrow domains. Therefore, we believe that methods that allow identifying such sub-sentential fragments in comparable corpora and allow extracting them from

Chapter editors: Mārcis Pinnis and Nikola Ljubešić

M. Pinnis · I. Skadiņa · T. Gornostaja
Tilde, Riga, Latvia
e-mail: Inguna.Skadina@tilde.lv

N. Ljubešić (✉) · M. Tadić
Faculty of Humanities and Social Sciences, University of Zagreb, Zagreb, Croatia

D. Ştefănescu
Research Institute for Artificial Intelligence, Romanian Academy, Bucharest, Romania

Š. Vintar · D. Fišer
Faculty of Arts, University of Ljubljana, Ljubljana, Slovenia

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_4

the comparable corpora are essential for MT system development, especially in areas where domain-specific parallel data is scarce or non-existing.

Research in the ACCURAT project was focussed on three types of sub-sentential units—terms, named entities and lexical units (i.e. general words). This chapter, therefore, presents methods and tools developed in the ACCURAT project for cross-lingual term extraction (TE), named entity recognition (NER), and bilingual lexica creation. We particularly investigated the developed method applicability to morphologically rich languages and under-resourced languages.

The chapter is further structured into three main sub-sections. Section 4.2 presents methods and tools for automatic bilingual term collection creation from comparable corpora (including a workflow for monolingual term extraction, term tagging in documents, and cross-lingual term mapping that is freely available as a part of the *ACCURAT Toolkit*). Section 4.3 presents *TildeNER*—an open source, freely available named entity recognition toolkit for NER system development using an NER model bootstrapping approach. Many cross-lingual natural language processing applications require bilingual lexica, but their compilation is still a major bottleneck in computational linguistics due to the lack of sufficient parallel corpora for many language pairs. Therefore, Sect. 4.4 focusses on methods for automatic extraction of bilingual lexica from comparable corpora.

4.2 Term Extraction, Tagging and Mapping for Under-Resourced Languages

Terms and their translations, as one of the sub-sentential units that can be frequently found in comparable corpora, have shown to be beneficial for statistical machine translation system adaptation to narrow domains (Pinnis and Skadiņš 2012). Therefore, this section presents three different directions of methods that allow extracting bilingual term pairs from comparable corpora. At first, in Sect. 4.2.1, we describe related work in the area of bilingual term extraction, followed by three methods for monolingual term extraction (a method for statistically, linguistically, and reference corpora motivated term extraction and tagging in documents, a method for statistically and lightly linguistically motivated term extraction from large corpora and a method for linguistically and reference corpora motivated multi-word term extraction) and two conceptually different methods for bilingual term mapping (a context-independent and a context-dependent method) in Sects. 4.2.2–4.2.4.

4.2.1 Related Work

Term extraction (TE) has been the focus of extensive work in natural language processing for over 25 years. Approaches may be characterised according to whether

they use local grammars, statistical co-occurrence measures or a combination of the two.

Systems like *LEXTER* (Bourigault 1992), *TERMS* (Justeson and Katz 1995) and *Termight* (Dagan and Church 1994) primarily use local grammar approaches in the form of hand-authored regular expressions over part-of-speech tags, while systems like Pantel and Lin (2001) make no use of linguistic information at all, using solely statistical co-occurrence measures between words. Often both approaches are combined in hybrid methodologies (Daille 1994; Dagan and Church 1994; Georgantopoulos and Piperidis 2000).

Despite the long history of term extraction, TE tools for Central and East European languages appeared later. Even nowadays, there is a significant gap between European analytical languages, on the one side, and synthetic ones, on the other side, due to their under-resourced status with the lack of necessary language resources and tools (Kruglevskis 2010).

For Croatian, the first experiments on collocation extraction and TE were presented by Tadić and Šojat (2003) using pointwise mutual information as the statistical co-occurrence measure for detecting collocations and multi-word term candidates. The *TermeX* system (Delač et al. 2009), which was developed later for Croatian and English, provides the possibility to use nine different co-occurrence measures for collocations.

For Lithuanian, the first experiments on TE were described by Zeller (2005). Grigonyte et al. (2011) evaluated the extraction of domain-specific terminology using four approaches: keyword cluster identification, keyword extraction with machine learning (ML), collocation extraction, and grammar-based. The collocation extraction and grammar-based approach appeared to be reliable in terms of recall but not precision.

For Latvian, the first experiment on TE showed that the linguistic method based on morpho-syntactic analysis is more appropriate than the statistical one that proved to be adequate for analytical languages (Kruglevskis and Vancane 2005). Semi-automatic TE has been applied to Latvian texts recently (Krugļevskis 2010).

In term tagging, the question '*What is a term?*' must be addressed not only from the termhood view but also from the unithood, that is the syntagmatic nature of a term, in the case of the so-called nested terms in particular—'*those <terms> that appear within other longer terms and may or may not appear by themselves in the corpus*' (Mima and Ananiadou 2000) (cf. Frantzi et al. 2000; Kageura and Umino 1996). Therefore, in a term candidate list, there may be overlaps between term candidates with different lengths, and the task of term tagging is to identify which are the correct terms in different given contexts.

Automatic bilingual term mapping from comparable corpora has received greater attention recently. Fung and McKeown (1997) and Rapp (1995) are considered the beginners of context-based term mapping methods based on the hypothesis that two terms are likely to be translations of each other when they occur in similar contexts. Many authors have experimented with different measures of context similarity (e.g. Chiao and Zweigenbaum 2002; Morin et al. 2007) and report up to 80% accuracy in finding the correct translation among the 20 best candidates.

Compositional analysis (Grefenstette 1999; Daille and Morin 2008), as an alternative method, may also be applied to the task of term mapping. It should be noted that these early approaches deal almost exclusively with single-word terms and also that nearly all authors conclude that the size and comparability of the corpora play a key role in achieving good performance. Different methods extend the bilingual term mapping task using cognate detection (Saralegi et al. 2008), while Lee et al. (2010) propose an expectation maximisation (EM) based hybrid model for term mapping. In view of bilingual lexicon extraction, symbolic, statistical and hybrid techniques have been implemented (Morin and Prochasson 2011). However, prior to the ACCURAT project, term mapping for morphologically rich, under-resourced languages received little attention in research (Weller et al. 2011).

4.2.2 Term Extraction, Tagging and Mapping with the ACCURAT Toolkit

This section is based on a publication by Pinnis et al. (2012) and describes the workflow for bilingual term extraction from a document-aligned comparable corpus using the tools created for the *ACCURAT Toolkit*. The workflow consists of three main terminology processing components:

1. *CollTerm* for monolingual term candidate extraction, which is described in Sect. 4.2.2.1.
2. *Tilde's Wrapper System for CollTerm (TWSC)* for monolingual term tagging in documents (i.e. term mark-up in documents using term candidate lists of different lengths extracted by *CollTerm*), which is described in Sect. 4.2.2.2.
3. *TerminologyAligner* for cross-lingual term mapping in a document-aligned and term-tagged comparable corpus, which is described in Sect. 4.2.2.3.

4.2.2.1 Term Candidate Extraction with CollTerm

CollTerm is a tool for collocation and term candidate extraction, and it combines two major approaches: (a) a linguistically motivated approach via morpho-syntactic patterns and (b) a statistically motivated approach via co-occurrence statistics and reference corpus statistics. The diagram of *CollTerm* and its processing flow, as depicted in Fig. 4.1, defines the four processing steps of the system: (a) linguistic (morpho-syntactic) filtering, (b) minimum frequency filter, (c) statistical ranking, and (d) cut-off method.

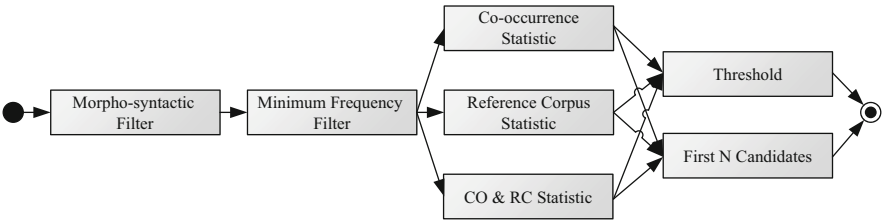


Fig. 4.1 Diagram of the CollTerm processing flow

Linguistic Filtering

CollTerm starts with linguistic filtering. Linguistic patterns of term candidates are defined in a phrase table containing regular expressions of acceptable phrases (see Fig. 4.2). In order to find valid term candidates, the regular expressions have to be crafted using the same tagset that is used by the morpho-syntactic (or part-of-speech) tagger during pre-processing of text documents. The phrase table allows defining agreements between the constituents of a phrase (as far as the tagset of the morpho-syntactic tagger allows).

Additionally, a stopword list can be used to filter out invalid term candidates. Stopword position restrictions can be specified in the phrase table. The example in Fig. 4.3 states that stopwords are not allowed to be the first and last tokens of trigram and four-gram term candidates.

The pattern lists for Latvian and Lithuanian, for instance, contain 120 different patterns. Initially, these patterns were automatically extracted from morphologically tagged (Pinnis and Goba 2011) texts in which terms were marked by human annotators. Since this initial list contained patterns for specific cases and not general language rules, the obtained patterns were then manually revised and generalised.

^[AG].fsn.*	^N...g.*	^N.fsn.*
^[AG].fsg.*	^N...g.*	^N.fsg.*
^[AG].fsd.*	^N...g.*	^N.fsd.*
^A.msg.*	^N.msg.*	^N.*
^A.mpg.*	^N.mpg.*	^N.*

Fig. 4.2 Fragment of Latvian morpho-syntactic patterns defining agreement between adjective (A) and noun (N) in gender (*m* masculine, *f* feminine), number (*s* singular, *p* plural), and case (*n* nominative, *g* genitive, *d* dative)

!STOP	*	!STOP	
!STOP	*	*	!STOP

Fig. 4.3 Example of morpho-syntactic patterns with stopword restrictions in a phrase table

Minimum Frequency Filter

The second phase consists of the minimum frequency filter, where all linguistically accepted phrases occurring less than the set minimum frequency are discarded from further processing, in order to limit the necessity of manual intervention of a domain expert and/or a terminologist who would have to evaluate the produced list and extract relevant terms (cf. frequency threshold in Frantzi et al. 2000). As document length and term frequency distribution vary from domain to domain, the minimum frequencies of acceptable term candidates have to be tuned (e.g. technical manuals are in general longer than news articles and may require higher minimum frequencies). The application of different minimum frequencies influences the recall and precision of term tagging.

Statistical Ranking

The third phase performs ranking of term candidates using co-occurrence and/or reference corpus statistics. Five different co-occurrence statistics can be used: Dice coefficient (DICE), modified mutual information (MI), chi-square statistics (CS), log-likelihood (LL), and *t*-score statistic (TS).

Table 4.1 shows the top 10 normalised bigram term candidate lemma sequences extracted from the Wikipedia article '*Automobile*' using TS with a minimum frequency of three for English and two for Latvian and Lithuanian (the minimum frequencies differ due to the article length difference).

Since unigrams cannot be ranked using co-occurrence statistics; inverse document frequency (IDF) (Spärck Jones 1972) scores of word lemmas from a reference corpus can be provided as additional input information. The reference corpus has to be large enough to represent the language. For instance, the Latvian corpus, from

Table 4.1 Top 10 normalised English, Latvian and Lithuanian term candidates consisting of two words and their scores obtained with the *t*-score statistic

English bigram term candidates		Latvian bigram term candidates		Lithuanian bigram term candidates	
Driverless car	1.00	caurejamības automobilis	1.00	antiblokavimas sistema	1.00
Propulsion technology	0.84	iekšdedze dzinējs	0.66	benzininis variklis	0.93
Internal combustion	0.83	protektors raksts	0.57	degimas variklis	0.87
Combustion engine	0.75	lauksaimniecība traktors	0.52	variklis cilindras	0.85
Automotive industry	0.73	tvaiks dzinējs	0.49	sauga diržas	0.84
Automotive market	0.64	ciets segums	0.48	dyzelinis variklis	0.82
Light truck	0.48	krava pārvadāšana	0.46	Lenktyninis Automobilis	0.78
Assembly line	0.40	dzinējs automobilis	0.38	vidus degimas	0.77
Automobile use	0.37	sacīkstes automobilis	0.37	vairas mechanizmas	0.75
Main article	0.36	ātrums rekords	0.33	įpurškimas sistema	0.72

which lemma IDF scores have been extracted, consists of Wikipedia articles (7.6 million tokens) and Web news articles (8.2 million tokens). Unigram term candidate ranking is calculated as a multiplication of the term's frequency within a document and the corresponding IDF score (Spärck Jones 1972). The term frequency—inverse document frequency (TF-IDF) ranking can also be applied to *n*-grams of length greater than one by calculating an average IDF score.

If the IDF score file is given and a co-occurrence statistic is used for *n*-gram term candidate ranking, a linear combination of TF-IDF and its co-occurrence statistic is computed. In the case where a non-dummy phrase file with linguistic patterns is given, the term candidates are extracted and ranked using all three information sources—the linguistic, the statistical representing co-occurrence data and the statistical representing reference corpus data.

Cut-Off Method

In the fourth phase, two different '*cut-off*' methods can be applied on the ranked candidate term list:

- Application of a term candidate ranking threshold (every term candidate below the threshold will be filtered out)
- Extraction of the first N candidates

The threshold '*cut-off*' method is more robust. It is less affected by document length differences and whether the document contains more or less valid term candidates after linguistic filtering.

The resulting list of term candidates can be exported as a sequence of lemmas (suitable for term tagging) or a sequence of the most frequent phrases in a text (more suitable for human inspection) with or without the lemma rankings.

4.2.2.2 Term Tagging in Documents

CollTerm creates a document containing a list of term candidates of fixed length (up to four tokens), where the term candidates are ranked according to one of the ranking methods. This requires CollTerm to be executed multiple times to cover single and multi-word terms. However, when combining the individual term candidate lists into one, the resulting term candidate list contains lexical overlaps between term candidates with different lengths, which are the so-called nested terms (cf. Frantzi et al. 2000). Consider the following example: '*A crash course in physics.*' As an output, CollTerm might find two term candidates: a unigram term candidate '*crash*' and a bigram term candidate '*crash course*' (both may be correct depending on the context). However, in order to capture a more specific representation of terms in the source document, only one of the term candidates is a valid term, for example in the example above, the intuitive selection is '*crash course*' if the document is about education. Our approach is application-oriented: in the case of machine translation, for example, the less specific term may cause wrong translation.

<TENAME>Servisa aprīkojumā</TENAME> ietilpst <TENAME>bremžu pārbaudes stends</TENAME>, <TENAME>motora diagnostikas ierīce</TENAME>, <TENAME>riteņu balansēšanas stends</TENAME>, <TENAME>amortizatoru pārbaudes stends</TENAME>, <TENAME>riteņu montēšanas stends</TENAME> u.c.

Fig. 4.4 Fragment of a term-tagged plaintext document in Latvian

Servisa	N	serviss	N-msg---------n----------f-	28	111	28	117	B-TERM	0.37
aprīkojumā	N	aprīkojums	N-msl---------n----------l-	28	119	28	128	I-TERM	0.37
ietilpst	V	ietilpt	Vp----3--i----------l-	28	130	28	137	O	0
bremžu	N	bremze	N-fpg---------n----------l-	28	139	28	144	B-TERM	0.45
pārbaudes	N	pārbaude	N-fsg---------n----------l-	28	146	28	154	I-TERM	0.45
stends	N	stends	N-msn---------n----------l-	28	156	28	161	I-TERM	0.45
,	T	,	T----------,	28	162	28	162	O	0

Fig. 4.5 Fragment of a term-tagged tab-separated document in Latvian

Due to ambiguities, we treat the term candidate lists as intermediate data and tag the terms in the source document with the tool *Tilde's Wrapper System for CollTerm* (*TWSC*), which has been specifically built in the ACCURAT project for term tagging purposes. As input, *TWSC* takes plaintext or pre-processed tab-separated (broken into sentences, tokenised, and part-of-speech or morpho-syntactically tagged) documents. *TWSC* then produces either term tagged plaintext, where term candidates are marked with tags *<TENAME>* (see Fig. 4.4 for an example), or tab-separated documents (see Fig. 4.5 for an example), where term candidates are marked with tags *B-TERM* (for the first token) and *I-TERM* (for the remaining tokens).

Within one term candidate list, it is possible to select the term candidate that is ranked higher. However, if the overlap is between term candidates from different (*n*-gram) lists, the selection is not straightforward. During our experiments, we have applied two methods for combining different *n*-gram term candidate lists into one. The first method prioritises longer *n*-grams, while the second combines all lists within one list using linear interpolation of term candidate confidence scores.

Term Tagging Evaluation for Latvian and Lithuanian

Evaluation of the term tagging tool *TWSC* for Latvian and Lithuanian was performed on manually annotated texts in the IT domain (software manuals, IT news, software reviews, etc.). The human annotated corpora were split in two parts—a development set and a test set. The former was used for tuning different parameters of *CollTerm* and *TWSC*, including (a) minimum *n*-gram frequencies, (b) *CollTerm* confidence score thresholds and (c) linear interpolation coefficients for the second term candidate list combination method. The human annotated corpora statistics for the Latvian and Lithuanian corpora are given in Table 4.2.

Table 4.2 Latvian and Lithuanian human annotated corpora statistics

	Latvian		Lithuanian	
	Test set	Development set	Test set	Development set
Tokens	15,230	7795	4547	2339
Proportion	66.15%	33.85%	66.03%	33.97%
Terms	2362	1127	751	380
Unigram terms	1540	656	417	198
Multi-word terms	822	471	334	182

Table 4.3 Latvian and Lithuanian human annotated corpora statistics

Language	Configuration	Term candidate ranking method	Minimum *n*-gram frequency for *n*-grams up to length 4				*R*	*P*	*F*1
Latvian	No threshold tuning	LL	1	1	3	3	70.66	42.52	53.09
		MI	2	1	1	2	63.89	46.83	54.05
		CS	11	3	2	3	39.88	59.85	47.87
	Threshold tuning	LL	1	1	3	3	**71.04**	41.70	52.55
		MI	2	1	1	2	57.49	52.74	**55.01**
		CS	11	3	2	3	23.24	**64.14**	34.12
	Prioritised	MI	2	1	1	2	63.89	46.83	54.05
	Linear interpolation	MI	2	1	1	2	63.04	42.58	50.83
Lithuanian	No threshold tuning	MI	1	1	1	1	65.11	46.97	54.57
		MI	4	1	2	2	59.79	53.26	**56.34**
		MI	10	3	2	3	42.08	55.24	47.77
	Threshold tuning	MI	1	1	1	1	**65.78**	47.78	55.35
		MI	4	1	2	2	55.79	52.70	54.20
		MI	10	3	2	2	37.55	**56.97**	45.26
	Prioritised	MI	4	1	2	2	59.79	53.26	56.34
	Linear interpolation	MI	4	1	2	2	60.32	41.79	49.37

Note: Bold values indicate the highest scores for convenience

During evaluation, parameters were tuned on the development set using an iterative approach. At first, we tuned the minimum *n*-gram frequency constraints using the prioritised list combination method and evaluated which ranking methods achieve the highest precision, recall, and *F*-measure (*F*1), without application of *CollTerm* confidence score thresholds. Then, we tuned the term candidate confidence score thresholds. Results using various term candidate ranking methods on the Latvian and Lithuanian test sets are given in Table 4.3.

For Latvian, the results show that the best recall was achieved with the LL ranking method (70.66%), the best precision was achieved with the CS statistic (59.85%), and the best *F*-measure was achieved with the MI ranking method (54.05).

However, the difference between the different methods is relatively insignificant. For instance the best achieved *F*-measure without confidence score threshold tuning with the LL statistic is 54.26 (54.23 on the development set) and with the MI statistic—54.05 (54.35 on the development set). As the development set for the Lithuanian language is relatively small, all term candidate ranking methods produced identical results for Lithuanian. Thus, the MI statistic was selected for further tuning of parameters for Lithuanian.

Table 4.3 shows that threshold tuning on the Latvian development set improves results (in terms of recall, precision and *F*-measure) on the test set as well. Although the evaluation shows an *F*-measure drop for Lithuanian, we believe that the size of the tuning corpus needs to be increased in order to reliably tune the parameters.

Finally, we tuned the interpolation parameters in order to achieve better *F*-measure with the interpolation-based term candidate list combination method. The results suggest that the prioritisation method significantly outperforms the interpolation-based method. Moreover, the tuned parameters suggest that longer *n*-grams are preferred (even in the interpolation-based method).

The lower performance of the interpolation-based method can partially be explained with the fact that we use a morpho-syntactic tagger for Latvian and Lithuanian TE. This allows us to define more complex phrase patterns requiring morpho-syntactic property agreements (e.g. agreement in gender, number, and case), which may already mean that longer *n*-grams are valid terms.

The tuning of parameters is very important when it is necessary to tune the system for specific tasks (e.g. document alignment, term mapping, information retrieval, question answering, etc.), because different tasks may require either higher recall or higher precision.

Term Tagging Evaluation for Croatian

The evaluation for Croatian was performed on a manually annotated corpus of automotive texts containing 15,603 tokens and 1430 (849 single-word and 581 multi-word) tagged terms, of which 652 were unique terms. While working on the Croatian data, we took into account the conclusions drawn from the evaluation on Latvian and Lithuanian by starting the tuning process with MI as the co-occurrence statistic and using the prioritisation method by tagging the terms in a greedy fashion. Besides tuning the parameters for Croatian, we also focussed our efforts on the effects of the morpho-syntactic phrase patterns used in linguistic filtering.

At first, we removed 32 tags that were longer than 4 tokens from the corpus and split it into a development set (7772 tokens and 645 terms) for tuning and a test set (7831 tokens, 753 terms) for final evaluation.

During the whole tuning process, we were maximising *F*-measure. The tuning was done in an iterative fashion similar to Latvian and Lithuanian. We started by searching for the optimal *n*-gram frequency thresholds. In this iteration, we improved the *F*-measure on the development set from 27.2 to 36.6. The next iteration

Table 4.4 Term tagging evaluation results for Croatian by gradually applying tuned parameters

Minimum *n*-gram frequency for *n*-grams up to length 4				Term candidate ranking method	*P*	*R*	*F*1
–				–	17.33	79.55	28.46
5	2	2	1	–	24.20	41.17	30.48
5	2	2	1	LL	39.07	35.59	37.25

focussed on the optimal co-occurrence statistic and its threshold values. In this step, *F*-measure was improved from 36.6 to 44.7. It is important to stress that the thresholds had a much higher impact on the performance increase than the statistic itself.

Finally, we evaluated the approach on our test set. We added the tuned parameter values one by one and, thereby, observed the impact of the tuning process in a more objective fashion. The results are given in Table 4.4. Obviously, both tuning steps improve results significantly.

An additional insight that we wanted to obtain during our work on Croatian data is the importance of the valid term phrase patterns. For that reason, we built three versions of the patterns:

- 24 detailed morpho-syntactic patterns. The example below specifies a four token term phrase consisting of a noun phrase (*adjective* + *noun*) in any case with an additional genitive noun phrase (*adjective* + *noun*) attached to it:

$$\text{^A.*} \quad \text{^Nc.*} \quad \text{^Af . . . g.*} \quad \text{^Nc . . . g.*}$$

- 12 more general rules obtained by simplifying the initial rules to just part of speech information (only the first letter of the morpho-syntactic tag). The example below describes the simplified previous example:

$$\text{^A.*} \quad \text{^N.*} \quad \text{^A.*} \quad \text{^N.*}$$

- 4 rules allowing any morpho-syntactic pattern combination. The example defines a four token phrase without any restrictions to morpho-syntactic properties:

$$.* \quad .* \quad .* \quad .*$$

Results obtained on the test set with the three phrase files are given in Table 4.5.

Table 4.5 Term tagging evaluation results for Croatian

Phrase file	*P*	*R*	*F*1
1	39.07	35.59	37.25
2	41.19	35.99	38.41
3	4.55	24.17	7.66

These results show that the simplified phrase file did even slightly outperform the initial one (probably because of some morpho-syntactic annotation errors). The finding that almost identical results can be achieved by using linguistic filtering based only on part-of-speech information is very important, since detailed morpho-syntactic taggers are not always available for under-resourced languages. However, the question remains if usage of more detailed phrase patterns, such as those applied on Latvian and Lithuanian (24 vs. 120 phrase patterns), would still increase the tagging quality in terms of precision. On the other hand, complete absence of linguistic filtering deteriorates the results drastically.

4.2.2.3 Term Mapping

To find possible translation equivalents of terms tagged in bilingual comparable corpora, the term mapping tool *TerminologyAligner* (*TEA*) was developed. Given term-tagged bilingual document pairs, the term mapping tool is designed to extract two lists of terms and to find pairs of expressions that are reciprocal translations. The tool analyses candidate pairs, assigning them translation scores based on (a) the translation equivalents and (b) the cognates that can be found in those pairs:

$$\text{translationScore}(\text{pair}) = \max(\text{ete}(\text{pair}), \text{ecg}(\text{pair})) \tag{4.1}$$

In this case, ete(pair) is the translation equivalence score and ecg(pair) is the cognate score for the expressions forming the candidate pair.

The translation equivalence score for two expressions is computed based on the word-level translation equivalents. Each word w_s in the source terminological expression e_s is paired with its corresponding word w_t in e_t so that the translation probability is maximal, according to a *Giza++* like probabilistic unigram translation dictionary (Och and Ney 2003). The score is normalised with the length of expression e_s. Still, we modify the denominator in order to penalise the pairs according to the length difference between source and target expressions:

$$\text{ete}(e_s, e_t) = \frac{\sum_{w_s \in e_s} \max_{w_t \in e_t} \text{wte}(w_s, w_t)}{\text{length}(e_s) + \frac{|\text{length}(e_s) - \text{length}(e_t)|}{2}}. \tag{4.2}$$

The cognate score for two expressions is computed as a modified Levenshtein distance (LD) between them. The expressions are normalised by removing double letters and replacing some character sequences: '*ph*' by '*f*', '*y*' by '*i*', '*hn*' by '*n*' and '*ha*' by '*a*'. This type of normalisation is often employed by spelling and alteration systems (Ştefănescu et al. 2011). Moreover, the score takes into account the length of the longest common substring of the two expressions, normalised by the maximum value of their lengths:

$$\text{ecg}(e_s, e_t) = \frac{1 - \frac{\text{LD}(\text{normalize}(e_s), \text{normalize}(e_t)) + 1}{\min(\text{length}(e_s) + 1, \text{length}(e_t) + 1)} + \frac{\text{length}(\text{LCS}(e_s, e_t))}{\max(\text{length}(e_s), \text{length}(e_t))}}{2}. \tag{4.3}$$

As probable translation equivalents, term pairs are selected only if the score of ete (pair) or ecg(pair) for the bilingual term pair is higher than a specified threshold. The threshold regulates the trade-off between precision and recall of TEA.

In order to evaluate the precision and recall of TEA, we used the EuroVoc thesaurus which is '*the thesaurus covering the activities of the EU and the European Parliament in particular*' (Steinberger et al. 2002). The EuroVoc thesaurus contained (at the time of our evaluation activities) a total of 6797 unique bilingual terms for every language pair (English–Croatian, English–Latvian, English–Lithuanian, and English–Romanian). For the English–Latvian language pair, we used two different unigram translation dictionaries to show the difference in recall when an in-domain or an out-of-domain dictionary is used.

The results (given in Table 4.6) show a significantly higher recall if an in-domain dictionary is used (a maximum of 31.03%), in contrast to an out-of-domain dictionary (a maximum of 18.11%). The obvious advantage to using the in-domain translation dictionary is a higher maximum precision of 99.12%, in contrast to 97.91% for the out-of-domain dictionary. However, we believe that, in a real-life scenario, the user won't have an in-domain dictionary at his or her disposal when trying to map terms in an under-resourced domain. Therefore, the recall and precision will be closer to the results obtained with the out-of-domain translation dictionary.

For other language pairs, we used only one translation dictionary (see Table 4.7). The results show that the highest *F*-measure is achieved for English–Romanian (23.48) followed by English–Croatian (21.66), and the lowest results have been achieved for English–Lithuanian (an *F*-measure of 19.99). For comparison, using a different in-domain dictionary (with higher term coverage) on English-Romanian *TEA* achieves an *F*-measure of 51.1 (Ştefănescu 2012).

Table 4.6 TEA evaluation results for English–Latvian on the EuroVoc thesaurus using in-domain and out-of-domain translation dictionaries

	In-domain dictionary			Out-of-domain dictionary		
Threshold	*R*	*P*	*F*1	*R*	*P*	*F*1
0.0	2.10	2.10	2.10	2.10	2.10	2.10
0.1	3.46	3.48	3.47	3.90	3.96	3.93
0.2	9.39	10.21	9.78	8.84	10.87	9.75
0.3	21.86	29.71	25.19	15.4	28.06	19.89
0.4	29.66	53.76	38.23	**18.11**	**55.00**	**27.25**
0.5	**31.03**	**79.52**	**44.64**	13.74	79.97	23.45
0.6	23.48	89.66	37.22	7.47	85.52	13.75
0.7	15.92	98.54	27.41	4.81	96.46	9.16
0.8	9.92	99.12	18.03	3.59	96.44	6.92
0.9	5.62	98.96	10.64	2.75	97.91	5.35
1.0	3.63	98.41	7.01	2.62	97.80	5.10

Note: Bold values indicate the highest scores for convenience

Table 4.7 TEA evaluation results for English–Lithuanian, English–Croatian and English–Romanian on the EuroVoc thesaurus

Threshold	English–Lithuanian (in-domain)			English–Croatian(out-of-domain)			English–Romanian (in-domain)		
	R	*P*	*F*1	*R*	*P*	*F*1	*R*	*P*	*F*1
0.0	1.79	1.79	1.79	3.94	3.94	3.94	6.08	6.08	6.08
0.1	2.91	3.07	2.99	5.02	5.35	5.18	7.22	7.54	7.38
0.2	5.40	7.40	6.24	7.31	9.93	8.42	9.08	10.31	9.65
0.3	9.96	25.52	14.33	11.71	28.92	16.67	12.06	19.35	14.86
0.4	**12.27**	**53.84**	**19.99**	**13.49**	**54.88**	**21.66**	14.21	38.36	20.74
0.5	10.37	79.21	18.34	12.08	81.94	21.05	**14.24**	**66.8**	**23.48**
0.6	7.00	93.15	13.03	8.50	95.54	15.62	12.81	88.34	22.38
0.7	5.00	96.87	9.51	6.50	98.66	12.20	10.11	95.82	18.29
0.8	3.35	98.28	6.49	4.99	99.41	9.50	8.37	99.13	15.44
0.9	2.15	99.32	4.21	4.08	99.64	7.83	6.19	99.76	11.66
1.0	1.47	80.00	2.89	4.00	99.63	7.69	6.06	99.76	11.43

Note: Bold values indicate the highest scores for convenience

4.2.2.4 Comparable Corpus Term Mapping Task

In order to show the capabilities of the term extraction, tagging and mapping process chain, we have run a full experiment on an English–Latvian comparable Web-crawled corpus in the automotive domain (car service manuals, reviews, marketing materials, etc.). The corpus was collected using the *ACCURAT Focused Monolingual Crawler* (*FMC*) and then bilingually aligned at the document level using the *ACCURAT DictMetric* comparability metric tool. *TWSC* was used to tag terms in both English and Latvian documents. In order to tag terms in English documents, the documents were pre-processed with *TreeTagger* (Schmid 1994). The comparable corpora statistics are given in Table 4.8.

Table 4.8 shows that many of the phrases in both Latvian and English documents have been marked as terms. This is due to the configuration, which in our experiment was set to achieve a better *F*-measure and not precision.

Once terms were tagged in all documents, we executed TEA on the aligned document pairs with a threshold of 0.6. TEA produced a total of 4414 term pairs, which were then filtered, preserving only the highest scored pair for each Latvian term, thus reducing the final pair count to 972. The results were then manually evaluated in terms of precision, as shown in Table 4.9.

Table 4.8 English–Latvian bilingual comparable automotive domain term-tagged corpus statistics

	English	Latvian
Documents	24,124	5461
Unique sentences	1,114,609	247,846
Tokens in unique sentences	15,660,911	3,939,921
Total number of term phrases	2,851,803	1,792,344
Unique term phrases	432,059	162,312

Table 4.9 TEA term mapping results with a threshold of 0.6 on the comparable English–Latvian automotive domain corpus

TEA translation equivalence score	Correct mapping	Incorrect mapping	Precision
≥0.60	714	258	73.46
≥0.65	501	115	81.33
≥0.70	331	38	89.70
≥0.75	228	24	90.48
≥0.80	142	14	91.03
≥0.85	93	10	90.29
≥0.90	50	9	84.75
≥0.95	36	7	83.72
≥1.00	30	7	81.08

Error analysis of TEA results shows five distinct error types:

- Term pairs are falsely aligned, because too many characters overlap, which results in a high cognate matching score. For instance '*auto mode*' in Latvian ('*auto fashion*' in English) and '*auto model*' in English get a score of 0.86. This type of error was present in 22.9% of all errors in the experiment.
- Multi-word terms are misaligned because of different word order. Consider the following example: '*water pressure*' and '*pressure water*'. These are two different terms. This type of error was evident in 2.3% of all misalignments.
- Terms are aligned with longer terms containing additional tokens that change the semantic meaning of the term. For instance '*modernie dīzeļi*' in Latvian ('*modern diesels*' in English) and '*modern diesel engine*' in English get a translation equivalence score of 0.8. This is the most frequent *TEA* error: 53.1% of all errors in our experiment were of this type.
- Terms are wrongly aligned with terms in the same language (e.g. English–English instead of the required English–Latvian), because no language identification is performed in the term level. It is frequent (especially in Web crawled documents) that a part of a document or some specific terms are written in another language. In the case of identical terms, this results in a high cognate translation score (e.g. in both Latvian and English documents, '*combustion process*' gets a cognate score of 1.0). This type of error was present in 11.6% of all misalignments.
- Terms are misaligned because of many out-of-domain translations in the probabilistic dictionary. If the dictionary is built from bad-quality parallel data or the dictionary features many translations of terms in other domains, false translation equivalents can be produced. For instance a '*notebook*' may be an electronic device or a book for notes depending on the context. We found that 2.7% of errors in our experiment were of this type.

The remaining 7.4% of errors were caused by either a combination of the above mentioned error types or by other less frequent cases.

Despite the errors, *TEA* achieved a precision of 73.46% with the translation equivalence threshold of 0.6, which can be increased up to 91% (as seen in Table 4.9) by using an out-of-domain dictionary.

4.2.2.5 Discussion

In this section, we presented methods and tools for TE, tagging, and bilingual mapping in comparable corpora. Term tagging has been applied and evaluated for Latvian and Lithuanian, and bilingual term mapping has been applied and evaluated for Croatian, Latvian, Lithuanian and Romanian.

The real-world scenario, in which bilingual terms were acquired from a comparable Web crawled corpus (in a domain unknown to the tools), shows that regardless of the relatively low precision of term tagging, bilingual term mapping in the presented process chain can achieve a precision up to 91%.

The defined process chain combines statistical and knowledge-based approaches and can be fine-tuned for specific tasks where different quality measures (recall or precision) apply. The TE tool CollTerm, the term tagging tool TWSC and the term mapping tool TEA presented in the paper are published under the Apache 2.0 license and are freely available as part of the ACCURAT Toolkit.

4.2.3 Experiments with English and Romanian Term Extraction

During the ACCURAT project, a different statistical (both reference corpora-based and co-occurrence measure-based) and linguistic method for TE was investigated for Romanian and English. The method differs in the fact that it allows identifying terms in length up to two words and, for bigram term candidates, the two words do not necessarily have to be consecutive (adjacent or next to each other). Contrary to the previously described method, which can be applied on documents, this method can be applied on a large corpus that has not been partitioned into many documents (e.g. machine translation system training data). The section is based on a publication by Ştefănescu (2012).

4.2.3.1 Single-Word Term Extraction

The task of single-word term extraction was approached by improving the method introduced by Damerau (1993). The method has been reported to yield very good results (Schütze 1998; Paukkeri et al. 2008). Damerau's approach compares the relative frequencies of words in the documents of interest (user corpus—C_U) to the

relative frequencies of the words in a reference collection (reference corpus—C_R). The original formula for computing the score of a word w is

$$\text{score}(w) = \frac{f(w, C_U)}{|C_U|} \div \frac{f(w, C_R)}{|C_R|}, \tag{4.4}$$

where $f(w, C)$ is the frequency of w in corpus C and $|C|$ is the total number of words in C. One can immediately notice that the score for a word is calculated according to the likelihood ratios of occurring in both corpora (that of the user and the reference). The main idea is to compare the maximum likelihood estimates (MLEs) computed on the user corpus to the ones on the reference corpus. Consequently, the reference corpus should be a large, balanced and representative corpus for the language of interest. Essentially, the MLE on such a corpus is equivalent to a unigram language model:

$$P_{\text{MLE}}(w) = \frac{f(w, C_R)}{|C_R|}. \tag{4.5}$$

In practice, such models are usually used in information retrieval to determine the topic of documents. Thus, Damerau's formula works by comparing two unigram language models.

However, it has been proved that due to data sparseness, the unigram language models constructed only by means of MLE behave poorly and that proper smoothing should be performed (Chen and Goodman 1999). To do this, we employ a variant of Good–Turing estimator smoothing (Kochanski 2006):

$$P_{\text{GT}}(w) = \frac{f(w, C_R) + 1}{|C_R| + |V_R|} \cdot \frac{E(f(w, C_R) + 1)}{E(f(w, C_R))}, \tag{4.6}$$

where V_R is the vocabulary (the unique words in C_R) and $E(n)$ is the probability estimate of the word to occur exactly n times.

Let us consider a slightly modified example from Kochanski (2006): let us say we have a (reference) corpus with 40,000 English words that contains only one instance of the word '*unusual*': $f(w, C_R) = 1$. Let us also say that the corpus contains 10,000 different words that appear once (i.e. $E(1)$= 10,000/40,000) and that we have 5500 words that appear twice (i.e. $E(2)$= 5500/40,000). Again, let us consider that the total number of unique words in the corpus $|V_R|$ is 15,000. The Good–Turing estimate of the probability of '*unusual*' is

$$P_{\text{GT}}(\text{unusual}) = \frac{1+1}{40,000 + 15,000} \cdot \frac{\frac{5500}{40,000}}{\frac{10,000}{40,000}} = \frac{2}{55,000} \cdot \frac{5500}{10,000} = \frac{1}{50,000}.$$

But by using MLE, we would have had a larger value:

$$P_{\mathrm{MLE}}(w) = \frac{1}{40,000}.$$

Because the sum of the probabilities must be 1, we have a remaining probability mass (P_{R}) to be reassigned to the unseen words (U). Consequently, for computing the estimated probability of a single unseen word u_w, we should divide this mass with the estimated number of unseen words $|U|$:

$$P_{\mathrm{GT}}(u\_w) = \frac{P_{\mathrm{R}}}{|U|} = \frac{E(1)}{(|C_{\mathrm{R}}| + |V_{\mathrm{R}}|) \cdot |U|}. \tag{4.7}$$

Going back to Damerau's formula, we now have

$$\mathrm{score}(w) = \frac{f(w, C_{\mathrm{U}})}{|C_{\mathrm{U}}|} \div P_{\mathrm{GT}}(w \ in \ C_{\mathrm{R}}). \tag{4.8}$$

The words having the highest scores are terminological terms. In the case that C_{U} is a large corpus, we can also compute Good–Turing estimators for the numerator. For small corpora, however, this is unreliable, since one cannot compute the estimates $E(n)$ with high enough confidence.

This approach can be improved by additional pre-processing of the corpora involved. First, in order to better capture the real word distribution, it is better to use word lemmas (or stems) instead of word surface forms. Second, the vast majority of the single terminological terms are nouns, and, therefore, one can apply part-of-speech (POS) filtering in order to disregard the other grammatical categories. Both can be resolved by employing stand-alone applications that can POS-tag and lemmatise the considered texts. As the research and development of the author is mainly focused on English and Romanian, he usually makes use of the TTL pre-processing Web Service (Ion 2007; Tufi et al. 2008) when dealing with these languages.

The method presented above can be reinforced with the well-known *TF-IDF* measure (Spärck Jones 1972), provided that the corpus of interest is partitioned into many documents or that this partitioning can be automatically performed. As reference corpora, the author used the *Agenda* corpus (Tufi and Irimia 2006) and a collection of *Wikipedia* documents for Romanian, while for English, *Wikipedia* documents were used.

4.2.3.2 Multi-word Term Extraction

The approach for the identification and extraction of collocations has been described in several publications (Ştefănescu et al. 2006; Todirascu et al. 2009; Ştefănescu 2010). For the purposes of the current task (i.e. multi-word term extraction), we define a collocation as a pair of words, for which

- The distance between the words is relatively constant.
- The words appear together more often than expected by chance: *Log-Likelihood*.

Looking at this definition, one can notice that, from a strict linguistic point of view, such a construction can be seen as a strong co-occurrence rather than a collocation.

The first component of our approach is based on a method developed by Smadja (1993). It uses the average and the standard deviation computed on distances between words to identify pairs of words that regularly appear together at the same distance, a fact that is considered to be the manifestation of a certain relation between those words. Collocations can be found by looking for such pairs for which standard deviation is small.

In order to find terminological expressions, we employ POS filtering, computing the standard deviation for **only** the *noun–noun* and *noun–adjective* pairs within a window of 11 non-functional words, and we keep all the pairs for which standard deviation is smaller than 1.5—a reasonable value according to Manning and Schütze (1999). This method allows us to find good candidates for multi-word expressions, but this is not good enough. We want to further filter out some of the pairs so that we keep only those composed by words that appear together more often than expected by chance. We do this by computing the Log-Likelihood (LL) scores for all of the above-obtained pairs and by taking into account only the occurrences of the words having the selected POS. We take into consideration the pairs for which the LL values are higher than 9, as the probability of error for this threshold is less than 0.004 according to the *chi square* tables.

As terminological expressions, we further keep only those for which at least one of the words composing them can be found among the *single-word* terminological terms, disregarding their context. In this way, we aim at filtering out commonly used expressions that have no terminological value.

4.2.3.3 Experiments and Results

In order to evaluate the term extraction method, we used the EuroVoc thesaurus (Steinberger et al. 2002) as a source of authoritative in-domain English and Romanian terminology. In the experiment, we used 950 English–Romanian parallel documents from the *JRC-Acquis* corpus (Steinberger et al. 2006). They are all from 2006 and contain about 3.5 million tokens per language. To assess the performance of the term extraction method, we generated lists containing only those EuroVoc terms that appeared in these documents for both languages and counted how many of the recognised terms were found in these corresponding restricted lists (Table 4.10).

If a word becomes more and more frequent, approaching its occurrence probability in the reference corpus, the tool cannot consider it terminological. This means that some of the terminology that is valid for the entire JRC-Acquis corpus cannot be

Table 4.10 EuroVoc terms identified as terminological

	English	Romanian
#Documents	950	950
Size of pre-processed	3.55 mil. tokens; 55.1 MB	3.34 mil. tokens; 61.8 MB
EuroVoc terms identified out of those found in the collection having at least 1 occurrence	793/2699 (29.38%)	744/1961 (37.93%)
10 occurrences	289/1185 (24.38%)	252/815 (30.92%)
50 occurrences	65/507 (12.82%)	63/326 (19.32%)
100 occurrences	24/318 (7.54%)	33/213 (15.49%)

discovered by considering only the documents from a single year, even though that terminology appears in those documents.

Regarding this evaluation methodology, one has to keep in mind that the list of EuroVoc terms is neither exhaustive nor definitive and, as such, that there may be valid non-EuroVoc terms that our method discovers. Examples of such extracted English terms include '*Basel convention*', '*standards on aviation*', '*Strasbourg*', '*national safety standards*', '*avian influenza*', etc. This is the reason why the method has not been evaluated in terms of standard precision and recall measures.

4.2.4 Multi-word Term Extraction and Context-Based Mapping for English-Slovene

In this section, we present an approach to automatically extract and map multi-word terms from an English–Slovene comparable health corpus. First, the terms are extracted from the corpus for each language separately using a list of user-adjustable morpho-syntactic patterns and a term-weighting measure (i.e. using linguistically and statistically motivated term extraction methods). Then, the extracted terms are mapped in a bag-of-equivalents fashion with a seed bilingual lexicon. In the extension of the approach, we also show that the small general seed lexicon can be enriched with domain-specific vocabulary by harvesting it directly from the comparable corpus, which significantly improves the results of multi-word term mapping. The section is based on a publication by Ljubešić et al. (2012).

4.2.4.1 Resources and Tools Used

Comparable Corpus

The main source of lexical knowledge in this experiment was the English–Slovene comparable corpus of on-line articles on health and lifestyle, which had already been used successfully in our previous research (Fišer et al. 2011). Health-related documents were extracted from the *ukWaC* (Baroni et al. 2009) and *slWaC* (Ljubešić and Erjavec 2011) web corpora by the criterion that the cosine similarity to a domain model had to be higher than 0.25. The domain model was built using documents from two main health-related internet domains. It is based on content words as features and TF-IDF feature weights, where the IDF weights were calculated on a news-domain corpus.

The subset of the constructed domain corpus that we used in this experiment contains 1.5 million tokens for each language.

Seed Lexicon

The seed lexicon that we used as an anchor between the two languages was constructed using the freely available Slovene–English and English–Slovene *Wiktionaries* that cover mostly general vocabulary. The entries from both *Wiktionaries* were merged, and, if the same pair of words was found in both resources, they were given a higher probability. The seed lexicon contains 6094 entries.

LUIZ

LUIZ is a hybrid bilingual term extractor that uses parallel or comparable corpora as input and outputs mono- and bilingual lists of term candidates (Vintar 2010).

Term extraction is performed on the basis of user-adjustable morpho-syntactic patterns provided for each language. The extracted candidate phrases are assigned a termhood value by comparing the frequency of each word to a reference corpus. Term mapping is performed using the bag-of-equivalents approach (Vintar 2010), which requires a probabilistic bilingual lexicon as input. A list of possible translation candidates for a source multi-word term is proposed by comparing each target term candidate to a bag of potential translation equivalents provided by the lexicon and computing an equivalence score.

ccExtractor

ccExtractor is a context-based bilingual lexicon extraction tool (Ljubešić et al. 2011; Fišer et al. 2011; Ljubešić and Fišer 2011) that allows building context vectors for a list of headwords from each corpus, translating features of context vectors from source language to target language via an existing seed lexicon, and calculating the best translation candidates between headwords in the source language and the target language. In this experiment, the tool is used to enhance the general small seed lexicon used for multi-word term alignment with LUIZ.

4.2.4.2 Experimental Setup

The main task of the experiment was to extract multi-word term translation candidates from the comparable corpus. The experiment was divided in three parts:

- In the first part of the experiment, we used *LUIZ* to extract multi-word term candidates from the monolingual corpora. The result is a list of 25,865 English and 27,102 Slovene multi-word term candidates.
- In the second part of the experiment, we mapped the extracted multi-word term candidates between English and Slovene with *LUIZ* using the seed lexicon.
- In the third part of the experiment, we tried to improve the results by enhancing the seed lexicon used by *LUIZ* with 412 translation equivalents of the domain-specific vocabulary in the corpus that is not covered in the seed lexicon, which we obtained with *ccExtractor*. Term extraction and mapping were then repeated with the same settings, the only difference being the extended seed lexicon. In this step, we combined contextual information obtained from *ccExtractor* with the constituent information provided by *LUIZ*.

Term Extraction

Term extraction in each part of the corpus was performed with the help of a predefined set of morpho-syntactic patterns for each language. These patterns describe part-of-speech sequences of mainly noun phrases up to 5 words in length. Once candidate phrases were extracted from the corpora, a term-weighting measure was used to assign a termhood value to each phrase. This measure computes single-word termhood by comparing the frequency of each word ($f_{n,\,\mathrm{D}}$) to a reference, non-specialised corpus ($f_{n,\,\mathrm{R}}$), and then combines the termhood scores of all constituent words with the frequency (f_a) and length (n) of the entire candidate phrase:

$$W(a) = \frac{f_a^2}{n} * \sum_{1}^{n} \left(\log \frac{f_{n,\mathrm{D}}}{N_{\mathrm{D}}} - \log \frac{f_{n,\mathrm{R}}}{N_{\mathrm{R}}} \right). \tag{4.9}$$

Term Mapping

The extracted multi-word terms were then mapped in the bag-of-equivalents fashion (see section "LUIZ") using the seed bilingual lexicon. For a given source multi-word term, each target term candidate is compared to a bag of potential translation equivalents provided by the lexicon and an equivalence score is computed, thus generating a ranked list of possible translation candidates. If, for example, the bilingual lexicon contains the English–Slovene entries

blood	kri	1.0		
flow	pretok	0.66	tok	0.33,

then the bag-of-equivalents for the English term candidate 'blood flow' will contain all three equivalents, 'kri', 'pretok' and 'tok'. We now compare the Slovene term candidates to the bag and compute the equivalence score as the sum of the translation probabilities found in the target term, normalised by term length. Thus, for the above English term, we extract the following candidates:

pretok krvi	0.83
tok krvi	0.66
šibak tok krvi	0.43.

This approach allows us to identify multiple translation equivalents for a source term, which is especially valuable in domains with less standardised terminology and large term variation. Furthermore, this approach is able to find translation equivalents for the terms for which seed lexicon entries are missing or faulty.

In our current setting, we are able to identify multi-word to multi-word equivalents of different lengths, but we do not identify single-word to multi-word term pairs.

Extension of the Seed Lexicon

In the third part of the experiment, the idea was to extend the mapping of the extracted multi-word terms with the extension of the seed lexicon by adding the most relevant vocabulary from the comparable corpus. Using the *ccExtractor*, we extracted three of the most probable Slovene translations for all English lemmas that were not already included in the initial seed lexicon.

The headwords in both parts of the corpus had to satisfy the minimum frequency constraint of 50 occurrences, which is a reasonable frequency threshold as identified in previous experiments (Ljubešić et al. 2011). When building context vectors, a window of three lemmas on both sides of the headword was used, and the collected features were weighted with the TF-IDF score. Context similarity was calculated with the DICE similarity metric. The probabilities of the translation candidates were calculated as their context similarity weights scaled to a probability distribution.

There were 412 English lemmas not present in the seed lexicon that satisfied the occurrence frequency criterion. Therefore, the extended lexicon contains 6506 entries. This lexicon was used in the third part of the experiment.

4.2.4.3 Evaluation of the Results

In this section, we report the results of manual evaluation of term extraction in both languages as well as the quality of term mapping. We focus here on measuring the precision of term extraction and alignment. While recall would also be interesting to study more closely, we were not able to do it in this experiment, because, in order to measure it, we would need either a comprehensive terminological dictionary of this area for measuring absolute recall or a manually annotated corpus with multi-word terms in both languages for measuring recall relative to the terms used in the corpus.

Evaluation of Term Extraction

In total, 25,865 term candidates were extracted from the English part of the corpus and 27,102 from the Slovene part. The extracted term candidates were assigned a termhood score, and, in order to evaluate the quality of the extracted terms, we manually evaluated the 100 highest-ranked term candidates for each language.

In the evaluation scheme, each candidate was categorised into one of the three possible categories:

- The candidate was a correctly extracted multi-word term from the health domain.
- The candidate was a correctly extracted multi-word term but did not belong to the health domain.
- The candidate was not correctly extracted (a part of a multi-word term), or the multi-word expression was not a term.

The results of manual evaluation are shown in Table 4.11. Among the English candidates, 76 were correctly extracted health terms (e.g. '*blood test*'), 5 were terms

Table 4.11 Evaluation of term extraction (TE) on the 100 highest ranked term candidates

Term quality	English (%)	Slovene (%)
Good term	76	86
Term from a different domain	5	3
Not a term	19	11

but belonged to some other domain (e.g. '*primary school*') and 19 of the candidates were either incorrectly extracted multi-word terms or multi-word expressions that belong to the general vocabulary (e.g. '*next year*'). The results for Slovene are slightly better: 86 of the candidates were correct, 3 were terms from a different domain and 11 were incorrectly extracted multiword terms or other multi-word combinations. The reason for better results in Slovene is probably a cleaner, less noisy corpus, both in terms of domain-specific documents and in terms of corpus annotation, because *slWaC* was built much more conservatively than *ukWaC*.

In both languages, two-word terms are by far the most frequent, with only four English and six Slovene candidates that are longer than two words. This is to be expected, because the longer the term, the less frequent it is in the corpus. However, it must also be noted that the corpus does not contain expert medical texts, but it mostly contains magazine articles about health issues and lifestyle advice for the general public, which have fewer complex medical terms.

Evaluation of Term Mapping

The quality of term mapping was evaluated for each run of the experiment, with the original and the extended seed lexicon, in order to evaluate the impact of the seed lexicon extension. The extension of the seed lexicon evaluated in previous work (Fišer et al. 2011) showed that the top 1 precision is 45% and the top 10 precision is 56%.

Further, we measured the precision of term mapping by manually inspecting the list of the 477 multi-word term pairs that received an equivalence score higher than 0.5 in either run of the experiment. 380 term pairs were identical in both runs, while 97 pairs were different. First, we evaluate the termhood of the source language candidates, and then, in case the candidates are considered a term, we evaluate whether the translation is correct.

The evaluation schema used when evaluating termhood is '*good term*', '*term from a different domain*', '*not a term*', while the evaluation schema used for evaluating the translation quality is '*correct translation*', '*close translation*' and '*incorrect translation*'.

As shown in Table 4.12, source language term candidates that have good probable translation equivalents (equivalence score higher than 0.5) are partial or full terms in 56% of the cases. This is much lower than when evaluating the top ranked term candidates. However, this is because (1) we analysed terms with a high

Table 4.12 Evaluation of term extraction (TE) on the 477 source language term candidates with equivalence score higher than 0.5

Score	Percentage
Good term	43.6
Term from a different domain	12.6
Not a term	43.8

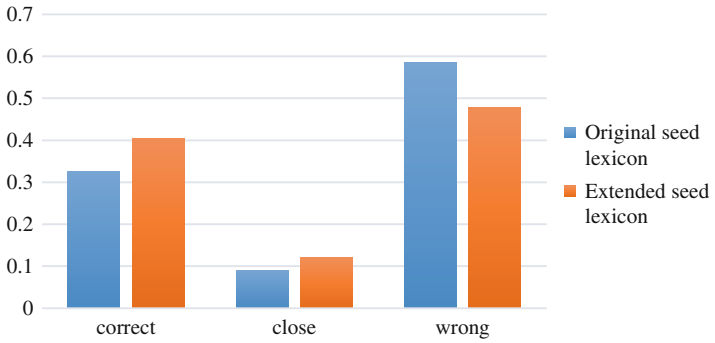


Fig. 4.6 Evaluation of term mapping with the equivalence score higher than 0.5

equivalence score and not necessarily a high termhood score, and (2) term candidates with a high equivalence score consist of constituents found in the general seed lexicon from which terms are rarely built.

The quality of term mapping is shown in Fig. 4.6. When using the original seed lexicon, translations for 41.5% of the terms are correct or close to correct, while, when using the extended seed lexicon, 52.2% of translations are correct or close to correct. It is interesting to note that there is an increase of almost 8% of the correctly mapped terms, while the number of close to correct terms increases by 3%. At the same time, the number of incorrectly mapped terms drops by almost 11%. The result clearly shows that it is very beneficial to add a relevant vocabulary to the seed lexicon, even if the automatically extracted equivalents are noisy.

Another interesting observation is the fact that the pairs that were shared among the two seed lexicons are already of a relatively high quality. Also, the extension of the seed lexicon helped in exactly those cases where the original lexicon was not able to handle well at all, either because it was too small in size or too general for this particular domain. This shows that the already existing resources can easily and successfully be complemented with a simple and fully automatic technique, thereby giving a big boost to the quality of term mapping.

4.2.4.4 Discussion

In this section, we presented an approach to extract translations of multi-word terms from domain-specific comparable corpora. We used *LUIZ*, a hybrid tool for bilingual multi-word term extraction and mapping. Additionally, we used *ccExtractor*, which is a statistical tool for finding translation equivalents for single-word terms in comparable corpora, in order to extend the seed lexicon with the most relevant terms in the corpus, which improved the results of multi-word term mapping by almost 11% for English–Slovene. Additionally, this is the first extrinsic evaluation of context-based single-word lexicon extraction from comparable corpora.

While these results do not outperform the benchmark results achieved by *LUIZ* on parallel data, this is understandable because looking for MWT equivalents in comparable corpora is a much more difficult task. Additionally, the number of resulting MWTs obtained in this experimental setting is not very large; however, their precision is much higher than in the regular SWT extraction and mapping approach. With this in mind, the results that we obtained with the extended seed lexicon are very encouraging and can be beneficial for terminologists who can save time by cleaning the mapped term pairs instead of creating the whole term collections from scratch.

4.3 Named Entity Recognition Using TildeNER

Another type of sub-sentential fragments (phrases) that were analysed in the ACCURAT project were named entities (NE). Translations of NEs in different languages can be frequently found in news articles written in different languages on similar topics, encyclopaedic articles (e.g. inter-language linked Wikipedia articles) and other sources of potential comparable corpora. The ability to identify named entities is, therefore, an essential pre-requisite in the process of building NE dictionaries from comparable corpora.

Named entity recognition (NER) has been actively researched for over 25 years. However, most of the research has focussed on resource-rich languages: for instance English, French and Spanish. Therefore, during the ACCURAT project, an open source, freely available toolkit (named *TildeNER*) was developed that allows bootstrapping named entity recognition systems for languages with limited annotated data. The toolkit makes use of existing supervised learning methodology [e.g. the Stanford NER conditional random field (CRF) classifier (Finkel et al. 2005)] enriched with heuristic refinement methods in order to bootstrap NER models using unlabelled data, thus creating a '*highly supervised*' semi-supervised named entity recogniser. *TildeNER* and its evaluation for two under-resourced languages—Latvian and Lithuanian—will be described in the following subsections.

The current dominant approach to developing named entity recognition systems is supervised learning (Nadeau and Sekine 2007). This, however, means that a prerequisite for NER model training is a large named entity (NE) annotated data corpus. This is not an issue for resource-rich languages, but it is for under-resourced languages (e.g. the Baltic languages). For Latvian and Lithuanian, there has been very little previous research in the field of named entity recognition. Most of the existing research has dealt with only toponym recognition: for instance Skadiņa (2009) describes toponym recognition from image annotations using lexicons and patterns. Also, the lack of annotated named entity corpora for both languages does not allow (without significant financial input for corpora creation) the development of a truly supervised NER system. Because of the available resource constraints for Latvian and Lithuanian, a semi-supervised NER system development approach was selected, more precisely, bootstrapping.

The next section provides a description of the seed NE-annotated corpora followed by a section describing the design and methods applied in *TildeNER* and evaluation in Sect. 4.3.4. This section is based on a publication by Pinnis (2012).

4.3.1 Annotated Corpora

For the task of named entity recognition, relatively small NE annotated corpora were created. The corpora for both languages consist of IT localisation (software reviews, manuals and other IT-related articles), news (current news from news web portals) and Wikipedia articles in equal proportions. The first two parts were acquired using comparable corpora web crawling tools (the Focussed Monolingual Crawler and the RSS News Collecting tool) described in the previous chapter. The corpora statistics are shown in Table 4.13.

For the annotation task, NE mark-up guidelines[1] were prepared. The guidelines are mostly compliant with the MUC-7 (Chinchor 1997) NE annotation guidelines (with minor adaptation for Latvian and Lithuanian). The following NE categories were annotated: *organisation*, *person name*, *location*, *product*, *date*, *time* and *money*.

The corpora were annotated by two annotators and disagreements were resolved by a third annotator. The inter-annotator agreement between the first two annotators using the Cohen's kappa statistic (Cohen 1968) is 0.885 for Latvian and 0.822 for Lithuanian. This score represents the overall complexity of the corpora including non-entities strictly classified as non-entities by both annotators. Therefore, separate NE category and NE border detection inter-annotator agreement scores are given in Table 4.14.

Table 4.13 Latvian and Lithuanian corpora statistics

	Latvian	Lithuanian
Document count		
Seed	40	37
Development	25	33
Test	66	55
Total	131	125
Word count		
Seed	20,959	18,852
Development	10,053	17,827
Test	41,208	36,239
Total	72,220	72,918

[1]Published as part of TildeNER in the 'Toolkit for multi-level alignment and information extraction from comparable corpora', public deliverable of the project ACCURAT, 2011.

Table 4.14 Inter-annotator agreement on Latvian and Lithuanian corpora

	Latvian	Lithuanian
Full NE agreement		
NE border agreement	0.749	0.671
Category agreement on matching borders	0.964	0.967
Token level agreement		
LOCATION	0.790	0.703
ORGANISATION	0.708	0.623
PERSON	0.932	0.910
PRODUCT	0.641	0.683
DATE	0.812	0.696
TIME	0.713	0.662
MONEY	0.785	0.599
Total token agreement	0.807	0.723

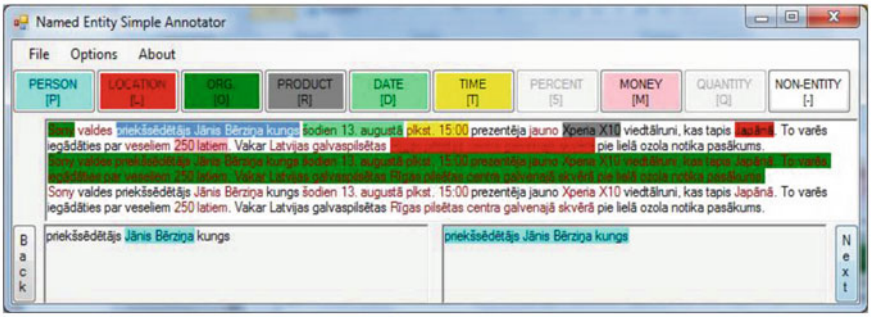


Fig. 4.7 Disambiguation view of *NESimpleAnnotator*

In the process of annotation, a tool named *NESimpleAnnotator* was used (available as a part of the ACCURAT Toolkit). The annotation tool allows fast one-dimensional (non-hierarchical) annotation of NEs of the defined categories. The annotation tool also features disambiguation functionality for a judge. The annotation tool in the disambiguation view is depicted in Fig. 4.7.

After annotation, both corpora were split in seed, development, and test sets. The NE statistics in the disambiguated corpora are shown in Table 4.15.

The NE annotated data is stored in plaintext format containing MUC-7 style NE tags. An example of the format is given in Fig. 4.8.

Table 4.15 Latvian and Lithuanian NE annotated corpora statistics

	Latvian			Lithuanian		
NE type	Seed	Development	Test	Seed	Development	Test
DATE	498	249	843	548	297	711
LOCATION	682	479	1453	470	563	1086
MONEY	123	18	148	150	147	313
ORGANISATION	464	219	966	240	275	603
PERSON	267	172	601	202	169	604
PRODUCT	381	103	382	174	310	389
TIME	200	46	107	67	57	109
Total	2615	1286	4500	1851	1818	3815

<ENAMEX TYPE="PERSON">Bruno Kalniņš**</ENAMEX>** dzimis **<TIMEX TYPE="DATE">**1899. gada 7. maijā**</TIMEX>** **<ENAMEX TYPE="LOCATION">**Tukumā**</ENAMEX>** ievērojamo sociāldemokrātu **<ENAMEX TYPE="PERSON">**Paula Kalniņa**</ENAMEX>** un **<ENAMEX TYPE="PERSON">**Klāras Kalniņas**</ENAMEX>** ģimenē.

Fig. 4.8 Sample of Latvian human annotated NE corpora using *NESimpleAnnotator*

4.3.2 System Design

TildeNER is a named entity recognition toolkit that consists of multiple workflows for NER model training, NE tagging, and evaluation.[2] For training and tagging as a machine learning (ML) component, *TildeNER* uses the conditional random field classifier *StanfordNER* (Finkel et al. 2005), which contains a large set of feature functions required in a supervised NER system (and does not require inventing the wheel a second time).

4.3.2.1 Feature Function Selection

The feature functions for both Latvian and Lithuanian were selected using *iterative minimum error-rate training*. The method starts with a seed feature function set and, in each iteration, trains multiple (depending on the number of altering feature functions) NER models with altered (set to '*true*' or '*false*' or assigned a different value) feature functions, where each model has a different feature function altered. The feature function set of the model that increases the *F*-measure the most is selected as the base set for the next iteration.

Although such an iterative approach allows finding only a local maximum, it is sufficient for selecting good performance feature functions. In every iteration,

[2]A detailed list of available workflows is listed in the deliverable D2.6 of the ACCURAT project.

85 different models were trained, and the performance on Latvian development data increased from a token level *F*-measure of 63.29 to 69.47, which gives a significant increase of the system's performance (though on development data).

4.3.2.2 Data Pre-processing

The human annotated data and unlabelled data that are used in NER model training or tagging are pre-processed using morpho-syntactic taggers. For Latvian and Lithuanian, we used the maximum entropy-based tagger by Pinnis and Goba (2011).

After morpho-syntactic tagging, positional information is added in order to trace every token from the tab-separated document back to its positions in the plaintext input document. In the case of gold annotated data, NE categories are also assigned to each token. As introduced in the CoNLL 2002 conference (Tjong and Sang 2002), we use the BIO scheme for annotation of non-entity tokens and NE tokens (e.g. '*B-ORG*' and '*I-ORG*' for first and further tokens of an organisation).

The data pre-processing step allows introducing a new feature for the NER model training—the morpho-syntactic tag. This feature has been integrated in the *StanfordNER* conditional random field (CRF) classifier used by *TildeNER*. It can be used as an additional feature to describe the context around a token in the range from 1 to N tokens to the left and to the right from each token.

In *TildeNER*, a new language can be integrated by providing a morpho-syntactic tagger that tokenises and tags data in a tab-separated format as defined in Fig. 4.9. However, the morpho-syntactic tag is optional and, for morphologically simpler languages, it can also be omitted by changing the NER model training and NE tagging property files required by the Stanford NER CRF classifier.

4.3.2.3 NER Model Bootstrapping

The NE annotated corpora for Latvian and Lithuanian are relatively small compared to data sets that are used, for instance, for English NER system development and model training. Therefore, *TildeNER* features an NER model bootstrapping module,

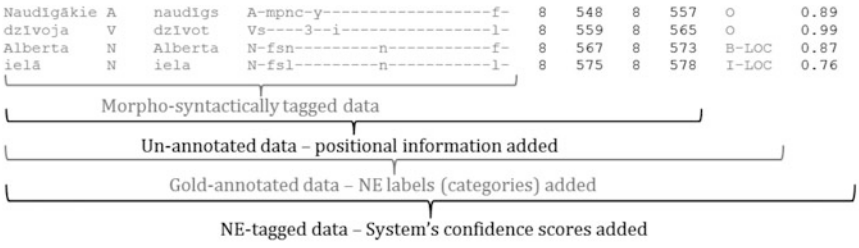


Fig. 4.9 Pre-processed data format sample of different intermediate output files within *TildeNER* workflows

which uses a bootstrapping method similar to that proposed by Liao and Veeramachaneni (2009).

In order to bootstrap an NER model, the system requires seed, development, and test data (human annotated data). In addition to the human annotated data, unlabelled data is required (articles from Wikipedia and Web news were used in our experiments). Once all data is available, the bootstrapping system iteratively

- **Trains an NER model**. In the first iteration, only seed data is used as training data. In further iterations, in addition to the seed data, new training data, which is extracted in previous iterations, is used.
- **Evaluates the trained model on development and test data**. *TildeNER* allows enforcing only positive iteration usage, dropping all iterations that decrease performance on the development set. An iteration can be considered positive if it increases either precision, recall, or *F*-measure (depending on the configuration).
- **Tags the unlabelled data with the newly trained NER model**. In the case that the configuration requires only positive iteration data propagation and the current trained model decreases performance, unlabelled data is tagged with a model from the last positive iteration.
- **Extracts new training data**. Sentences from the unlabelled data that contain NEs, which have been annotated with the heuristic and statistical refinement methods, are ranked, and the top *N* sentences of each NE category are selected for inclusion in the training data. It is important in this step to use good refinement methods that are able to tag new and unseen by the supervised classifier NEs. If the raw data that the NER classifier outputs is used, the bootstrapping learns only the cases that it already knows, as the supervised classifier's performance on unseen data is unreliable.
- **Extracts new gazetteer data from the newly tagged unlabelled data**. This step is optional but can be used in automatic gazetteer bootstrapping. *TildeNER* allows using bootstrapped NE lists in training of further iteration NER models.

4.3.2.4 Refinement Methods

In NER model bootstrapping and tagging, *TildeNER* applies refinement methods in order to improve upon the NE classification results produced by the *StanfordNER* CRF classifier. During bootstrapping, the refinements help in identifying NEs in new unseen contexts. When tagging documents, refinements allow achieving better precision and/or recall.

Refinement methods are functions that analyse a document and re-classify tokens or sequences of tokens as named entities or non-entities. The following refinements have been implemented in *TildeNER*:

- **Removal of unlikely NEs**. Named entities that are classified by the CRF classifier below a threshold are re-classified as non-entities (*increases precision*).

- **Consolidation of equal lemma sequences**. In NER, a common assumption is to classify equal NEs with the same category (one sense per discourse rule). This method analyses such cases and decides whether for certain NEs, which are classified as belonging to multiple categories, one main category can be identified. Misclassified entities in such situations are re-classified (*increases precision*).
- **Enforcing equal lemma sequences to be tagged** (*increases recall*). Similarly as in the previous method, the CRF classifier tends not only to misclassify, but also to miss some NEs (mostly in contexts unknown to the NER model). This method classifies lemma sequences that are misclassified as non-entities if there exists an NE that is classified with a confidence score of over a configurable threshold and has the same lemma sequence as the non-entity sequence. This refinement method also enforces the one sense per discourse rule.
- **NE border correction** for entities that contain an odd number of quotation marks or brackets (*increases both precision and recall*). When bootstrapping, the new training data tends to contain classified sequences that lack, for instance, a bracket or a quotation mark, because the classifier's confidence has been too low to tag the misclassified token as part of the NE. This method tries to expand or reduce the NEs containing bracketing and quotation mistakes.
- **Artefact removal methods** (*increases precision*). When applying the NER system to different domain texts, different in-domain artefacts (e.g. hyperlinks in web crawled documents, corrupt mark-up from corpora processing, etc.) can be present in texts.
- **Person name analysis** (*increases recall*). As person names may consist of multiple tokens (first name, middle name, last name, title, etc.), the refinement method splits all person NEs that have a CRF classifier's confidence score above a defined threshold into separate tokens and tags missed non-entity tokens.
- **Validation of NEs that start sentences**(*increases recall*). Sentence beginnings have proven to be difficult cases for NER, because the capitalised tokens may be misleading. If the CRF classifier classifies a token as an NE, but it can be found elsewhere in the same document as a common word (i.e. in a lowercased form), then the NE is re-classified as a non-entity.

Refinement methods can be applied in any required sequence by passing a '*refinement order definition string*' when running *TildeNER*. This allows boosting the system's performance by either recall or precision (and in some cases by both).

4.3.3 Evaluation

4.3.3.1 Non-comparative Evaluation

As a baseline, we use the supervised system (without bootstrapping and refinements) trained with only the *StanfordNER* CRF classifier using the feature functions selected in the iterative minimum error rate training. Table 4.16 shows the baseline

Table 4.16 Evaluation results on test data

System	Latvian				Lithuanian			
	P	*R*	*A*	*F*1	*P*	*R*	*A*	*F*1
Baseline (Only CRF Classifier)								
Token	74.70	56.23	91.02	64.16	74.44	63.54	92.30	68.56
Full NE	62.43	48.01	–	54.28	67.42	58.60	–	62.70
Baseline (CRF + refinement methods) tuned for precision								
Token	86.47	41.51	88.86	56.09	84.04	53.74	91.53	65.56
Full NE	75.61	35.05	–	47.90	77.01	49.63	–	60.36
Baseline (CRF + refinement methods) tuned for F-measure								
Token	74.63	57.15	91.17	64.73	76.31	63.50	92.47	69.32
Full NE	62.32	49.66	–	55.27	68.57	59.39	–	63.65
Bootstrapped (CRF + refinement methods + bootstrapping) for better precision								
Token	87.27	45.17	89.57	59.53	–	–	–	–
Full NE	79.18	41.85	–	54.76	–	–	–	–
Bootstrapped (CRF + refinement methods + bootstrapping) for better F-measure								
Token	75.55	61.34	91.86	67.71	76.90	63.77	92.42	69.72
Full NE	64.98	56.06	–	60.19	71.32	59.91	–	65.12

performance with an *F*-measure of 54.28 for Latvian and 62.70 for Lithuanian on full NEs (i.e. full phrases).

An obvious question is: '*Why is there such a huge difference*?' The answer is quite simple—the test sets and training sets vary in content complexity. For instance, the Latvian texts feature automatically web crawled data, which also includes extracted tables with vague structure (space or tab separated), many short fragments with missing contexts and fragments with comma separated NEs.

The Lithuanian corpora, on the other hand, is manually selected and extracted from news portals, Wikipedia and other sources. Therefore, it consists of less complex structures. All these points result in lower Latvian results on the test set, and, if comparison between the two system evaluations is performed, test data complexity has to be taken into account.

Once the baseline systems were prepared, the refinement method parameters and the refinement method sequences were tuned using the development sets. As a result, two *refinement order definition* configurations were acquired:

- A configuration that allows increasing precision by up to 10% and more (at the cost of recall) with the following *refinement order definition string*:
  '*L N S F T_0.8 C P_0.8 R_0.8*'. The string states that after CRF classification, the following refinements are applied to the raw classified data in this exact sequence:
  - NE border correction ('*L*')
  - Artefact removal methods ('*N*' and '*S*')
  - Validation of NEs that start sentences ('*F*')
  - Tagging of equal lemma sequences with a threshold of 0.8 ('*T_0.8*')

  - Consolidation of equal lemma sequences (‘*C*’)
  - Person name analysis with a threshold of 0.8 (‘*P_0.8*’)
  - Removal of unlikely NEs with a threshold of 0.8 (‘*R_0.8*’)
- A configuration that allows increasing *F*-measure (though only up to 1%) with the following refinement order definition string: ‘*L N S F C T_0.6 P_0.5*’.

The evaluation results using refinement methods on top of the baseline CRF based system are given in Table 4.16. Using bootstrapped models (with the respective refinement configurations), precision and *F*-measure can be increased by up to 4.92% over the refined supervised results for full NEs and up to 16.55% for precision and up to 5.91% for *F*-measure over the baseline systems. For comparison, in their work for Czech, which is also a morphologically rich language with different NE capitalisation rules as in English, Kravalová and Žabokrtský (2009) achieve an *F*-measure of 71 using 10 NE categories and a corpus twice as large.

In the precision bootstrapped NER model for Latvian, a total of 75% of errors are caused by missing NEs in the tagged data, 15% are caused by incorrect border detection and the remaining 10% are wrong category classification mistakes.

4.3.3.2 Experimental Comparative Evaluation

In order to better understand the performance figures and to be able to better compare results of TildeNER to different language NER systems, for experimental purposes, we performed a comparative evaluation task on English–Latvian parallel and strongly comparable corpora. As the NE coverage and the document structural complexity is the same (or at least very close) in documents of both languages, we can perform cross-lingual analysis of NER system performance.

As *TildeNER* relies on the *StanfordNER* CRF classifier, a Stanford NER model[3] that achieves an *F*-measure of 93.0 for English on the ‘*CoNLL 2003 testa*’ data set[4] was selected for comparative evaluation.

For the comparative evaluation, a set of ten documents (five parallel and five strongly comparable) was selected. The comparable documents are Wikipedia articles and European Commission bilingual news articles, while the parallel documents are legal documents. NEs in both languages were annotated by a human annotator in order to create a reference (gold) data set for evaluation. The corpora statistics are shown in Table 4.17.

The NE types were limited to organisation, person and location. The evaluation results are shown in Table 4.18.

[3] *StanfordNER* English model from the University of Stanford: ‘*conll.distsim.iob2.crf.ser.gz*’, available for download from: http://nlp.stanford.edu/software/crf-faq.shtml (point 11).

[4] As reported by the University of Stanford in: http://nlp.stanford.edu/software/crf-faq.shtml (point11).

Table 4.17 Comparative evaluation corpora statistics for English–Latvian

NE type	English	Latvian
Organisation	441	404
Location	291	329
Person	113	148
Total	845	881

Table 4.18 English-Latvian comparative evaluation results

	Precision	Recall	*F*-measure
StanfordNER			
Location	37.5	31.91	34.48
Person	37.12	45.37	40.83
Organisation	60.89	70.89	65.51
Latvian bootstrapped for better precision			
Location	76.47	39.63	52.21
Person	76.27	30.41	43.48
Organisation	93.16	44.14	59.90
Latvian bootstrapped for better F-measure			
Location	63.85	50.61	56.46
Person	54.08	71.62	61.63
Organisation	77.82	56.86	65.71

The comparative evaluation results suggest that even if the results of *TildeNER* are lower than state-of-the-art English NER system results, those cannot be compared without taking test set characteristics into account. The results also suggest that *TildeNER* for Latvian performs slightly better for location and person name NEs on the 10 document comparative evaluation scenario.

One important note when analysing the results has to be also taken into account—the test set of the comparative evaluation is more in favour of the *TildeNER* Latvian NER system, as that has been trained on a mixed set of documents that also include Wikipedia articles, which are out of domain articles for the *StanfordNER* English model. Nevertheless, the methodology of bilingual comparative evaluation is a means to compare NER systems for different languages.

4.3.4 Discussion

In this section, we presented *TildeNER*—an NER system developed for two Baltic languages, which had not previously had supervised and semi-supervised ML methods applied for NER. Although the results show improvements in *F*-measure using raw data refinement methods as well as *F*-measure targeted bootstrapping, the methods have to be improved in order to achieve a significant increase over the supervised learning models.

Good refinement methods are the most important factor for a successful NER model bootstrapping system that is based on supervised learning-based classification. It is essential that the refinement methods are able to find new and unseen data.

The *TildeNER* toolkit offers large configuration possibilities for various NER tasks (aid in question answering, automatic gazetteer extraction, machine translation, keyword extraction, etc.) where different requirements for higher precision or higher *F*-measure can be set.

4.4 Lexica Extraction

Bilingual lexica are a key component of all cross-lingual NLP applications, and their compilation remains a major bottleneck in computational linguistics. Automatic extraction of translation equivalents from parallel texts has been shown to be extremely successful (e.g. Och and Ney 2000; Tiedemann 2005). However, such a scenario is not feasible for all language pairs or domains, because ready-made parallel corpora do not exist for many of them, and compilation of such corpora is slow and expensive. This is why an alternative approach that relies on texts in two languages, which are not parallel but nevertheless share several parameters, such as topic, time of publication and communicative goal (Fung 1998; Rapp 1999), has been increasingly explored in the past decade. Compilation of such comparable corpora is much easier, especially since the availability of rich web data (Xiao and McEnery 2006).

This section is based on publications by Ljubešić and Fišer (2011) and Fišer and Ljubešic (2011). It presents methods for bilingual lexica extraction from comparable corpora that were explored in the ACCURAT project.

4.4.1 Related Work

The seminal papers in bilingual lexicon constructions are Fung (1998) and Rapp (1999), who showed that texts do not need to be parallel in order to extract translation equivalents from them. Instead, their main assumption, central to distributional semantics, is that the term and its translation appear in similar contexts. Therefore, the task of finding the appropriate translation equivalent of a term is reduced to finding the word in the target language whose context vector is most similar to the source term's context vector based on their occurrence in a comparable corpus. This is basically a three-step procedure:

- **Building context vectors**. When representing a word's context, some approaches look at a simple co-occurrence window of a certain size, while others include some syntactic information as well. For example Otero (2007) proposes using bilingual correspondences between lexico-syntactic templates, while Yu and Tsujii (2009) use dependency parsers, and Marsi and Krahmer (2010) match

syntactic trees. Instead of context windows, Shao and Ng (2004) use language models. Next, words in co-occurrence vectors can be represented as binary features, by term frequency, or weighted by different association measures, such as TF-IDF (Fung 1998), PMI (Shezaf and Rappoport 2010), or, one of the most popular, the log likelihood score. Approaches also exist that weigh co-occurrence terms differently if they appear closer to or further from the nucleus word in the context (e.g. Saralegi et al. 2008).

- **Translating context vectors**. Finding the most similar context vectors in the source and target language is not straightforward, because a direct comparison of vectors in two different languages is not possible. This is why most researchers first translate features of source context vectors with machine-readable dictionaries, mostly called seed lexicons, and compute similarity measures on those translated vectors. Koehn and Knight (2002) construct the seed dictionary automatically based on identically spelled words in the two languages. Similarly, cognate detection is used by Saralegi et al. (2008) by computing the longest common subsequence ratio. Déjean et al. (2005), on the other hand, use a bilingual thesaurus instead of a bilingual dictionary.
- **Selecting translation candidates**. After source context vectors have been translated, they are ready to be compared to the target context vectors. A number of different vector similarity measures have been investigated. Rapp (1999) applies the city-block metric, whilst Fung (1998) works with cosine similarity. Recent work often uses the Jaccard index or DICE coefficient (Saralegi et al. 2008). Additionally, some approaches include a subsequent re-ranking of translation candidates based on cognate detection (e.g. Shao and Ng 2004).

All the described approaches disregard the central problem of language—polysemy. Recently, there have been approaches that include semantic disambiguation in the overall process, both at the level of disambiguating the context (Apidianaki et al. 2013) by performing WSD on context vector features and on the level of disambiguating the headwords (Fišer et al. 2012) by building context vectors in the source language only from contexts that were previously, by means of WSD, identified as belonging to a specific sense of the headword.

4.4.2 Experiments on Bilingual Lexicon Extraction

4.4.2.1 Experimenting with Key Parameters

The key parameters that can be taken into account in the process of extracting lexica from comparable corpora are the following:

- Corpus size
- Corpus comparability
- Size and type of the seed lexicon
- Method of building a context vector

  - The context window used
  - The weighting function for context words
- The similarity measure used to compare context vectors from L1 and L2

We inspect these key parameters on a task of extracting English–Slovene translation candidates from the JRC-Acquis parallel corpus (Steinberger et al. 2006), which was previously tagged and lemmatised. The parallelism of the corpus is not exploited directly at any point but is used to control the comparability level of the generated sub-corpora.

4.4.2.2 Corpus Size and Comparability

To observe the impact of corpus size and level of comparability, we produce two sets of comparable sub-corpora: one with high and one with low comparability. We call the corpora of high comparability '*easy*' and the corpora with lower comparability '*hard*'. We produce '*easy*' and '*hard*' corpora of different sizes: from 1.6 million to 8 million tokens, extending the size of the corpus on each level with an additional 1.6 million tokens, thereby producing five levels of corpus size for each corpus comparability level.

We produce these sub-corpora by slicing both sides of the initial corpus in 10 equally sized slices in chronological order, so that the first slice contains the oldest texts in the corpus and the last slice the most recent ones. By combining more distant text slices, we produce sub-corpora of lower comparability. We compute the comparability level via the Spearman rank correlation coefficient (Kilgarriff 2001), which compares the ranks of the N most frequent words in each corpus. The size, structure and resulting comparability score of the ten sub-corpora are described in Table 4.19.

Table 4.19 Description of the ten sub-corpora used in the experiments

Size	Slovenian slices	English slices	ρ
High comparability ('easy1–5' corpora)			
1.6	s3	s4	0.92
3.2	s1+s3	s2+s4	0.93
4.8	s1+s3+s5	s2+s4+s6	0.95
6.4	s1+s3+s5+s7	s2+s4+s6+s8	0.95
8	s1+s3+s5+s7+s9	s2+s4+s6+s8+s10	0.96
Low comparability ('hard1–5' corpora)			
1.6	s2	s9	0.50
3.2	s1+s2	s9+s10	0.52
4.8	s1+s2+s3	s8+s9+s10	0.59
6.4	s1+s2+s3+s4	s7+s8+s9+s10	0.66
8	s1+s2+s3+s4+s5	s6+s7+s8+s9+s10	0.74

It is important to observe that at no point were parallel sections of the text included in any corpus and, thereby, the comparability criterion is met in each of the ten corpora.

4.4.2.3 Seed Lexicons

To inspect the impact of quality and size of the seed lexicon on the task, we experiment with three different lexicons: a general large-sized bilingual dictionary (*Grad*) consisting of 42,700 entries, a medium-sized Wiktionary that covers basic vocabulary (*Wiki*) containing 6600 entries and a small domain-specific lexicon that was extracted from a word-aligned parallel corpus from the same domain (*Acquis*) containing 2800 entries.

To assess the lexicon quality given the domain of the text, we compute the coverage of the corpus by each lexicon. The results are shown in Table 4.20. We can nicely observe how the largest Grad lexicon and the smallest Acquis lexicon have similar coverage on the token level regardless of the huge difference in the size of the lexicons. The Wiki lexicon, despite being more than two times the size of the Acquis lexicon, has half the token coverage of the Acquis lexicon. That size matters is shown when the Grad and Acquis lexicons are compared at the type level, Grad having multiple times better coverage, pointing towards a much wider coverage of lower-frequency general-domain terms.

4.4.2.4 Vector Building and Comparison

Besides experimenting with corpus size and comparability on one side and seed lexicon size and quality on the other, we experiment with different methods of

- Building context vectors
  - Whether we encode the position of the content words or disregard it?
  - What size of the context window to use?
  - Which feature weighting function to use (relative frequency, pointwise mutual information (PMI), TF-IDF or log-likelihood (LL)?
- Metrics for comparing context vectors across languages. We experiment with the Manhattan distance, Jaccard and Dice indices adapted to non-binary values (Grefenstette 1994), the Tanimoto index, cosine similarity and Jensen–Shannon divergence.

Table 4.20 A comparison of vocabulary coverage between the three dictionaries and the JRC-Acquis corpus

	Types (%)	Tokens (%)
Grad	13.82	81.73
Wiki	3.86	41.07
Acquis	3.14	78.35

4.4.2.5 Evaluation of Results

Since the result of our approach to lexicon extraction from comparable corpora is a ranked result (for each lexeme in L1, lexemes from L2 are ranked by their context similarity to the L1 lexeme), we use the mean reciprocal rank (MRR)[5] (Voorhees 2001) as our evaluation metric. We evaluate the result of a specific setting by comparing the automatic ranked result to a manually validated lexicon of 1000 entries, which is not present in any of the seed lexicons.

The first set of experiments targets the best set of features for representing the context vector (i.e. whether to encode the position and what window size to use) and which seed lexicon out of the three lexicon candidates to use. These experiments are run on the largest corpus with lower comparability, called '*hard5*'.

The best-performing features for building context vectors turned out to be the window size of 7 with encoded position of context words. The best-performing seed dictionary for translating vectors was the Acquis dictionary, which was obtained from a small domain-specific word-aligned parallel corpus.

The second set of experiments focusses on the feature weighting function and the context vector similarity function. The results presented in Table 4.21 show that best results are obtained with TF-IDF and log-likelihood weighting. Good similarity measures are the Jaccard index and the Jensen–Shannon divergence.

Finally, we focus on the impact of the corpus quality and corpus size. We depict the relationship between the corpus size and the MRR result on two levels of corpus comparability in Fig. 4.10. We can observe that the level of comparability of the corpora plays a major role in the quality of the extracted translation lexicon, especially when very little data is used. However, the size of the corpus (on the level of having from 1.6 to 8 million words on each side) is only significant with less comparable corpora.

Table 4.21 Evaluation of the results for different feature weighting functions and similarity measures on the *hard5* sub-corpus

	relfreq	pmi	tfidf	ll
manh	0.07	0.11	0.15	0.04
jacc	0.70	0.62	0.74	0.74
tanim	0.57	0.49	0.60	0.43
cos	0.60	0.46	0.61	0.44
jenshan	0.68	0.51	0.69	0.78

[5]If many L2 candidates were correct translations of the L1 lexeme, it would be more reasonable to use mean average precision (MAP).

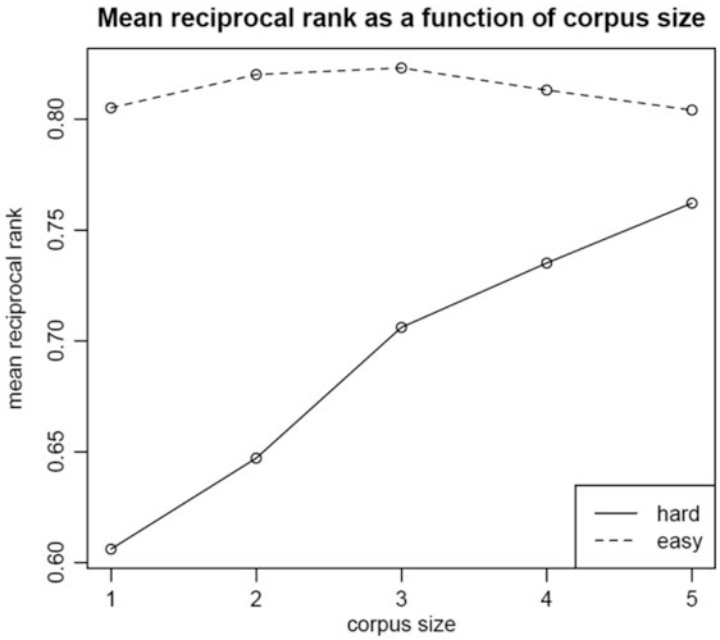


Fig. 4.10 The relationship between the corpus size and the MRR result

4.4.3 Bilingual Lexicon Extraction for Closely Related Languages

The approach of extracting bilingual lexica from comparable corpora relies on the assumption that the term and its translation appear in similar contexts. However, a direct comparison of vectors in two different languages is not possible, which is why a dictionary is needed to map the features of source context vectors into the feature space of the target language so that vector similarity can be computed. At this point, we seem to be caught in a vicious cycle: the very reason why we are resorting to a complex comparable corpus approach for mining translation equivalents is the fact that we do not have a bilingual dictionary to use in the first place. For closely related languages, approaches have been suggested that break the mentioned vicious circle by exploiting the lexical similarity of those languages and therefore not requiring a seed lexicon.

We showcase here an approach to building a Slovene–Croatian lexicon without using any seed lexicon. Slovene and Croatian are both South Slavic languages, and their closeness is considered to be similar to the one of Czech and Slovak or Spanish and Portuguese.

4.4.3.1 Building the Comparable Corpus

We built a Croatian–Slovene comparable news corpus from the 1.2 billion-word *hrWaC v1.0* and the 380 million-word *slWaC v1.0*, which were constructed from the web by crawling the *.hr* and *.si* top-level domains (Ljubešić and Erjavec 2011). We extracted all documents from the domains *jutranji.hr* and *delo.si*, which are on-line editions of national daily newspapers with a high circulation and a similar target audience. The documents were already tokenised, PoS-tagged and lemmatised, resulting in 13.4 million tokens for Croatian and 15.8 million tokens for Slovene.

4.4.3.2 Building the Seed Lexicon

The baseline experiment on extracting translation equivalents relied only on lexical overlap (using identically spelled forms in both languages). We performed manual evaluation of 100 random entries of such an induced seed lexicon and show the results in Table 4.22.

By relying only on identically spelled words, we built a 33k-entry lexicon of fair quality, where each fourth entry is wrong.

4.4.3.3 Extending the Seed Lexicon with Cognates

We continued extending this automatically built seed lexicon by including cognates. We calculate '*cognateness*' with BI-SIM, the longest common subsequence of bigrams with a space prefix added to the beginning of each word in order to punish the differences at the beginning of the words (Kondrak and Dorr 2004). The threshold for cognates has been empirically set to 0.7. In this step, translation equivalents were calculated by using best performing settings from previous experiments for all content words (nouns, adjectives, verbs and adverbs), taking into account 20 top-ranking translations and analysing them for potential cognates in the given order. If we found a translation equivalent that met the cognate threshold of 0.7, we added that pair to the lexicon and continued to the next L1 lexeme. If the seed lexicon already contained a translation for a cognate that we identified with this procedure, we replaced the existing lexicon entry with the new identified cognate pair. Replacing entries is a decision based on empirical results.

Table 4.22 Size and the precision of the seed dictionary

PoS	Size	Precision (%)
Nouns	25,703	88
Adjectives	4042	76
Verbs	3315	69
Adverbs	435	54
Total	33,495	72

Table 4.23 Manual evaluation of contextually proven cognates

PoS	Size	Correct (%)
Nouns	1560	84
Adjectives	779	92
Verbs	706	74
Adverbs	114	85
Total	3159	84

Table 4.24 Automatic evaluation of translation extraction with a seed lexicon including cognates

Lexicon	*N*	*A*	*V*	All
Baseline	0.605	0.566	0.579	0.592
Cognates-*N*	0.657	0.578	0.596	0.630
Cognates-*A*	0.669	0.567	0.590	0.634
Cognates-*V*	0.630	0.497	0.555	0.589
Cognates-*R*	0.604	0.573	0.608	0.598
Cognates-all	0.708	0.534	0.604	0.653

The size and quality of this extension of the seed lexicon is shown in Table 4.23. Although the number of entries added to the seed lexicon with this method is 10 times smaller than when relying on identically spelled words, the interesting property of these entries is that they are overall of more than 10 percent higher accuracy.

Table 4.24 contains the results of an automatic evaluation of bilingual lexicon extraction with mean-reciprocal rank (MRR) on a gold standard. The baseline lexicon refers to the lexicon of identically spelled words, while the remaining lexicons are extensions of the baseline lexicon with cognates of a specific part-of-speech. Noun and adjective cognates contribute to the task the most, although the amount of adjectives added to the lexicon is half the size of nouns. Adding cognates of all parts of speech, the baseline lexicon gives the largest improvement with an increase of 6.1 points in MRR.

4.4.3.4 Extending the Seed Lexicon with First Translations

Previous research has shown that precision of the first translation candidates of highly frequent words in the corpus is especially high (Fišer et al. 2011). Therefore, we decided to also include them to the seed lexicon and inspect their impact on the task of bilingual lexicon extraction. We only took into account the first translation candidates for words that appear in the corpus at least 200 times. If the seed lexicon already contained an entry that we were able to translate with this procedure, we again replaced the old pair with the new one. We show the number and quality of entries obtained by this method in Table 4.25.

Overall, first translation candidates yielded 1635 more entries for the seed lexicon than cognates, but their quality is much lower (by 22% on average). More than 52% of the extracted first translation candidates are nouns, which are also of the highest

Table 4.25 Manual evaluation of first translations of the most frequent words

PoS	Size	Correct (%)
Nouns	2510	71
Adjectives	957	57
Verbs	1002	63
Adverbs	325	59
Total	4794	62

Table 4.26 Automatic evaluation of translation extraction with a seed lexicon including first translations

Lexicon	*N*	*A*	*V*	All
Baseline	0.605	0.566	0.579	0.592
First-N	0.665	0.665	0.626	0.659
First-A	0.700	0.581	0.589	0.656
First-V	0.643	0.513	0.546	0.599
First-R	0.610	0.583	0.581	0.599
First-all	0.757	0.607	0.639	0.705

quality (71%). It is interesting that many of the manually evaluated first translation candidates were also cognates (34%), especially among nouns (48%). In 23% of the cases, the incorrect translation candidates were semantically closely related words, such as hypernyms, co-hyponyms or opposites that are not correct themselves but probably still contribute to good modelling of contexts and thereby help bilingual lexicon extraction.

Table 4.26 gives the results of automatic evaluation of bilingual lexicon extraction with the baseline lexicon that was extended with first translation candidates. As with cognates, nominal first translations have the most impact on the size of the extended lexicon (2510 new entries), but share an almost identical precision gain with adjectives. Best performance, again, is achieved when adding all parts of speech to the seed lexicon, improving the baseline results by 11.3 MRR points, 85% more than in the case of adding cognates to the seed lexicon. This shows a higher importance of adding high-frequency first translation candidates to the seed lexicon as opposed to adding contextually proven cognates.

4.4.3.5 Combining Cognates and First Translations of the Most Frequent Words to Extend the Seed Lexicon

In order to study the total impact of seed lexicon extension with new information that was extracted from the corpus automatically, we combine the cognates and first translation candidates in order to measure the gain of both information sources. Thereby, the seed lexicon was extended with 2303 new entries, amounting to 35,798 entries overall. When we start adding cognates and then add first translations of the most frequent words (overwriting the existing lexicon entries with new information), we achieve a precision of 73.1%, while changing the order gives a slightly lower

score of 72.3%. Once again, this shows that first translations are more beneficial for the context vector translation for bilingual lexicon extraction.

We performed an additional manual evaluation of a random sample of 100 translation equivalents that we extracted when using the best-performing extended seed lexicon. This evaluation showed that 88 entries contained the correct translation among the ten top-ranking translation candidates and that 64 of those were found in the first position. What is more, many lists of the ten top-ranking translation candidates contained not one but several correct translation variants. Also, as many as 59 of the correct translation candidates were cognates, and 41 of them appeared in the first position, suggesting that the results could be improved even more by a final re-ranking of translation candidates based on cognate clues, which we describe in the following section.

4.4.3.6 Re-ranking of Translation Candidates with Cognate Clues

Once we obtained translation candidates ranked according to our similarity measure, the final re-ranking of the ten highest-ranking translation candidates was performed. The source word was compared by the previously described BI-SIM function with each of the ten translation candidates. Two lists were formed, one with words satisfying the 0.7 cognate threshold, and another one with the words not satisfying the criterion. Finally, the lists were concatenated by putting the cognate list of translation equivalents in front of the non-cognate list.

Table 4.27 shows the baseline results for all parts of speech, the results obtained by using the extended seed lexicon and the results of re-ranking the final translation candidates.

The best score is achieved for nouns with a total increase in MRR of 40%. The overall improvement of the MRR results for all parts of speech is 34.6%.

4.4.4 Discussion

In this section, first, we presented a knowledge-light approach for bilingual lexicon extraction from comparable corpora. The method was evaluated on two comparable corpora for the English–Slovene language pair—a corpus with a high level of comparability and a corpus of a low level of comparability. The results showed that for the highly comparable corpus, the method achieves stable results regardless

Table 4.27 Automatic evaluation of translation extraction per part of speech with re-ranking

PoS	Baseline	Extended	Re-ranked
Nouns	0.605	0.768	0.848
Adjectives	0.566	0.605	0.698
Verbs	0.579	0.658	0.735
All	0.592	0.731	0.797

of the corpus size. However, for the lowly comparable corpus, it is evident that the method's performance is almost proportional to the corpus size.

Then, we also presented a method for bilingual lexicon extraction from comparable corpora for closely related languages. When tested on a comparable news corpus for Croatian and Slovene, the method has shown to outperform related approaches both in terms of precision (0.797 for nouns, adjectives and verbs) and recall (46%). Unlike most related approaches, the method deals with all content words (not just nouns) and enriches the seed lexicon used for translating context vectors from the results of the translation procedure itself, thereby experiencing a 35% precision increase in the lexicon extraction task. The proposed approach is directly applicable to a number of other similar language pairs for which there is a lack of bilingual lexica.

References

Apidianaki, M., Ljubešić, N., & Fišer, D. (2013). Vector disambiguation for translation extraction from comparable corpora resources used comparable corpus. *Informatica (Slovenia), 37*(2), 193–201.

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation, 43*(3), 209–226.

Bourigault, D. (1992). Surface grammatical analysis for the extraction of terminological noun phrases. *Proceedings of the 14th Conference on Computational Linguistics* (Vol. 3, pp. 977–981). Association for Computational Linguistics.

Chen, S. F., & Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer Speech and Language, 13*(4), 359–393.

Chiao, Y.-C., & Zweigenbaum, P. (2002). Looking for candidate translational equivalents in specialized, comparable corpora. *Proceedings of the 19th International Conference on Computational Linguistics* (Vol. 2). Association for Computational Linguistics.

Chinchor, N. (1997). MUC-7 named entity task definition. *Proceedings of the 7th Conference on Message Understanding*.

Cohen, J. (1968). Weighted Kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220.

Dagan, I., & Church, K. (1994). Termight: Identifying and translating technical terminology. *Proceedings of the Fourth Conference on Applied Natural Language Processing* (pp. 34–40). Association for Computational Linguistics.

Daille, B. (1994). Study and implementation of combined techniques for automatic extraction of terminology. *Proceedings of the Workshop The Balancing Act: Combining Symbolic and Statistical Approaches to Language (Language, Speech, and Communication)* (pp. 29–36). Association for Computational Linguistics, Las Cruces, NM.

Daille, B., & Morin, E. (2008). Effective compositional model for lexical alignment. *Proceedings, IJCNLP 2008: Third International Joint Conference on Natural Language Processing* (Vol. 1, pp. 95–102).

Damerau, F. J. (1993). Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management, 29*(4), 433–447.

Déjean, H., Gaussier, E., Renders, J.-M., & Sadat, F. (2005). Automatic processing of multilingual medical terminology: Applications to thesaurus enrichment and cross-language information retrieval. *Artificial Intelligence in Medicine, 33*(2), 111–124.

Delač, D., Krleža, Z., Šnajder, J., Bašić, B. D., & Šarić, F. (2009). TermeX: A tool for collocation extraction. In *Computational Linguistics and Intelligent Text Processing* (pp. 149–157). Springer.

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 363–370). Association for Computational Linguistics.

Fišer, D., & Ljubešic, N. (2011). Bilingual lexicon extraction from comparable corpora for closely related languages. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'11)* (pp. 125–131).

Fišer, D., Vintar, Š., Ljubešić, N., & Pollak, S. (2011). Building and using comparable corpora for domain-specific bilingual lexicon extraction. *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web* (pp. 19–26). Association for Computational Linguistics.

Fišer, D., Ljubešić, N., & Kubelka, O. (2012). Addressing polysemy in bilingual lexicon extraction from comparable corpora. *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC'12* (pp. 3031–3035).

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: The C-value/NC-Value Method. *International Journal on Digital Libraries, 3*(2), 115–130.

Fung, P. (1998). A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In *Machine translation and the information soup* (pp. 1–17). Springer.

Fung, P., & McKeown, K. (1997). A technical word- and term-translation aid using noisy parallel corpora across language groups. *Machine Translation, 12*(1–2), 53–87.

Georgantopoulos, B., & Piperidis, S. (2000). A hybrid technique for automatic term extraction. *Proceedings of the ACIDCA 2000 Conference*.

Grefenstette, G. (1994). *Explorations in automatic thesaurus discovery*. Heidelberg: Springer.

Grefenstette, G. (1999). The World Wide Web as a resource for example-based machine translation tasks. *Proceedings of the ASLIB Conference on Translating and the Computer* (Vol. 21).

Grigonyte, G., Rimkute, E., Utka, A., & Boizou, L. (2011). Experiments on lithuanian term extraction. *Proceedings of the NODALIDA 2011 Conference* (pp. 82–89).

Ion, R. (2007). *Word sense disambiguation methods applied to English and Romanian*. PhD Thesis, Romanian Academy, Bucharest.

Justeson, J. S., & Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering, 1*(01), 9–27.

Kageura, K., & Umino, B. (1996). Methods of automatic term recognition: A review. *Terminology, 3*(2), 259–289.

Kilgarriff, A. (2001). Comparing Corpora. *International Journal of Corpus Linguistics, 6*(1), 97–133.

Kochanski, G. (2006). *Lecture 4-good-turing probability estimation*. Oxford.

Koehn, P., & Knight, K. (2002). Learning a translation lexicon from monolingual corpora. *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition* (Vol. 9, pp. 9–16). Association for Computational Linguistics.

Kondrak, G., & Dorr, B. (2004). Identification of confusable drug names: A new approach and evaluation methodology. *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics.

Kravalová, J., & Žabokrtský, Z. (2009). Czech named entity corpus and SVM-based recognizer. *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration* (pp. 194–201). Association for Computational Linguistics.

Krugļevskis, V. (2010). Semi-automatic term extraction from Latvian texts and related language technologies. *Magyar Terminologia (Journal of Hungarian Terminology)*.

Kruglevskis, V., & Vancane, I. (2005). Term extraction from legal texts in Latvian. *Proceedings of the Second Baltic Conference on Human Language Technologies* (pp. 155–161).

Lee, L., Aw, A., Zhang, M., & Li, H. (2010). EM-based hybrid model for bilingual terminology extraction from comparable corpora. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 639–646). Association for Computational Linguistics.

Liao, W., & Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing* (pp. 58–65). Association for Computational Linguistics.

Ljubešić, N., & Erjavec, T. (2011). hrWaC and slWac: Compiling web corpora for Croatian and Slovene. *Text, Speech and Dialogue 2011 Conference Proceedings* (pp. 395–402). Springer.

Ljubešić, N., & Fišer, D. (2011). Bootstrapping bilingual lexicons from comparable corpora for closely related languages. *Text, Speech and Dialogue* (pp. 91–98).

Ljubešić, N., Fišer, D., Vintar, Š., & Pollak, S. (2011). Bilingual lexicon extraction from comparable corpora: A comparative study. *First International Workshop on Lexical Resources*.

Ljubešić, N., Vintar, Š., & Fišer, D. (2012). Multi-word term extraction from comparable corpora by combining contextual and constituent clues. *Proceedings of the 5th Workshop on Building and Using Comparable Corpora (BUCC 2012)* (pp. 143–147). ELRA, Istanbul.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

Marsi, E., & Krahmer, E. (2010). Automatic analysis of semantic similarity in comparable text through syntactic tree matching. *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 752–760). Association for Computational Linguistics.

Mima, H., & Ananiadou, S. (2000). An application and evaluation of the C/NC-value approach for the automatic term recognition of multi-word units in Japanese. *Terminology, 6*(2), 175–194.

Morin, E., & Prochasson, E. (2011). Bilingual lexicon extraction from comparable corpora enhanced with parallel corpora. *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web* (pp. 27–34).

Morin, E., Daille, B., Takeuchi, K., Kageura, K. (2007). Bilingual terminology mining – Using brain, not brawn comparable corpora. *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (pp. 664–671). Association for Computational Linguistics.

Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes, 30*(1), 3–26.

Och, F. J., & Ney, H. (2000). Improved statistical alignment models. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 440–447). Association for Computational Linguistics.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics, 29*(1), 19–51.

Otero, P. G. (2007). Learning bilingual lexicons from comparable English and Spanish corpora. *Proceedings of MT Summit XI* (pp. 191–198).

Pantel, P., & Lin, D. (2001). A statistical corpus-based term extractor. *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence – Advances in Artificial Intelligence (AI 2001)* (pp. 36–46). Ottawa, Canada. Berlin: Springer.

Paukkeri, M.-S., Nieminen, I. T., Pöllä, M., & Honkela, T. (2008). A language-independent approach to keyphrase extraction and evaluation. *Proceedings of COLING 2008* (pp. 83–86).

Pinnis, M. (2012). Latvian and lithuanian named entity recognition with TildeNER. *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)* (pp. 1258–1265). European Language Resources Association (ELRA), Istanbul, Turkey.

Pinnis, M., & Goba, K. (2011). Maximum entropy model for disambiguation of rich morphological tags. In C. Mahlow & M. Piotrowski (Eds.), *Proceedings of the 2nd International Workshop on Systems and Frameworks for Computational Morphology* (pp. 14–22). Zurich: Springer.

Pinnis, M., & Skadiņš, R. (2012). MT adaptation for under-resourced domains – What works and what not. *Human Language Technologies – The Baltic Perspective – Proceedings of the Fifth International Conference Baltic HLT 2012* (Vol. 247, pp. 176–184). Tartu, Estonia: IOS Press.

Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., & Gornostay, T. (2012). Term extraction, tagging, and mapping tools for under-resourced languages. *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012)* (pp. 193–208), Madrid.

Rapp, R. (1995). Identifying word translations in non-parallel texts. *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics* (pp. 320–322). Computation and Language, Association for Computational Linguistics.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 519–526). Association for Computational Linguistics, Stroudsburg, PA.

Saralegi, X., San Vicente, I., & Gurrutxaga, A. (2008). Automatic extraction of bilingual terms from comparable corpora in a popular science domain. *Proceedings of Building and Using Comparable Corpora Workshop* (pp. 27–32).

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. *Proceedings of International Conference on New Methods in Language Processing* (Vol. 12, pp. 44–49).

Schütze, H. (1998). The hypertext concordance: A better back-of-the-book index. *Proceedings of First Workshop on Computational Terminology*.

Shao, L., & Ng, H. T. (2004). Mining new word translations from comparable corpora. *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA.

Shezaf, D., & Rappoport, A. (2010). Bilingual lexicon generation using non-aligned signatures. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 98–107). Association for Computational Linguistics.

Skadiņa, I. (2009). Jaunas iespējas attēlu meklēšanā: ģeotelpiskajā informācijā un valodu tehnoloģijās balstīta attēlu meklēšanas platforma TRIPOD. *Latvijas Nacionālās bibliotēkas zinātniskie raksti* (pp. 182–192). National Library of Latvia.

Smadja, F. (1993). Retrieving collocations from text: Xtract. *Computational Linguistics, 19*(1), 143–177.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Ştefănescu, D. (2010). *Intelligent information mining from multilingual corpora*. PhD Thesis, Romanian Academy, Bucharest.

Ştefănescu, D. (2012). Mining for term translations in comparable corpora. *The 5th Workshop on Building and Using Comparable Corpora* (pp. 98–103). Turkey, Istanbul.

Ştefănescu, D., Tufiş, D., & Irimia, E. (2006). Automatic identification and extraction of collocations from texts. *Proceedings of the 2nd Romanian Workshop for Linguistic Tools and Resources* (Vol. 3). Bucharest, Romania.

Ştefănescu, D., Ion, R., & Boroş, T. (2011). TiradeAI: An ensemble of spellcheckers. *Proceedings of the Spelling Alteration for Web Search Workshop* (pp. 20–23).

Steinberger, R., Pouliquen, B., & Hagman, J. (2002). Cross-lingual document similarity calculation using the multilingual thesaurus EuroVoc. *Computational Linguistics and Intelligent Text Processing* (pp. 115–424).

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufi, D., Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)* (Vol. 4, pp. 2142–2147).

Tadić, M., & Šojat, K. (2003). Finding multiword term candidates in Croatian. In *Proceedings of Information Extraction for Slavic Languages 2003 Workshop* (pp. 102–107).

Tiedemann, J. (2005). Optimization of word alignment clues. *Natural Language Engineering, 11*(03), 279–293.

Tjong, E. F., & Sang, K. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. *Proceedings of the 6th Conference on Natural Language Learning* (Vol. 20, pp. 142–147). Association for Computational Linguistics, Taipei, Taiwan.

Todirascu, A., Gledhill, C., & Stefanescu, D. (2009). Extracting collocations in contexts. *Human Language Technology. Challenges of the Information Society* (pp. 336–349). Springer.

Tufi, D., & Irimia, E. (2006). RoCo-news: A hand validated journalistic corpus of Romanian. *Proceedings of the 5th LREC Conference* (pp. 869–872). Genoa, Italy.

Tufi, D., Ion, R., Ceauşu, A., & Ştefănescu, D. (2008). RACAI's linguistic web services. *Proceedings of the 6th Language Resources and Evaluation Conference-LREC* (pp. 327–333).

Vintar, Š. (2010). Bilingual term recognition revisited: The bag-of-equivalents term alignment approach and its evaluation. *Terminology, 16*(2), 141–158.

Voorhees, E. M. (2001). Overview of the TREC-9 question answering track. *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Weller, M., Gojun, A., Heid, U., Daille, B., & Harastani, R. (2011). Simple methods for dealing with term variation and term alignment. *Proceedings of the 9th International Conference on Terminology and Artificial Intelligence (TIA 2011)* (pp. 86–92).

Xiao, R., & McEnery, T. (2006). Collocation, semantic prosody, and near synonymy: A cross-linguistic perspective. *Applied Linguistics, 27*(1), 103–129.

Yu, K., & Tsujii, J. (2009). Bilingual dictionary extraction from Wikipedia. *Proceedings of Machine Translation Summit XII* (pp. 379–386).

Zeller, I. (2005). *Automatinis terminu atpazinimas ir apdorojimas*. VDU Lietuviu Kalbos Institutas.

Chapter 5
Mapping and Aligning Units from Comparable Corpora

Ahmet Aker, Alexandru Ceaușu, Yang Feng, Robert Gaizauskas, Sabine Hunsicker, Radu Ion, Elena Irimia, Dan Ştefănescu, and Dan Tufiș

Abstract Extracting parallel units (e.g. sentences or phrases) from comparable corpora in order to enrich existing statistical translation models is an avenue that has attracted a lot of research in recent years. There are experiments that convincingly show how parallel sentences extracted from comparable corpora are able to improve statistical machine translation (SMT). Yet, the existing body of research on the subject does not take into account the degree of comparability of the corpus being processed nor the computation time that it takes to extract translational similar pairs from a corpus of a given size. We will show that the performance of a parallel unit extractor crucially depends on the degree of comparability, such that it is more difficult to mine for parallel data in a weakly comparable corpus than a strongly comparable corpus.

Most of the research in parallel data mining from comparable corpora focusses on parallel sentence mining, but parallel phrase mining (i.e. sub-sentential fragments) is of equal importance, because it can be more robust in the presence of weakly comparable corpora that usually do not contain whole translated sentences. We will present different approaches to parallel sentence and phrase mining from comparable corpora developed in the ACCURAT project, and we will evaluate them both in terms of absolute measures (e.g., *P*, *R* and *F*1) and with respect to their ability to generate significant improvements of the BLEU scores of a statistical translation system. Comprehensive testing of these algorithms in the context of statistical machine translation will be undertaken in Chap. 6.

Chapter editors: Radu Ion and Dan Tufiș

A. Aker · Y. Feng · R. Gaizauskas (✉)
University of Sheffield, Sheffield, UK
e-mail: R.Gaizauskas@sheffield.ac.uk

A. Ceaușu · R. Ion · E. Irimia · D. Ştefănescu · D. Tufiș
Research Institute for Artificial Intelligence of the Romanian Academy (RACAI), Bucharest, Romania

S. Hunsicker
The German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_5

5.1 Introduction

Statistical machine translation (SMT) is in a constant need of good-quality training data both for translation models and for the language models. Regarding the latter, monolingual corpora is evidently easier to collect than parallel corpora, and the truth of this statement is even more obvious when it comes to pairs of languages other than those both widely spoken and computationally well-treated around the world, such as English, Spanish, French or German. Parallel corpora acquisition from the World Wide Web has traditionally been geared towards identifying similar structures (searching for anchors in titles, sections, images with identical descriptions, same inbound/outbound links, etc.) of the parallel documents and/or their referring URLs (Zhang et al. 2006), but recent research has been oriented towards scoring the candidate target document as to how well it dictionary-translates the source counterpart (Tsvetkov and Wintner 2010). In contrast, comparable corpora are easier to collect than parallel corpora, basically because no parallel websites need to be identified a priori and no (usually complicated) HTML parsing is required in order to identify the parallel parts at crawling time. Comparable corpora are exempted from the tedious task of particular HTML parsing by assuming that documents are related in some (explicitly stated) way, and thus, it is sufficient for the crawler to just collect all the text from the HTML document.

Comparable corpora came as a possible solution to the problem of the scarcity of parallel corpora with the promise that it may serve as a reliable source for parallel data extraction. We accept the definition of a comparable corpus from Munteanu and Marcu (2005): a pair of comparable documents '*... while not parallel in the strict sense, are somewhat related and convey overlapping information*'. The relatedness of a pair of comparable documents has been defined and experimented with in many ways, including the membership to the same domain/topic/genre, the same publication date/period (especially for News documents), the appearance of the same named entities (NEs) (names of persons, geographical entities, numeric entities, dates, times, etc.) and so on. The predominant paradigm for automatically collecting domain-dependent, bilingual comparable corpora from the Web is the cross-language information retrieval method based on seed lists of source document URLs (Talvensaari et al. 2008). One usually begins by collecting a list of t terms as seed data in both the source and the target languages. Each term (in each language) is then queried on the most popular search engine, and the first N document hits are retained. The final corpus will contain $t \times N$ documents in each language, and the document boundaries are often disregarded in subsequent usage. Collection of comparable corpora in the ACCURAT project is the subject of Chap. 3.

The direct consequence of the nature of comparable corpora and its collecting mechanism is the large size compared to (truly) parallel corpora collected from the World Wide Web. The difference is several orders of magnitude, and the size of the comparable corpora prompts for dealing with computational challenges that are not encountered when searching for parallel units in parallel corpora. If we have

M documents in the source language and N documents in the target language, a parallel data-mining algorithm should look in every document pair from the set containing MN pairs in order to achieve the maximum recall. But when M and/or N are/is large, this is not feasible, and one has to pre-align the documents in the comparable corpus as to the likelihood to contain parallel data.

At this point, it is important to stress the importance of the **pairing of documents** in a comparable corpus. Suppose that we want to word-align a bilingual comparable corpus consisting of M documents per language, each with k words, using the IBM-1 word alignment algorithm (Brown et al. 1993). For each source word, this algorithm searches the target words that have a maximum translation probability with the source word. Aligning all the words in our corpus with no regard to document boundaries would yield a time complexity of k^2M^2 operations. The alternative would be to find a 1:p (with p being a small positive integer, usually 1, 2 or 3) document assignment (a set of aligned document pairs) that would enforce the 'no search outside the document boundary' condition when doing word alignment and have the advantage of reducing the time complexity to k^2Mp operations. When M is large, the reduction may actually be vital for achieving a result in a reasonable amount of time. The downside of this simplification is the loss of information: two documents may not be correctly aligned and thus deprive the word-alignment algorithm of the part of the search space that would have contained the right alignments.

The task of mining for parallel sentences in comparable corpora is much more difficult than aligning sentences in parallel corpora. Sentence alignment in parallel corpora usually exploits simple empirical evidence (turned into assumptions), such as

1. The length of a sentence is directly proportional to the length of its translation (Gale and Church 1993),
2. The discourse flow is necessarily the same in both parts of the bi-text (Gale and Church 1993).

Consequently, the extraction tools search for parallel sentences around the same (relative) text positions, and this characteristic of the parallel corpora makes sentence alignment a much easier task when compared to the kind of work undertaken for comparable corpora. Specifically, we are referring to the positional information of the translation units in parallel texts (paragraphs and sentences), which constitute a natural pruning technique when searching for parallel sentences in parallel corpora: for the ith sentence of the source document, the sought aligned sentence is to be found in a window of $\pm k$ sentences around the jth sentence of the target document, where i and j are proportional. In the case of comparable corpora, this assumption does not hold anymore. Parallel sentences, should they exist at all, are scattered all around the source and target documents, and as such, any two sentences have to be processed in order to determine if they are parallel or not. Moreover, we aim at also finding pairs of quasi-parallel sentences that are not parallel entirely but contain spans of contiguous text that is parallel. Thus, finding parallel sentences in comparable corpora is confronted with the vast search space one has to consider, since any positional clues indicating parallel or partially parallel sentences are not available.

When mining for parallel sentences/phrases in a comparable document pair, the brute force approach is to analyse every element of the Cartesian product built between the two sets containing sentences/phrases in the source and target languages. This approach is clearly impractical, because the resulting algorithm would be very slow and/or would consume a lot of memory.[1] In order to reduce the search space, we turned to a framework that belongs to information retrieval (IR): cross-language information retrieval (CLIR). The idea is simple: use a search engine to find sentences in the target corpus that are the most probable translations of a given sentence from the source corpus. The first step is to consider the target sentences as documents and index them. Then, for each sentence in the source corpus, one selects the content words and translates them into the target language according to a given dictionary. The translations are used to form a Boolean query that is then fed to the search engine. The top hits are considered to be translation candidates.

Using the CLIR approach to select a set of candidate target sentences (out of all target sentences) for the input source sentence is one way to dramatically reduce the search space. The reduced search space will serve another practical concern: the execution time. Thus, each candidate target sentence can be compared with the source input sentence using a computationally much more complex translation similarity measure (TSM) that would otherwise require an unacceptable amount of time to finish analysing all possible pairs.

5.2 Related Work

Extracting parallel data from comparable corpora in order to enrich existing statistical translation models is an avenue that has attracted a fair amount of research in recent years, one of the major reasons being the fact that the Web can be seen as a vast source of comparable corpora. Generally speaking, we identified two approaches to parallel sentence mining from comparable corpora in the existing literature on the subject:

1. Pair documents in a comparable corpus (by using cross-language information retrieval techniques or translation similarity measures); for each document pair, generate the Cartesian product of the source and target sentence sets; score (by classification or a type of translation similarity measure) each sentence pair from the Cartesian product as to its parallelism degree;
2. Use an initial SMT system (trained on existing parallel/comparable data) to translate every source sentence from a comparable corpus into the target language; use standard information retrieval techniques to find the target sentences most similar to the translation and thus parallel to the initial source sentence.

[1] With the possible exception of parallelising the computations.

The main challenges in these two approaches are **the document pairing in a comparable corpus** and the parallelism degree scoring function of a sentence/phrase pair, which we will call **the translation similarity measure**.

One approach to cutting the search space is to perform document alignment inside the comparable corpus first and then to attempt extracting parallel sentences by inspecting only the constructed document pairs. Document pairing inside a comparable corpus has been attempted in various ways. For instance, Munteanu and Marcu (2005), followed in spirit by Tillmann (2009) and Quirk et al. (2007), solved the document pairing problem by generating target language queries from a source document by using the first 5 most probable translations of each word and then interrogating a standard IR engine to find the most similar 20 target documents for the source document. Fung and Cheung (2004) use a word translation similarity measure to discover similar documents.

Another approach to reduce the search space that received significant attention is the use of CLIR to find translation candidates for input source sentences. Rauf and Schwenk (2011) index the target sentences directly and use a baseline (BL) SMT system to translate the input source sentence into the target language. Munteanu and Marcu (2005) use a dictionary to translate some of the words of the source sentence and then use these translations to query a database for finding matching translation candidate sentences. They choose the translation candidate sentences based on word overlap with the (translation of the) source sentence.

Given a pair of sentences, the first in the source language and the last in the target language, the job of the translation similarity measure is to assess 'how parallel' the two sentences are.

Munteanu and Marcu (2002) align sentences in an English–French comparable corpus of 1.3M words per language by comparing suffix trees of the sentences. Each sentence from each part of the corpus is encoded as a suffix tree, which is a tree that stores each possible suffix of a string from the last character to the full string. Using this method, Munteanu and Marcu are able to detect correct sentence alignments with a precision of 95% (out of 100 human-judged and randomly selected sentences from the generated output). The running time of their algorithm is approximately 100 hours for 50,000 sentences in each of the languages.

Fung and Cheung (2004) attempt parallel sentence mining from 'very-non-parallel corpora' by devising a bootstrapping mechanism in which, after an initial document pairing and consequent sentence alignment using a lexical overlapping similarity measure, the IBM-4 model (Brown et al. 1993) is employed to enrich the bilingual dictionary that is used by the similarity measure. The process is repeated until the set of identified aligned sentences does not grow anymore. The precision of this method on English–Chinese sentence alignment is 65.7% (out of the top 2500 identified pairs).

Munteanu and Marcu (2005) (followed by Tillmann (2009) with an improved variation) devised a maximum entropy classifier that will assign the label of 'parallel' or 'not parallel' to the pair of sentences. Among the features involved, we can mention the following: fertility (Brown et al. 1993), contiguous aligned spans of words, sentence lengths (with length difference and length ratio), percentage of the

words in a source sentence that have translations in a target sentence (translations are taken from pre-existing translation lexicons), etc. The training data consisted of a small parallel corpus of 5000 sentences per language. Since the number of negative instances ($5000^2 - 5000$) is far larger than the number of positive ones (5000), the negative training instances were selected randomly out of instances that passed a certain word overlap filter (see the paper for details). The classifier precision is around 97% with a recall of 40% for the Chinese–English task and around 95% with a recall of 41% for the Arabic–English task. Hewavitharana and Vogel (2011) also adopt a classification approach for parallel phrase extraction. However, their approach requires manual intervention in data preparation.

Rauf and Schwenk (2011), and independently Thi Ngoc Diep et al. (2010), use a previously trained SMT system to translate the source sentence into the target language and then apply MT assessment measures such as WER (the Levenshtein distance), TER (Snover et al. 2006), and TERp (Snover et al. 2009) in order to monolingually see how similar the translation of the source sentence is to the target sentence.

A radical, different approach to determine if the source sentence is similar to the target sentence is given by the generative models of Quirk et al. (2007). They assume that the target sentence (or phrase) is conditionally generated by the source sentence and proceed to model the probability of this generation. The parameters of the model are the source and target words and the positions of the source words that generated the translated target words.

5.3 Document Alignment in Comparable Corpora

If the comparable corpus contains a large number of documents both in the source and in the target language, then it is impractical to evaluate every source sentence with every target sentence to see if they are parallel or not. Thus, one approach to cutting this search space is to perform document alignment inside the comparable corpus first and then to attempt extracting parallel sentences by inspecting only the constructed document pairs. Recently, Ion et al. (2011a) devised an expectation-maximization (EM) algorithm to reveal the hidden document alignments in a comparable corpus on the assumption that there are certain word translation pairs that are very good indicators for these alignments. The presentation of this section follows the one in Ion et al. (2011b).

5.3.1 EMACC

We propose *a specific instantiation of the well-known general EM algorithm* for aligning documents, which we will name EMACC (an acronym for 'expectation-maximization alignment for comparable corpora'). We draw our inspiration from the

famous IBM models (specifically from the IBM-1 model) for word alignment (Brown et al. 1993) where the translation probability [Eq. (5.5)] is modelled through an EM algorithm where the hidden variable a models the assignment (1:1 word alignments) from the French sequence of words (indexes) to the English one.

By analogy, we imagined that between two sets of documents—let's call them $\boldsymbol{E}$ and $\boldsymbol{F}$, there is *an assignment* (a sequence of 1:1 document correspondences[2]), the distribution of which can be modelled by a hidden variable z taking values in the set {true, false}. This assignment will be largely determined by the existence of word translations between a pair of documents, translations that can differentiate between one another in their ability to indicate a correct document alignment versus an incorrect one. In other words, we hypothesise that there are certain pairs of translation equivalents that are better indicators of a correct document correspondence than other pairs of translation equivalents.

We take the general formulation and derivation of the EM optimisation problem from Borman (2009). The general goal is to optimise $P(X|\Theta)$, that is to find the parameter(s) Θ for which $P(X|\Theta)$ is maximum. In a sequence of derivations that we are not going to repeat here, the general EM equation is given by

$$\Theta_{n+1} = \underset{\Theta}{\operatorname{argmax}} \sum_{z} P(z|X, \Theta_n) \ln P(X, z|\Theta), \tag{5.1}$$

where $\sum_z P(z|X, \Theta_n) = 1$. At step $n + 1$, we try to obtain a new parameter Θ_{n+1} that is going to maximise (the maximisation step) the sum over z (the expectation step), which in turn depends on the best parameter Θ_n obtained at step n. Thus, in principle, the algorithm *should iterate over the set of all possible* Θ *parameters*, compute the expectation expression for each of these parameters, and choose the parameter(s) for which the expression has the largest value. But as we will see, in practice, the set of all possible parameters has a dimension that is exponential in terms of the number of parameters. This renders the problem intractable, and one should back off to heuristic searches in order to find a near-optimal solution.

We now introduce a few notations that we will operate with from this point forward. We suggest to the reader *to frequently refer to these notations* in order to properly understand the next equations:

- $\boldsymbol{E}$ is the set of source documents, $|E|$ is the cardinal of this set.
- $\boldsymbol{F}$ is the set of target documents with $|F|$ as its cardinal.
- d_{ij} is a pair of documents, $d_i \in E$ and $d_j \in F$.
- w_{ij} is a pair of translation equivalents $\langle w_i, w_j \rangle$ such that w_i is a lexical item that belongs to d_i and w_j is a lexical item that belongs to d_j.
- $\boldsymbol{T}$ is the set of all existing pairs of translation equivalents $\langle w_{ij}, p \rangle$ and p is the translation probability score, as the one given, e.g., by GIZA++ (Gao and Vogel

[2]Or 'alignments' or 'pairs.' These terms will be used with the same meaning throughout this section.

2008). We assume that GIZA++ translation lexicons already exist for the pair of languages of interest.

- In order to tie Eq. (5.1) to our problem, we define its variables as follows:
- Θ is the sequence of 1:1 document alignments of the form $D_{i_1j_1}, D_{i_2j_2}, \ldots$ where $D_{ij} \in \{d_{ij} | d_i \in E, d_j \in F\}$. We call Θ *an assignment,* which is basically a sequence of 1:1 document alignments. If there are $|E|$ 1:1 document alignments in Θ and if $|E| \leq |F|$, then the set of all possible assignments has the cardinal equal to $|E|!\binom{|F|}{|E|}$ where $n!$ is the factorial function of the integer n and $\binom{n}{k}$ is the binomial coefficient. It is clear now that with this kind of dimension of the set of all possible assignments (or Θ parameters), we cannot simply iterate over it in order to choose the assignment that maximises the expectation.
- $z \in \{\text{true}, \text{false}\}$ is the hidden variable that signals if a pair of documents d_{ij} represents a correct alignment (true) or not (false).
- X is the sequence of pairs of translation equivalents W_{ij} from $\boldsymbol{T}$ in the order they appear in each document pair from Θ.

Having defined the variables in Eq. (5.1) this way, we aim at maximising the translation equivalents probability over a given assignment, $P(X|\Theta)$. In doing so, through the use of the hidden variable z, we are also able to find the 1:1 document alignments that attest for this maximisation.

We proceed by reducing Eq. (5.1) to a form that is readily amenable to software coding. That is we aim at obtaining some distinct probability tables that are going to be (re-)estimated by the EM procedure. The general form of the derived EM equation is

$$\Theta_{n+1} = \underset{\Theta}{\operatorname{argmax}} \left[\ln P(X|\Theta) + \ln P\big(\text{true}|\Theta\big)\right]. \tag{5.2}$$

Equation (5.2) suggests a method of updating the assignment probability $P(\text{true}|\Theta)$ with the lexical alignment probability $P(X|\Theta)$ in an effort to provide the alignment clues that will 'guide' the assignment probability towards the correct assignment. Now, all that remains to be done is to define the two probabilities.

The **lexical document alignment probability** $P(X|\Theta)$ is defined as follows:

$$P(X|\Theta) = \prod_{d_{ab} \in \Theta} \frac{\sum\limits_{w_{ij} \in X} P\big(d_{ab}|w_{ij}\big)}{|E||F|}, \tag{5.3}$$

where $P(d_{ab}|w_{ij})$ is the simplified lexical document alignment probability, which is initially equal to $P(w_{ij})$ from the set $\boldsymbol{T}$. This probability is to be read as 'the contribution w_{ij} makes to the correctness of the d_{ab} alignment'. We want that the alignment contribution of one pair of translation equivalents w_{ij} to be distributed over the set of all possible document pairs, thus enforcing that

$$\sum_{d_{ab} \in \{d_{xy} | d_x \in E, d_y \in F\}} P(d_{ab}|w_{ij}) = 1. \tag{5.4}$$

The summation over X in Eq. (5.3) is actually over all pairs of translation equivalents that are to be found only in the current d_{ab} document pair, and the presence of the product $|E||F|$ ensures that we still have a probability value.

The **assignment probability** $P(\text{true}|\Theta)$ is also defined in the following way:

$$P(\text{true}|\Theta) = \prod_{d_{ab} \in \Theta} P(d_{ab}|\text{true}), \tag{5.5}$$

for which we enforce the following condition:

$$\sum_{d_{ab} \in \{d_{xy} | d_x \in E, d_y \in F\}} P(d_{ab}|\text{true}) = 1. \tag{5.6}$$

Using Eqs. (5.2), (5.3) and (5.5), we deduce the final computation-ready EM equation:

$$\Theta_{n+1} = \underset{\Theta}{\text{argmax}} \sum_{d_{ab} \in \Theta} \left[\ln \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|E||F|} + \ln P(d_{ab}|\text{true}) \right]. \tag{5.7}$$

As it is, Eq. (5.7) suggests an exhaustive search *in the set of all possible* Θ *parameters*, in order to find the parameter(s) for which the expression that is the argument of 'argmax' is maximum. But the size of this set is prohibitive to the attempt of enumerating each Θ assignment and computing the expectation expression. Our quick solution to this problem was to directly construct the 'best' Θ assignment[3] using a *greedy algorithm*: simply iterate over all possible 1:1 document pairs and, for each document pair $d_{ab} \in \{d_{xy} | d_x \in E, d_y \in F\}$, compute the alignment count (since it is not a probability, we call it a 'count', following the terminology of the IBM-1 model)

$$\ln \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|E||F|} + \ln P(d_{ab}|\text{true})$$

[3]We did not attempt to find the mathematical maximum of the expression from Eq. (5.7), and we realise that the consequence of this choice and of the greedy search procedure is not finding the true optimum.

Then, we construct the best 1:1 assignment Θ_{n+1} by choosing those pairs d_{ab} for which we have counts with the maximum values. Before this cycle (which is the basic EM cycle) is resumed, we perform the following updates:

$$P(d_{ab}|\text{true}) \leftarrow P(d_{ab}|\text{true}) + \frac{\sum_{w_{ij} \in X} P(d_{ab}|w_{ij})}{|E||F|} \tag{5.8a}$$

$$P(d_{ab}|w_{ij}) \leftarrow \sum_{d_{xy} \in \Theta_{n+1}} P(d_{xy}|w_{ij}). \tag{5.8b}$$

and normalise the two probability tables with Eqs. (5.6) and (5.4). The first update is to be interpreted as the contribution the lexical document alignment probability makes to the alignment probability. The second update equation aims at boosting the probability of a translation equivalent if and only if it is found in a pair of documents belonging to the best assignment so far. In this way, we hope that the updated translation equivalent will make a better contribution to the discovery of a correct document alignment that has not yet been discovered at step $n + 1$.

Before we start the EM iterations, we need to initialise the probability tables $P(d_{ab}|\,\text{true})$ and $P(d_{ab}|\,w_{ij})$. For the second table, we used the GIZA++ scores that we have for the w_{ij} pairs of translation equivalents and normalised the table with Eq. (5.4). For the first probability table, we have (and tried) two choices:

- (**D1**) a uniform distribution: $\frac{1}{|E||F|}$
- (**D2**) a lexical document alignment measure $L(d_{ab})$ (values between 0 and 1) that is computed directly from a pair of documents d_{ab} using the w_{ij} translation equivalents pairs from the dictionary $\boldsymbol{T}$

$$L(d_{ab}) = \frac{\sum_{w_i} f_{d_a}(w_i) \sum_{w_j} f_{d_b}(w_j)}{|d_a||d_b|}, \tag{5.9}$$

where $|d_a|$ is the number of words in document d_a and $f_{d_a}(w_i)$ is the frequency of word w_i in document d_a (please note that, according to our definitions, w_{ij} *is not* a random pair of words, but a pair of *translation equivalents*). If every word in the source document has at least one translation (of a given threshold probability score) in the target document, then this measure is 1. We normalise the initialised table using this measure with Eq. (5.6).

EMACC finds only 1:1 alignments of textual units in its present form, but a document pair d_{ab} can be easily extended to a document bead following the example from Chen (1993). The main difference between the algorithm described by Chen and ours is that the search procedure reported there is invalid for comparable corpora in which no pruning is available due to the nature of the corpus. A second very important difference is that Chen only relies on lexical alignment information, on the parallel nature of the corpus, and on sentence length correlations, while we add the

probability of the whole assignment, which, when initially set to the $D2$ distribution, produces a significant boost of the precision of the alignment.

5.3.2 EMACC Evaluations

The test data for document alignment was compiled from the corpora that was previously collected in the ACCURAT project[4] and that is known to the project members as the 'initial comparable corpora' or ICC for short (Skadiņa et al. 2010). It is important to know the fact that ICC contains all types of comparable corpora from parallel to weakly comparable documents, but we have classified document pairs in three classes: parallel (class name: **p**), strongly comparable (**cs**), and weakly comparable (**cw**). We have considered the following pairs of languages: English–Romanian (en-ro), English–Latvian (en-lv), English–Lithuanian (en-lt), English–Estonian (en-et), English–Slovene (en-sl), and English–Greek (en-el). For each pair of languages, ICC also contains a gold standard (GS) list of document alignments that were compiled by hand for testing purposes.

We trained GIZA++ translation lexicons for every language pair using the DGT-TM corpus (Steinberger et al. 2012). The input texts were converted from their Unicode encoding to UTF-8 and were tokenised using the tokeniser web service described by Ceauşu (2009). Then, we applied a parallel version of GIZA++ (Gao and Vogel 2008) that gave us the translation dictionaries of only content words (nouns, verbs, adjective and adverbs) at wordform level. For Romanian, Lithuanian, Latvian, Greek and English, we had lists of inflectional suffixes that we used to stem entries in respective dictionaries and processed documents. Slovene remained the only language that involved wordform level processing.

The accuracy of EMACC is influenced by three parameters whose values have been experimentally set:

- The threshold over which we use translation equivalents from the dictionary $\boldsymbol{T}$ for textual unit alignment; values for this threshold (let's name it **`ThrGiza`**) are from the ordered set {0.001, 0.4, 0.8}.
- The threshold over which we decide to update the probabilities of translation equivalents with Eq. (5.8b); values for this threshold (named **`ThrUpdate`**) are from the same ordered set {0.001, 0.4, 0.8}.
- The top **`ThrOut%`** alignments from the best assignment found by EMACC. This parameter will introduce precision and recall with the 'perfect' value for recall equal to **`ThrOut%`**. Values for this parameter are from the set {0.3, 0.7, 1}.

We ran EMACC (10 EM steps) on every possible combination of these parameters for the pairs of languages in question on both initial distributions $D1$ and $D2$. For comparison, we also performed a baseline document alignment using the greedy

[4] http://www.accurat-project.eu/

algorithm of EMACC with Eq. (5.9) supplying the document similarity measure. The following four tables report a synthesis of the results that we have obtained, which, because of the lack of space, we cannot give in full. We omit the results of EMACC with *D*1 initial distribution, because the accuracy figures (both precision and recall) are always lower (10–20%) than those of EMACC with *D*2.

In every table above, the **P**/R column gives the maximum precision and the associated recall that EMACC was able to obtain for the corresponding pair of languages using the parameters (**Prms.**) from the next column. The P/**R** column gives the maximum recall with the associated precision that we obtained for that pair of languages.

The **Prms.** columns contain parameter settings for EMACC (see Tables 5.1 and 5.3) and for the *D*2 baseline algorithm (Tables 5.2 and 5.4): in Tables 5.1 and 5.3, values for `ThrGiza`, `ThrUpdate` and `ThrOut` are given from the top (of the cell) to the bottom, and in Tables 5.2 and 5.4,values of `ThrGiza` and `ThrOut` are also given from top to bottom (the `ThrUpdate` parameter is missing because the *D*2 baseline algorithm does not carry out re-estimation). The **#** column contains the size of the test set: the number of documents in each language that have to be paired. The search space is # * # and the gold standard contains # pairs of human aligned document pairs.

To ease comparison between EMACC and the *D*2 baseline for each type of corpora (strongly and weakly comparable), we highlighted in grey the maximal values between the two: either the precision in the **P**/R column or the recall in the P/**R** column.

In the case of strongly comparable corpora (Tables 5.1 and 5.2), we see that the benefits of re-estimating the probabilities of the translation equivalents (based on which we judge document alignments) begin to emerge with precisions for all pairs

Table 5.1 EMACC with *D*2 initial distribution on strongly comparable corpora

cs	**P**/R	Prms.	P/**R**	Prms.	#
en–ro	1/0.69	0.4 0.4 0.7	0.85/0.85	0.4 0.4 1	42
en–sl	0.96/0.28	0.4 0.4 0.3	0.83/0.83	0.4 0.4 1	302
en–el	0.97/0.29	0.001 0.8 0.3	0.80/0.80	0.001 0.4 1	407
en–lt	0.97/0.29	0.4 0.8 0.3	0.72/0.72	0.4 0.4 1	507
en–lv	0.95/0.28	0.4 0.4 0.3	0.79/0.79	0.001 0.8 1	560
en–et	0.88/0.26	0.4 0.8 0.3	0.55/0.55	0.4 0.4 1	987

Table 5.2 *D*2 baseline algorithm on strongly comparable corpora

cs	**P**/R	Prms.	P/**R**	Prms.	#
en–ro	1/0.69	0.4 0.7	0.85/0.85	0.4 1	42
en–sl	0.97/0.29	0.001 0.3	0.81/0.81	0.4 0.1	302
en–el	0.94/0.28	0.001 0.3	0.71/0.71	0.001 1	407
en–lt	0.95/0.28	0.001 0.3	0.72/0.72	0.001 1	507
en–lv	0.91/0.27	0.001 0.3	0.80/0.80	0.001 1	560
en–et	0.87/0.26	0.4 0.3	0.57/0.57	0.4 1	987

Table 5.3 EMACC with *D*2 initial distribution on weakly comparable corpora

cw	**P**/R	Prms.	P/**R**	Prms.	#
en–ro	1/0.29	0.4 0.001 0.3	0.66/0.66	0.4 0.001 1	68
en–sl	0.73/0.22	0.4 0.4 0.3	0.42/0.42	0.4 0.4 1	961
en–el	0.15/0.04	0.001 0.8 0.3	0.07/0.07	0.001 0.8 1	352
en–lt	0.55/0.16	0.4 0.8 0.3	0.28/0.28	0.4 0.8 1	325
en–lv	0.23/0.07	0.4 0.4 0.3	0.10/0.10	0.4 0.4 1	511
en–et	0.59/0.17	0.4 0.8 0.3	0.27/0.27	0.4 0.8 1	483

Table 5.4 *D*2 baseline algorithm on weakly comparable corpora

cw	**P**/R	Prms.	P/**R**	Prms.	#
en–ro	0.85/0.25	0.4 0.3	0.61/0.61	0.4 1	68
en–sl	0.65/0.19	0.4 0.3	0.39/0.39	0.4 1	961
en–el	0.11/0.03	0.4 0.3	0.06/0.06	0.4 1	352
en–lt	0.60/0.18	0.4 0.3	0.24/0.24	0.4 1	325
en–lv	0.13/0.03	0.4 0.3	0.09/0.09	0.4 1	511
en–et	0.48/0.14	0.001 0.3	0.25/0.25	0.4 1	483

of languages (except en-sl) being better than those obtained with the *D*2 baseline. But the real benefit of re-estimating the probabilities of translation equivalents along the EM procedure is visible from the comparison between Tables 5.3 and 5.4. Thus, in the case of weakly comparable corpora, in which EMACC with the *D*2 distribution is clearly better than the baseline (with the only exception of en-lt precision), due to the significant decrease in the lexical overlap, the EM procedure is able to produce important alignment clues in the form of re-estimated (bigger) probabilities of translation equivalents that, otherwise, would have been ignored.

It is important to mention the fact that the results we obtained varied a lot with values of the parameters `ThrGiza` and `ThrUpdate`. We observed, for the majority of studied language pairs, that lowering the value for `ThrGiza` and/or `ThrUpdate` (0.1, 0.01, 0.001...) would negatively impact the performance of EMACC due to the facts of *introducing noise* in the initial computation of the *D*2 distribution and *re-estimating (increasing) probabilities for irrelevant translation equivalents*. At the other end, increasing the threshold for these parameters (0.8, 0.85, 0.9...) would also result in performance decreasing due to the fact that *too few translation equivalents (be they all correct) are not enough to pinpoint correct document alignments,* since there are great chances for them to actually appear in all document pairs.

So, we have experimentally found that there is a certain balance between *the degree of correctness of translation equivalents* and *their ability to pinpoint correct document alignments*. In other words, the paradox resides in the fact that if a certain pair of translation equivalents is not correct but the respective words appear only in documents that correctly align to one another, then that pair is very important to the alignment process. Conversely, if a pair of translation equivalents has a very high probability score (thus being correct) but appears in almost every possible pair of documents, then that pair is not informative to the alignment process and must be excluded. We see now that the EMACC aims at finding the set of translation equivalents that is maximally informative with respect to the set of document alignments.

We have introduced the `ThrOut` parameter in order to have better precision. This parameter actually instructs EMACC to output only the top (according to the alignment score probability $P(d_{ab}|\,\text{true})$) `ThrOut%` of the document alignments that it has found. This means that, if all are correct, the maximum recall can only be `ThrOut%`. But another important function of `ThrOut` is to restrict the translation equivalents re-estimation [Eq. (5.8b)] for only the top `ThrOut%` alignments. In other words, only the probabilities of translation equivalents that are to be found in the top `ThrOut%` best alignments in the current EM step are re-estimated. We introduced this restriction in order to confine translation equivalents probability re-estimation to correct document alignments found so far.

Regarding the running time of EMACC, we can report that on a cluster with a total of 32 CPU cores (4 nodes) with 6–8 GB of RAM per node, the total running time is between 12 and 48 hours per language pair (about 2000 documents per language) depending on the setting of the various parameters.

5.4 Parallel Sentence Mining from Comparable Corpora

In what follows, we present two types of algorithms for mining for parallel sentences in a document-aligned comparable corpus:

1. An adaptation of the hybrid CLIR/translation similarity measure approach to parallel sentence mining from comparable corpora (LEXACC)
2. An algorithm that exhaustively grades every sentence pair to get its parallelism degree for each document pair in the comparable corpus implementing the translation similarity measure used by LEXACC (PEXACC—Parallel phrase EXtractor from Comparable Corpora)

For each algorithm, we present a host of experiments aimed at assessing the absolute performance (precision, recall and $F1$-measure) and application performance through SMT experimenting with data produced by these algorithms.

5.4.1 LEXACC

The presentation of LEXACC ("Lucene-based Parallel Sentence Extraction from Comparable Corpora") follows the one in Ştefănescu et al. (2012).

5.4.1.1 Indexing Target Sentences

Our goal is to implement a simple yet effective solution that is easily replicable. First, we split the target corpus into sentences and transform them so that we keep only stemmed non-functional words.[5] We also compute the average length in words (μ) and the standard deviation (σ) for target sentences. We consider a sentence s to be short if length(s) $\leq \mu + \sigma$ and long if length(s) $\geq \mu - \sigma$. We consider the medium-sized sentences, where $\mu - \sigma \leq$ length(s) $\leq \mu + \sigma$, to be both short and long.

Following the general description presented in the introduction, we use the C# implementation of Lucene[6] to index the target sentences as Lucene documents. For each such document, we introduce three additional searchable fields, two of them corresponding to the sentence length:

1. A field specifying if the sentence is *small*.
2. A field specifying if the sentence is *long*.
3. A field specifying the document where the target sentence belongs; this field is based on the document alignment information of the comparable corpus being processed, and it is optional if such alignment information is not supplied.

[5] We keep functional words lists for all languages.

[6] http://incubator.apache.org/projects/lucene.net.html

5.4.1.2 Finding Translation Candidates for Source Sentences

Given an input source sentence (out of the total S source sentences), the role of the search engine is to return a list of translation candidates that are to be further analysed. The number of hits h we take into account regulates the size of the new search space: $h \times S$. The larger it is, the higher the number of candidates, which can potentially increase the recall, but also the computational complexity. For each sentence in the source corpus, we generate a Lucene query as follows:

1. We employ a GIZA++ (Och and Ney 2003) dictionary previously created from existing parallel documents. This dictionary is expected to be small due to the lack of necessary resources. For each content word, we keep the 50 best translation equivalents, which are also content words, having translation probabilities above 0.1. Each of them is stemmed and added as an disjunctive query term (SHOULD occur).
2. We add two disjunctive query terms (SHOULD occur) standing for the length of the source sentence: *short* and *long*. Each of these terms can be boosted according to the importance one wants to give to matching source and target lengths. In our implementation, the value of the boosting factor is 2.
3. We add a compulsory query term (MUST occur) specifying the target document where the source sentence translation should be searched. However, this term can be added only if the document alignment information exists and has also been used during index creation.

After the query is constructed, we use it to interrogate the default Lucene search engine (no modifications on the relevance method) in order to get the best h hits.

5.4.1.3 Filtering

The filtering step is designed to further reduce the new search space, selecting only the best candidates for the final stage in which the translation similarity measure (Sect. 5.4.1.4) is applied. Filtering must be very fast and good enough not to filter out parallel data. We do this by computing a viability score for each candidate sentence pair and then keeping only those above the average. For a candidate pair formed by a source sentence s and a target sentence t, the formula is

$$\text{viabilityScore} = \alpha \times \beta \times \text{se} \times \text{sim}, \tag{5.10}$$

where *se* represents the score returned by the search engine and *sim* is a similarity score, which we will come back to later. The other factors are aiming at favouring high scores for sentences with similar (α) and large (β) lengths. In our implementations, they are computed as

$$\alpha = 1 - \frac{\text{abs}(|s| - |t|)}{\max(|s|, |t|)}, \tag{5.11}$$

$$\beta = \frac{\min(|s|, |t|)}{\lambda}, \tag{5.12}$$

where 'abs' is the absolute value, $|s|$ is the length in words of sentence s and λ is an integer constant representing the length threshold from which we consider a sentence to be very long ($\lambda = 100$ in our implementation, but it can be chosen depending on the given corpora).

The similarity score (sim) from Eq. (5.10) is calculated according to the formula:

$$\text{sim} = \frac{2 \times \text{teFound} \times \text{te}}{|s| + |t|} \times \frac{1}{\sqrt{\text{coh}}}, \tag{5.13}$$

where 'teFound' is the total number of words in s for which we found translation equivalents in t, coh is the *cohesion score* computed as the average distance between the sorted positions of the translation equivalents found in t (the lower the better)[7] and te is calculated as

$$\text{te}(s, t) = \sum_{w_i \in s} \max_{w_j \in t} \text{dicScore}\left(w_i, w_j\right), \tag{5.14}$$

where dicScore is the translation probability score from the dictionary. The rationale behind Eq. (5.14) is induced by the assumption that a word w_i is translated by only one word w_j and so, dicScore(w_i,w_j) $\geq$ dicScore(w_i,w_k) for any w_k in t.

One should note that since we aim at collecting parallel data that is not already in the dictionary that we started with, we are more interested in finding long parallel sentences. It is more probable that such sentences would contain unknown parallel fragments beside already known translations.

5.4.1.4 The PEXACC Translation Similarity Measure

The binary classifier of Munteanu and Marcu (2005) associates a confidence probability with its decision, but when setting this confidence at 0.5 or 0.7 as they do, it is equivalent to saying that sentence pairs with a score below the confidence level are not interesting for SMT.[8] Our view is that all sentence pairs that actually improve the output of an SMT system are important, and we found that these range from parallel and quasi-parallel to strongly comparable.

[7] We experimented with different power values for the cohesion score. We had the best results with ½ (the square root).

[8] But we acknowledge the fact that the probability of a sentence pair being parallel as computed by the classifier of Munteanu and Marcu is a proper model of parallelism.

We modelled our translation similarity measure as a weighted sum of feature functions that indicate if the source piece of text is translated by the target. Given two sentences s in the source language and t in the target language, then the translation similarity measure $P(s,t)$ is

$$P(s,t) = \sum_i \theta_i f_i(s,t), \tag{5.15}$$

such that $\sum_i \theta_i = 1$. Each feature function $f_i(s,t)$ will return a real value between 0 (s and t are not related at all) and 1 (t is a translation of s) and contributes to the overall parallelism score with a specific fraction θ_i that is language-pair dependent and that will be automatically determined by training a logistic regression classifier on existing parallel data (see next section).

Each of the feature functions $f_i(s,t)$ has been designed to return a value close to 1 on parallel s and t by manually inspecting a fair amount of parallel examples in the English–Romanian pair of languages. By negation, we assume that the same feature functions will return a value close to 0 for non-parallel unrelated s and t, but this behaviour is critically influenced by the quality and completeness of the linguistic computational resources that we use: bilingual translation lexicons, lists of inflectional suffixes used for stemming and lists of stop-words. Thus, generally, a feature function that uses one (or more) of the resources mentioned above can falsely return a value close to 0 for parallel s and t, due to the fact that this decision was made in the absence of the relevant entries in that resource. The prototypical example here is that the translation lexicon does not contain the relevant translations for the words in s.

Features

Before being processed, sentences s and t are tokenised, functional words are identified and content words are stemmed using language-dependent inflectional suffixes. Given these transformations of s and t, all features $f_i(s,t)$ are language independent. We use 5 features.

$f_1(s,t)$ is the '**content words translation strength**' feature. Given a statistical translation dictionary obtained by, for example, applying GIZA++ on a parallel corpus,[9] we find the best 1:1 alignment A between content words in s and t such that the translation probability[10] is maximised. If $\left\langle cw_i^s, cw_j^t \right\rangle$ is a word pair from $A, p\left(\left\langle cw_i^s, cw_j^t \right\rangle\right)$ is the translation probability of the word pair from the dictionary, and $|s|$ is the length (in content words) of sentence s, then

[9]To obtain the dictionaries mentioned throughout this subsection, we have applied GIZA++ on the JRC Acquis corpus (Steinberger et al. 2006).

[10]For two source and target words, if the pair is not in the dictionary, we use a 0 to 1 normalised version of the Levenshtein distance in order to assign a 'translation probability' based on string similarity alone. If the source and target words are similar above a certain threshold (experimentally set to 0.7), we consider them to be translations.

$$f_1(s,t) = \frac{\sum_{\langle cw_i^s, cw_j^t \rangle \in A} p\left(\left\langle cw_i^s, cw_j^t \right\rangle\right)}{|s|}. \tag{5.16}$$

This feature has a maximum value of 1 if all content words from s are translated in t with the maximum probability of 1.

$f_2(s,t)$ is the '**functional words translation strength**' feature. The intuition is that functional words around content words aligned as in feature $f_1(s,t)$ will also align for parallel s and t because of the fact that, from a dependency-syntactic point of view, functional words (prepositions, determiners, articles, particles, etc.) are usually governed by or govern nearby content words. Mathematically, if $\langle fw_k^s, fw_l^t \rangle$ is the highest scored pair of aligned functional words near (in a window of ± 3 words) the aligned pair of content words $\left\langle cw_i^s, cw_j^t \right\rangle$ from A, $|A|$ is the cardinal of the best alignment as found by $f_1(s,t)$, and $p(\langle fw_k^s, fw_l^t \rangle)$ is the probability of the functional word pair from the dictionary, then

$$f_2(s,t) = \frac{\sum_{\langle cw_i^s, cw_j^t \rangle \in A} p(\langle fw_k^s, cw_l^t \rangle)}{|A|}. \tag{5.17}$$

The maximal value of $f_2(s,t)$ is 1, and it is reached when, for each aligned pair of content words from A, there is a pair of functional words that align with the maximum probability of 1.

$f_3(s,t)$ is the '**alignment obliqueness**' feature (Tufiş et al. 2006). Here we have redefined it to be a discounted correlation measure because there are pairs of languages for which the natural word order implies crossing word alignment links. $f_3(s,t)$ also uses the alignment set A of content words described for feature $f_1(s,t)$ from which we derive two source and target vectors x^s and x^t of the same length containing the indices i in the ascending order ($1 \leq i \leq |s|$) and j respectively ($1 \leq j \leq |t|$) of content words cw_i^s and cw_j^t that form an alignment pair in A. Alignment obliqueness is computed as

$$f_3(s,t) = \text{abs}\left(\rho_{x^s,x^t}\right) \frac{1}{1 + e^{-10\frac{|A|}{\min(|s|,|t|)}+5}}, \tag{5.18}$$

where ρ_{x^s,x^t} is the Pearson correlation coefficient of the x^s and x^t vectors and 'abs(x)' is the absolute value function. The second term is a modified sigmoid function $f(x) = \frac{1}{1+e^{-10x+5}}$ designed to be a discount factor with values between 0 and 1 when x takes on values between 0 and 1. The rather steep variation of $f(x)$ was experimentally modelled in order to heavily discount 'rare' alignments for which the Pearson correlation is high. Thus, if A contains only a few alignments relative to $\min(|s|, |t|)$

(the size of A is at most $\min(|s|, |t|)$), then, even if ρ_{x^s,x^t} is high, $f_3(s,t)$ should be small, because a few alignments usually do not indicate parallelism.

$f_4(s,t)$ is the '**strong translation sentinels**' feature. Intuitively, if sentences s and t are parallel, then, frequently (at least in our studied examples), one can find content words that align near the beginning and end of the considered sentences. $f_4(s,t)$ is a binary-valued feature, that is 1 if we can find 'strong' translation pairs (probability greater than 0.2; set experimentally) between the first 2 content words at the beginning of s and t and between the last 2 content words at the end of s and t. Otherwise, $f_4(s,t)$ is 0.

Finally, $f_5(s,t)$ is the '**end with the same punctuation**' feature. This is also a binary-valued feature, that is 1 if both s and t end with the same type of punctuation: period, exclamation mark, etc. It is also 1 if both s and t lack final punctuation. Otherwise, $f_5(s,t)$ is 0.

The observant reader has noticed by now that all the features with the exception of $f_5(s,t)$ are not symmetrical, because they all depend on the alignment A computed for $f_1(s,t)$, which is not symmetrical and, as such, the measure from Eq. (5.6) is not symmetrical as well. In order to have evidence from both directions, we will use the arithmetic mean to get the final measure:

$$M(s,t) = M(t,s) = \frac{P(s,t) + P(t,s)}{2} \tag{5.19}$$

Learning the Optimal Weights

The weights θ_2 and θ_3 corresponding to the features 'functional words translation strength' and 'alignment obliqueness' are language-pair dependent because of the specific word ordering of the source and target languages. At the same time, θ_1 through θ_4 have to be optimised with respect to the translation lexicon in use, since the construction of the word alignments is based on this dictionary. Also, since $P(s,t)$ is not symmetrical, we will have to learn different θ_i weights from source to target and vice versa.

In order to derive a set of optimal weights for each language pair and translation lexicon, we have trained a standard logistic regression classifier. Briefly, the logistic regression classifier learns the θ_i weights that define the hyperplane, whose equation is the same as Eq. (5.15), that best separates the positive training examples from the negative ones. In our case, the examples are the multidimensional points whose coordinates are given by the feature functions $f_i(s,t)$.

For each language pair, the training set consists of 9500 parallel sentences[11] for the positive examples and 9500 non-parallel sentences (obtained from the parallel pairs by random shuffling) for the negative examples. For the training set in question, we also have 500 additional parallel sentences together with 500 non-parallel sentences (obtained by random shuffling as well) as the test set.

[11]Mostly from the News domain for all language pairs.

Table 5.5 Optimal weights for the translation similarity measure (TSM)

Lang.	θ_1	θ_2	θ_3	θ_4	θ_5	F1/BL
en–ro	0.31	0.02	0.37	0.21	0.09	0.93/0.88
ro–en	0.31	0.01	0.37	0.20	0.11	0.93/0.91
en–de	0.31	0.02	0.3	0.17	0.2	0.94/0.89
de–en	0.35	0.02	0.28	0.16	0.19	0.96/0.92
en–sl	0.23	0.01	0.38	0.2	0.18	0.96/0.89
sl–en	0.2	0.03	0.38	0.19	0.2	0.94/0.89
en–el	0.61	0.08	0.21	0	0.1	0.99/0.98
el–en	0.47	0.08	0.28	0.07	0.1	0.98/0.98
en–lv	0.27	0.05	0.41	0.16	0.1	0.98/0.96
lv–en	0.49	0.03	0.41	0	0.07	0.99/0.96
en–lt	0.33	0.01	0.41	0.15	0.1	0.96/0.91
lt–en	0.28	0.01	0.41	0.15	0.15	0.94/0.90
en–et	0.28	0.08	0.36	0.17	0.11	0.98/0.96
et–en	0.27	0.07	0.38	0.18	0.1	0.96/0.93
en–hr	0.29	0.01	0.41	0.16	0.13	0.98/0.95
hr–en	0.25	0.02	0.44	0.17	0.12	0.98/0.97

An example[12] is obtained by computing all the feature functions $f_i(s,t)$ for the given positive (parallel) or negative (non-parallel) s and t.

Table 5.5 summarises the derived optimal weights for 8 language-pairs, in both directions. In every pair, one language is English (en) and the others are: Croatian (hr), Estonian (et), German (de), Greek (el), Lithuanian (lt), Latvian (lv), Romanian (ro) and Slovene (sl).

The column named 'F1/BL' indicates the gain in $F1$ measure when testing the translation similarity measure with the optimal weights on the test set as compared to a baseline (BL) consisting of applying the measure using fixed values of the weights corresponding to our intuition of their importance: $\theta_1 = 0.45$, $\theta_2 = 0.2$, $\theta_3 = 0.15$, $\theta_4 = 0.15$, $\theta_5 = 0.05$. For instance we imagined that the content words translation strength feature $f_1(s,t)$ is much more important compared to the rest of the features, but the training procedure proved us wrong.

5.4.1.5 Evaluations

Experimental Setting

We evaluated our approach on 7 pairs of languages under the framework of the ACCURAT project.[13] For each pair, the source language is English (en), while the target languages are Estonian (et), German (de), Greek (el), Lithuanian (lt), Latvian (lv), Romanian (ro) and Slovene (sl). In order to compute precision and recall when mining for parallel sentences, we have devised artificial comparable corpora for all

[12]When an example occurs multiple times with both labels, we retain all the occurrences of the example with the most frequent label and remove all the conflicting occurrences.

[13]http://www.accurat-project.eu/

mentioned language pairs, with different levels of controlled comparability. Starting from 100 news parallel sentences for all language pairs, the corpora were created by injecting noise (in specific proportions) extracted from the News corpora collected in the ACCURAT project. We experimented with 4 different amounts of noise: 2:1,[14] 5:1, 10:1, 100:1, corresponding to different degrees of comparability, from strongly comparable to weakly comparable. The worst-case scenario is by far the one with 100:1 noise, and so, most of our experiments were developed under this setting.

We evaluated the efficiency of LEXACC after each of its steps: (1) the extraction of translation pair candidates using the search engine, (2) candidate pair filtering and (3) the usage of the translation similarity measure. Moreover, we evaluated the impact of the extracted data when used for improving SMT translation models.

Search Engine Efficiency

To measure the efficiency of using the search engine for finding translation candidates in the worst-case scenario (100:1 noise ratio), we computed the recall that we would obtain if we would have kept the best 100 hits (target sentences) returned by the engine for each source sentence. Instead of brute force analysing 10, 100^2 sentence pairs, we can now look at only 1 million pairs. This means a search space reduction of about 100 times. Table 5.6 shows that this approach is effective for most of the language pairs, but poor for en–el and en–ro. One of the reasons might be the quality of the dictionaries we relied on when generating the search engine queries.

Filtering Efficiency

As already mentioned, filtering is an intermediary step designed to further reduce the search space used for the final analysis. The filtering module receives high scores for speed and search space reduction for all language pairs. However, in terms of preserving the recall upper boundary, it performs well only for en–lv and en–de and acceptably for en–ro and en–el. It loses about 40% recall for the other three language pairs. Table 5.7 summarises the results.

Table 5.6 Recall upper boundary (UB) and size (sentence pairs and disk space occupied) for the translation candidates returned by Lucene

Pair	Recall UB	Data size (pairs/disk size)
en–de	0.98	1,009,500/323 Mb
en–el	0.42	1,009,700/485 Mb
en–et	0.89	1,008,800/345 Mb
en–lt	0.93	1,008,200/350 Mb
en–lv	0.92	1,008,300/366 Mb
en–ro	0.69	1,009,800/294 Mb
en–sl	0.80	688,266/191 Mb

[14]For each parallel sentence, 2 noise sentences were added.

Translation Similarity Efficiency

We evaluated the efficiency of the Translation Similarity Measure (TSM) from Sect. 5.4.1.4 by comparing it with the MaxEntclassifier by Munteanu and Marcu (2005) on English–German (en–de) document pairs with different levels of comparability (2:1 noise ratio, 5:1 and 10:1; see section 'Experimental Setting'). For both TSM and MaxEnt (with the associated confidence score for the 'parallel' label), we took into account all possible thresholds with a granularity of 0.01 above which the candidate pairs are considered parallel. We report the results corresponding to the threshold that maximises $F1$ for TSM and $F1$ for MaxEnt (thresholds are not the same). We explored three possible scenarios. The first one (Table 5.8) is to compute TSM for all possible sentence pairs.

The second scenario (Table 5.9) is to compute TSM only for the candidate pairs proposed by the search engine, without filtering.

The third scenario (Table 5.10) is similar to the second one, only this time, we use filtering.

For strongly comparable corpora (with less noise, like the 2:1 corpus), the filtering step in fact worsens the results. This is something to be expected because the filtering step eliminates a large proportion of the candidate pairs returned by the engine. Thus, filtering should be used only for weakly comparable corpora.

In order to make things more clear, we performed yet another experiment, this time for 100:1 noise ratio, which corresponds to a very weakly comparable corpus. In this setting, taking into account all possible sentence pairs as candidate pairs would result in a huge running time, and so, we were able to compare only the results obtained by LEXACC with and without filtering (Table 5.11).

We can see that for weakly comparable corpora, at the same threshold (0.41), filtering gets rid of a lot of noise, keeping the precision high (compare 0.8 with 0.101) at a modest decrease of the recall (compare 0.64 with 0.71).

Table 5.7 Recall upper boundary and size after the filtering step

Pair	Recall UB	Recall loss (%)	Size (pairs/Disk size)	Search space drop (%)
en–de	0.83	15.30	20,868/10 Mb	97.93
en–el	0.30	28.57	108,629/69 Mb	89.24
en–et	0.54	39.32	34,051/22 Mb	96.62
en–lt	0.57	38.70	35,831/21 Mb	96.44
en–lv	0.83	9.78	91,305/45 Mb	90.94
en–ro	0.53	23.18	160,968/67 Mb	84.05
en–sl	0.44	45	65,191/28 Mb	90.52

Table 5.8 en–de comparison between the MaxEnt classifier (ME) and the TSM when applied individually onto all possible sentence pairs

	2:1	5:1		10:1		
	ME	TSM	ME	TSM	ME	TSM
P	0.800	0.791	0.789	0.760	0.523	0.724
R	0.560	0.760	0.450	0.700	0.450	0.630
$F1$	0.658	0.775	0.573	0.729	0.483	0.673

Table 5.9 en–de comparison between the MaxEnt classifier and LEXACC with no filtering

	2:1	5:1		10:1		
	ME	LEX	ME	LEX	ME	LEX
P	0.800	0.717	0.789	0.650	0.523	0.618
R	0.560	0.710	0.450	0.650	0.450	0.600
*F*1	0.658	0.713	0.573	0.650	0.483	0.609

Table 5.10 en–de comparison between the MaxEnt classifier and LEXACC with filtering

	2:1	5:1		10:1		
	ME	LEX	ME	LEX	ME	LEX
P	0.800	0.809	0.789	0.737	0.523	0.742
R	0.560	0.340	0.450	0.450	0.450	0.520
*F*1	0.658	0.478	0.573	0.559	0.483	0.611

Table 5.11 en–de comparison between LEXACC with and without filtering for 100:1 noise

	LEXACC no filtering		LEXACC with filtering
	Best	Same *T*	Best
P	0.327	0.101	0.800
R	0.370	0.710	0.640
*F*1	0.347	0.177	0.711
Threshold	0.59	0.41	0.41
Running time	49.72 minutes		5.53 minutes

Same *T*: results obtained without filtering for the threshold yielding the best results with filtering (0.41)

Table 5.12 shows the accuracy of LEXACC when running on the 100:1 noise ratio comparable corpora. The running times depend on the sentence lengths and the size of the dictionaries.

SMT Experiments

To test the quality of the data extracted by LEXACC, we ran a few experiments with domain-adapted SMT in the automotive industry domain. We manually created a parallel corpus from an English–German comparable corpus of about 3.5 million sentences per language collected from the Web. The results of the experiments with the LEXACC-extracted data were compared to the same experiments conducted with the manually extracted parallel data in order to examine and compare the influence of the LEXACC-extracted data. Table 5.13 shows the statistics on the sentence pairs and sentence counts in the parallel and LEXACC-extracted data.

We compared three systems in our experiments: the 'Baseline' system, which was trained only on the Europarl (Koehn 2005) and News Commentary corpus (NC),[15] 'Automotive.parallel', which added only the parallel data to the baseline and the 'Automotive.extracted', which added only the LEXACC-extracted data to the baseline. All resulting corpora were aligned using GIZA++, and the MT systems were

[15] http://www.statmt.org/wmt11/translation-task.html

Table 5.12 LEXACC (with filtering) run on the 100:1 noise ratio comparable corpora

Pair	*P*	*R*	*F*1	Thr.	Minutes
en–de	0.800	0.64	0.711	0.41	5.53
en–el	0.550	0.22	0.314	0.35	27.24
en–et	0.284	0.23	0.254	0.34	7.11
en–lt	0.398	0.41	0.403	0.39	8.24
en–lv	0.357	0.50	0.416	0.51	11.75
en–ro	0.473	0.27	0.343	0.65	37.33
en–sl	0.219	0.16	0.185	0.34	7.75

trained using the Moses SMT Toolkit (Koehn et al. 2007). The languages' models were trained using SRILM (Stolcke 2002).

The Baseline system only uses Europarl (EP) for both the translation and the language model, but, for the two adapted systems, we used an additional language model trained on the domain-specific texts. Tuning via minimum error rate training (MERT) (Och 2003) was performed for all systems on a domain-specific development set; testing also used text from the automotive domain. The translations were evaluated using BLEU (Papineni et al. 2002).

As Table 5.14 shows, it is possible to gain about 6.5 BLEU points over the baseline system with the LEXACC-extracted data. The parallel data outperforms LEXACC, which may be due to the fact that the parallel data includes more unique sentences (see Table 5.13). Although only approx. 30% of the available unique data was extracted, an increase of 6.5 BLEU points is recorded—more than half of the increase achieved with the full parallel data. This means that LEXACC is able to discover salient parallel data that brings significant gains in BLUE score despite its size.

Another area of interest is how the extracted parallel and strongly comparable data compares to clean parallel data. In the LEXACC-extracted data, every German sentence is linked to 3.5 English sentences on average. To examine the effect of this noise, we retrained 'Automotive.parallel' with increasing amounts of data. Table 5.15 shows that the extracted data corresponds to more than 15k of parallel data in terms of BLEU improvement (compare with Table 5.14).

The data that LEXACC extracts is of high enough quality to be useful for SMT purposes, as the noise is filtered out during the training phase.

Table 5.13 Statistics on parallel and extracted data

Data	#pairs	# unique sent. (de/en)
parallel	44,482	42,396/44,290
extracted	45,952	12,718/13,306

Table 5.14 BLEU scores

System	BLEU%
Baseline	18.81
Automotive.parallel	30.25
Automotive.extracted	25.44

Table 5.15 Experiments with adding data

System	Training Data	BLEU score (%)
Baseline	EP+NC	18.81
Automotive.5k	EP+NC+5k Automotive	22.02
Automotive.10k	EP+NC+10k Automotive	23.36
Automotive.15k	EP+NC+15k Automotive	24.98
Automotive.20k	EP+NC+20k Automotive	26.48
Automotive.45k	EP+NC+full Automotive	30.25

5.4.2 PEXACC

PEXACC is a 'Parallel phrase EXtractor from Comparable Corpora' that requires, as LEXACC does, that documents in the comparable corpus are aligned. In order to assign a parallelism score to a pair of sentences coming from a pair of documents, it implements a trainable, language-independent translation similarity measure that was subsequently used by LEXACC. This translation similarity measure has been described in Sect. 5.4.1.4. The presentation below partially follows the one in Ion (2012).

5.4.2.1 The Algorithm

The general workflow of PEXACC is as follows (given a pair of source and target documents):

1. Split the input source and target documents into sentences and then, if desired, into smaller parts (loosely called 'phrases' throughout this presentation) according to a list of language-dependent markers. By a 'marker' we understand a specific functional word that usually indicates the beginning of a syntactic constituent or a clause. For English, these markers include prepositions, particles and negations (the infinitive 'to', 'not'), auxiliary and modal verbs ('have', 'be', 'can', 'must'), interrogative and relative pronouns, determiners and adverbs ('which', 'what', 'who', 'that', 'how', 'when', 'where', etc.) and subordinating conjunctions ('that', 'as', 'after', 'although', 'because', 'before', etc.). An important design decision is choosing a set of markers such that, for the source and the target languages, the phrases we obtain by splitting are in a 1:1 correspondence as much as possible. Thus, for Romanian, the same types of markers can be considered and, in most of the cases, the phrases would align 1:1. See the next

example pair of parallel sentences (the markers are underlined, square brackets indicate the phrases):

en: [A simple example] [will demonstrate the splitting] [of this sentence] [into smaller parts] .

ro: [Un exemplu elementar] [va demonstra împărţirea acestei propoziţii] [în părţi mai mici] .

we have the following correspondences: '[A simple example] ⇔ [Un exemplu elementar]' (1:1 correspondence), '[will demonstrate the splitting] [of this sentence] ⇔ [va demonstra împărţirea acestei propoziţii]' (2:1), and '[into smaller parts] ⇔ [în părţi mai mici]' (1:1).

2. Score **each possible pair of sentences/phrases** as to their parallelism degree by using Eq. (5.6) of Sect. 5.4.1.4. This is the main difference between LEXACC and PEXACC: while LEXACC applies the translation similarity measure only on a reduced set of translation candidate pairs, PEXACC exhaustively applies the same translation similarity measure on every sentence pair from a given document pair. While its recall is usually better, its processing time is much greater than that of LEXACC.
3. Output all pairs of sentences/phrases for which Eq. (5.6) of Sect. 5.4.1.4 gives a score larger than a pre-defined threshold (default to 0.2 but the real parallelism threshold is dependent on the type of the corpus: parallel, strongly comparable, and weakly comparable and on the values of the weights).

   The computations in the second step of PEXACC are independent of each other and as such may be executed in parallel. In its current implementation, PEXACC is able to spread the computation of the translation similarity measure for different sentence pairs over multiple CPUs in order to considerably shorten the overall computation time.

   Equation (5.6) of Sect. 5.4.1.4 makes use of several feature functions that are designed to indicate the parallelism of two sentences/phrases s and t. These functions are designed to return 1 when s and t are perfectly parallel (i.e. t has been obtained from s by translation if $s$and t were to be presented together as a pair to a human judge). The functions should return a value close to 0 when s and t are not related at all, but this behaviour is critically influenced by the quality and the completeness of the dictionary that is used. Thus, s and t may still be parallel, but if individual words in s do not have the relevant t translations in the dictionary and/or the translations probabilities are small, the resulting (low) score could be misleading. This is the main reason for which we have incorporated a '**relevance feedback loop**'[idea from Fung and Cheung (2004)]. Thus, steps 2–4 of the algorithm are executed for a fixed number of times and the 4th step of PEXACC.
4. Takes the output of step 3 and trains a supplementary GIZA++ (Gao and Vogel 2008) dictionary on all sentence/phrase pairs with a certain parallelism score (to minimise noise) and adds it to the main initial dictionary. The combination method between the main dictionary D and the learnt one T is as follows:

(a) If the pair of the word translation equivalents e is found in both D and the T dictionaries, its new translation probability $p(e)$ will be interpolated and becomes $p(e) = 0.7p_D(e) + 0.3p_T(e)$, where $p_D(e)$ is the probability of e in the D dictionary and $p_T(e)$ is the probability of e in the T dictionary.
(b) If the pair of translation equivalents e is found in either D or T but not both, its probability is left unchanged.

5.4.2.2 Evaluations

We tested PEXACC in several scenarios:

1. In order to measure the precision, recall and $F1$ on different types of comparable corpora, we artificially inserted noise (unrelated sentences) into a parallel corpus in specified proportions and checked the ability of PEXACC to re-discover the parallel corpus in the presence of noise; section 'Computing P, R, and $F1$' gives the details.
2. To compare PEXACC to current state-of-the-art parallel sentence mining from comparable corpora, we implemented Munteanu and Marcu's (2005) maximum entropy classifier (MaxEntClass) for English–German (Ion et al. 2011b) and ran the two algorithms on the artificially created comparable corpora created for the first experiment; we showed that the task of extracting parallel sentences from a comparable corpus is progressively more difficult as the comparable corpus type varies from strongly comparable to weakly comparable; section 'Comparison with the State of the Art' describes the experiment.
3. In order to see how PEXACC behaves on real-world data, we have run it on a real English–Romanian comparable News corpus collected in the ACCURAT project; section'Running PEXACC on Real-World Data' presents the results.

Computing *P*, *R* and *F*1

To be able to compute recall, we needed to know exactly how many parallel sentence pairs are present in the test comparable corpus. Since the collected comparable corpora are usually very large and cannot be evaluated by hand, we decided to 'pollute' an existing parallel corpus with 'noisy sentences'—sentences that are drawn from the same domain but are unrelated. The procedure and the test corpora are exactly the same as in the case of the LEXACC evaluation (see section 'Experimental Setting'): the proportion of noisy sentences versus parallel sentences was controlled and was set to 2:1, 5:1 and 10:1. That is for a noise proportion of 2:1, for each pair of parallel sentences, two pairs of noisy sentences were added.

We performed the tests on English–German (en–de), English–Romanian (en–ro), English–Greek (en–el) and English–Latvian (en–lv) in order to have a diverse language representation. The parallel test corpus for each language pair is a News corpus containing 100 parallel sentence pairs. There is a source file containing 100 sentences (one sentence per line) and a target file containing the parallel counterparts on the same line numbers. After source and target noise sentences are

added to each file (in the specified proportion), the ordering of the sentences is destroyed by random shuffling.

A first interesting experiment is to judge the performance of PEXACC with the optimised weights versus the default values of the weights (see section 'Learning the Optimal Weights'). We ran PEXACC with the default values and with the optimised values for the weights on the English–German comparable corpus, 2:1 noise ratio. Figures 5.1 and 5.2 plot the precision (P), and recall (R), $F1$ measure ($F1$) and $F0.2$ measure for the two runs, in percents, over the values of the translation similarity measure (step 0.01) as computed by Eq. (5.15) of Sect. 5.4.1.4.

The $F0.2$ measure is computed as

$$F_\beta = \left(1 + \beta^2\right)\frac{PR}{\beta^2 P + R},$$

where $\beta = 0.2$. $F0.2$ weighs precision more than recall. We have plotted $F0.2$, because we are more interested in precision than recall, as the main usage of PEXACC is for SMT training.

When running with the default weights, the best $F1$ measure is 75.55% and the best $F0.2$ measure is 85.22%. With the optimised weights, the best $F1$ is 77.55% (+2%) and the best $F0.2$ is 86.21% (+1%). Studying Figs. 5.1 and 5.2 comparatively, the area of the graphic delimited by the $F1$ and $F0.2$ curves is significantly larger in the case when the optimised weights are run. This translates directly into a better behaviour of P (rapid increase) and R (slower decrease) across the range of the translation similarity measure values.

Tables 5.16, 5.17 and 5.18 present the performance of PEXACC exhibiting the best $F1/F0.2$ measures as a function of Eq. (5.15) (Sect. 5.4.1.4) translation

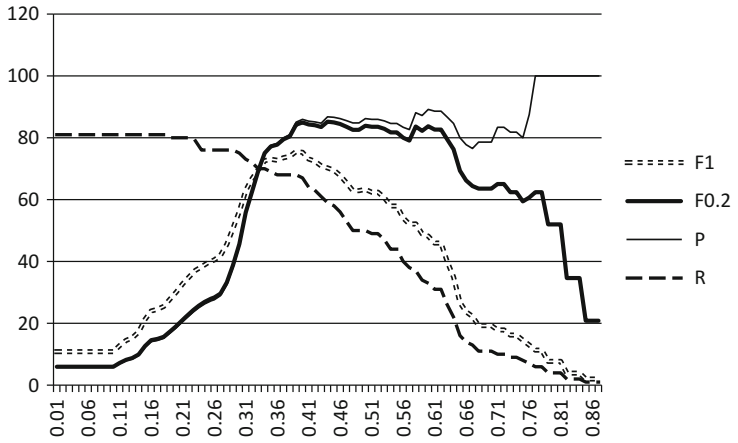


Fig. 5.1 P, R, $F1$, and $F0.2$ of PEXACC running with default weights

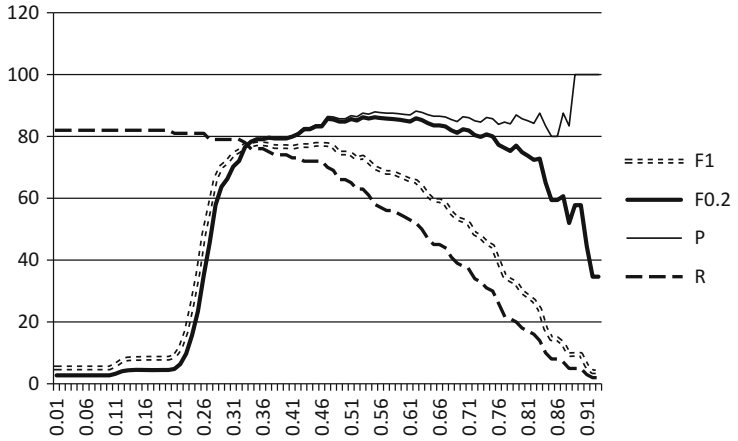


Fig. 5.2 *P*, *R*, *F*1, and *F*0.2 of PEXACC running with the optimised weights

similarity values. That is all the sentence pairs that are discovered by PEXACC are sorted in the decreasing order of the translation similarity measure, and we compute *P*, *R*, *F*1 and *F*0.2 for the top produced results up to the considered threshold (varied from 0 to 1 in steps of 0.01; see Fig. 5.2 for a graphical representation). We present the best values for *F*1 and *F*0.2.

Looking at these tables, we can see that the task of finding parallel sentences becomes harder with the increasing noise level in the comparable corpus. For instance, the English–Romanian *F*1 measure drops by 15.7% when processing a comparable corpus with noise ratio 10:1 compared to a comparable corpus with noise ratio of 2:1. Comparatively, the English–Greek *F*1 measure drops by only 7.7%, but the lower difference is explained by the fact that the noise was automatically introduced, and thus, the length ratio of the noisy sentence pairs has not been checked. As a consequence, in the case of English–Greek, some of the wrong pairs were filtered more easily.

The current implementation of PEXACC is in C# on .NET Framework 4.0. The processing time is dependent on the language pair, because the translation similarity measure computes, for each word, a type of transliteration in order to be able to compare and detect similar words in the source and target languages:

- For all languages, all diacritics are replaced with their diacritical mark free form.
- For Greek, a full transliteration is applied to get to the Latin alphabet.

Table 5.16 PEXACC run with optimised weights on the 2:1 noise ratio test set

	P	*R*	*F*1	*P*	*R*	*F*0.2
en–de	0.791	0.76	0.775	0.878	0.58	0.861
en–ro	0.684	0.78	0.728	1	0.38	0.94
en–el	0.864	0.83	0.846	0.971	0.67	0.954
en–lv	0.916	0.77	0.836	0.985	0.68	0.968

Table 5.17 PEXACC run with optimised weights on the 5:1 noise ratio test set

	P	*R*	*F*1	*P*	*R*	*F*0.2
en–de	0.76	0.7	0.729	0.862	0.5	0.838
en–ro	0.819	0.59	0.686	1	0.35	0.933
en–el	0.896	0.78	0.834	0.971	0.67	0.954
en–lv	0.88	0.74	0.804	0.947	0.54	0.92

Table 5.18 PEXACC run with optimised weights on the 10:1 noise ratio test set

	P	*R*	*F*1	*P*	*R*	*F*0.2
en–de	0.724	0.63	0.673	0.838	0.52	0.819
en–ro	0.814	0.44	0.571	0.916	0.33	0.858
en–el	0.789	0.75	0.769	0.944	0.51	0.914
en–lv	0.774	0.72	0.746	0.973	0.37	0.916

As a consequence, the number of processed sentence pairs varies with the language pair (as measured on a single core of an Intel i7 980 @ 3.33 GHz, 16 GB DDR3 @ 800 MHz):

- English–Romanian: 450 sentence pairs per second
- English–German: 500 sentence pairs per second
- English–Greek: 200 sentence pairs per second
- English–Latvian: 540 sentence pairs per second

As we already stated, PEXACC is able to distribute the computation on available CPU or CPU cores, and, as such, it is able to sustain the same sentence pair–processing rate on each of the 12 cores of the Intel i7 980 @ 3.33 GHz CPU on which we ran it. Thus, for instance, it finished (minus the time it needed to load the resources) the English–German processing of the 2:1 noise ratio text CC in approximately 15 seconds: 90,000 sentence pairs to process, 500 sentence pairs per second running on a single core, 12 CPU cores. This means that it is able to process $500 \times 12 = 6000$ sentence pairs per second on 12 cores, which gives $90{,}000/6000 = 15$ seconds of processing time in total.

Comparison with the State of the Art

In order to compare PEXACC with MaxEntClass, we ran both of them on the English–German comparable corpus constructed with the previously described methodology (see section 'Computing *P*, *R*, and *F*'). For both PEXACC and MaxEntClass (with the associated confidence score for the 'parallel' label), we took into account all possible thresholds with a granularity of 0.01, above which the candidate pairs are considered parallel. We report the results corresponding to the threshold that maximises *F*1 for PEXACC and *F*1 for MaxEntClass (thresholds are not the same) (Table 5.19).

We can see that the performance of both algorithms decreases significantly when asked to extract parallel sentences from weakly comparable corpora (noise ratio 10:1) as compared to strongly comparable corpora (noise ratio 2:1).

PEXACC has the advantage that its output can be trimmed by imposing a certain threshold on the value of the translation similarity measure as computed by

Table 5.19 Comparison between the MaxEntClass (MEC) and PEXACC (PXC) when applied on different types of English–German comparable corpora

	2:1		5:1		10:1	
	MEC	PXC	MEC	PXC	MEC	PXC
P	0.800	0.791	0.789	0.760	0.523	0.724
R	0.560	0.760	0.450	0.700	0.450	0.630
*F*1	0.658	0.775	0.573	0.729	0.483	0.673

Table 5.20 14-02-2012 News comparable corpus statistics

	Docs.	Sentences	Tokens	Size
en	17,845	464,961	9,309,338	53.7MB
ro	7120	121,104	2,605,976	16.9MB

Table 5.21 Accuracy of PEXACC on the first 7186 sentence pairs extracted from the 14-02-2012 News CC

Threshold	Precision	Sentence pairs
0.9	1	22
0.8	1	166
0.7	0.99	973
0.6	0.95	3267
0.5	0.92	7186

Eq. (5.15) of Sect. 5.4.1.4. Thus, above a certain threshold, the precision of PEXACC can even be as high as 1 at a significant cost of the recall (see Fig. 5.2). We consider this to be an asset of PEXACC, since, as already stated, its main use is to generate parallel data for SMT training. Furthermore, by carefully choosing a value for this threshold, the desired trade-off between precision and recall (e.g. measuring *F*0.2) may be achieved.

Running PEXACC on Real-World Data

PEXACC has been used to collect parallel sentence pairs for SMT training in the ACCURAT project. We present its evaluated precision on an English–Romanian weakly comparable news corpus that was collected in the project (version 14-02-2012) by continuously harvesting news articles from selected URLs. The documents in the corpus are aligned based on their titles and publication dates. Table 5.20 presents the corpus statistics.

We ran PEXACC on this corpus and kept all the sentence pairs with a translation similarity measure of at least 0.3, which are 22,352. We then sorted these pairs in **descending order** and inspected them by hand, from the pair with the largest score (0.99) until the last pair with a score larger than or equal to 0.5. Table 5.21 presents the results.

In Table 5.21, the 'Sentence pairs' column shows the number of sentence pairs that have a translation similarity score of at least the specified value in the 'Threshold' column. We cannot compute the recall, because we do not know how may parallel sentence pairs are in the corpus. We have inspected all produced pairs with a score of at least 0.7. Below this threshold, we did a random sampling of the results by selecting and evaluating 100 pairs from each threshold range (0.7–0.6 and 0.6–0.5).

Table 5.21 shows that PEXACC precision is consistent with the results reported in Tables 5.16 and 5.18.

5.5 Parallel Phrase Mining from Comparable Corpora

Parallel sentence mining from comparable corpora is successful when the comparable corpus actually contains translations of entire sentences, but there are types of comparable corpora (e.g. weakly comparable) in which such sentence translations are not available. When studying weakly comparable document pairs, one can easily spot sub-sentential fragments (from clauses to named entities or terminology) that are translated in the target part of the corpus. Thus, developing algorithms that will spot these sub-sentential (or phrasal) translations is a worthwhile enterprise, because the resulting parallel resource can be used in various ways to improve machine translation systems.

In what follows, we will present two algorithms that perform parallel phrase mining from a comparable corpus: a binary SVM classifier and PEXACC, which we have already introduced in Sect. 5.4.2. PEXACC can perform both parallel sentence and parallel phrase mining, but here we will focus on the phrasal capabilities.

5.5.1 Parallel Phrase Mining with SVM

We approach the task of parallel phrase extraction as a classification task and use feature extraction on the training data to train a support vector machines (SVMs) classifier to distinguish between parallel and non-parallel phrases. Our method is fully automatic and is essentially a 'generate and test' approach. In the generate phase, given source and target language sentences S and T, we first generate all possible phrases of a given length for S and for T and then compute all possible phrase pairings consisting of one phrase from S and one phrase from T. In the test phase, we use a binary SVM classifier to determine whether each generated phrase pair is parallel or not. The SVM classifier is trained using phrase pairs taken from parallel data that is word aligned using GIZA++.

We have tested our approach on the English–German, English–Greek and English–Latvian language pairs. Latvian is an under-resourced language, while text resources for Greek and German are more readily available. Considering all three languages allows us to directly compare our method's performance on resource-rich and under-resourced languages. We perform two different tests. First, we evaluate the performance of the classifier on phrases extracted from held-out parallel data using standard measures such as recall, precision and accuracy. Secondly, we test whether the phrases extracted by our method from comparable corpora lead to improved SMT quality, as measured using BLEU. The following presentation is based on the one from Aker et al. (2012b).

5.5.1.1 Phrase Pair Generation

Phrase pairs are generated under two different conditions. During training of the SVM phrase pair classifier, positive and negative instances of aligned phrase pairs are generated from existing parallel resources for the source and target languages. During testing, candidate phrase pairs are generated from arbitrary source and target language sentence pairs.

Training Example Extraction
We use whatever parallel data is available for a language pair to extract training examples for the SVM classifier. To get positive training examples (parallel phrases), we first align the parallel sentence pairs using the GIZA++ toolkit in both directions and then refine the alignments using a 'grow-diag-final-and' strategy. Then, we extract all phrases, as defined in the statistical machine translation literature (Koehn et al. 2003; Och and Ney 2004; Chiang 2005), and take these phrases as positive examples.

Let S denote a sentence, S_i the ith word in S, and S_i^j the subsequence of words in S from position i to j. Given a word-aligned sentence pair $\langle S, T\rangle$, $\left\langle S_i^j, T_{i'}^{j'}\right\rangle$ is a phrase if and only if

- S_k is aligned to T_k, for some $k \in [i,j]$ and $k' \in [i',j']$.
- S_k is not aligned to T_k, for all $k \in [i,j]$ and $k' \notin [i',j']$.
- S_k is not aligned to T_k, for all $k \notin [i,j]$ and $k' \in [i',j']$.

To get negative training examples (non-parallel phrases), for each sentence pair, we enumerate all segments on the source side and on the target side, the length of which falls in the range [minSrcLen ... maxSrcLen] and [minTrgLen ... maxTrgLen], respectively. Then, we pair each source segment with each target segment to get all possible training examples. Next, we leave out the positive examples and label the rest as negative examples.

A training example may be discovered many times during the extraction process. We do not keep duplicate occurrences but keep all unique training examples. As the alignment of the parallel corpus inevitably introduces some errors, we do some processing to remove the noise. For instance a training example may appear both as a positive example and as a negative example, but, in our approach, a training example can only have one label, that is positive or negative. For a training example, let's assume that the number of occurrences as a positive example is N_p and the number of occurrences as a negative example is N_n. We check the following conditions in order:

- If N_p is smaller than a count threshold τ, then we label this example as negative.
- If the ratio $\frac{N_n}{N_p}$ is below a ratio threshold π, then we label it as positive.

Test Instance Generation
To generate candidate parallel phrase pairs from unseen comparable text pairs, we proceed as follows. First, we generate all sentence pairs $\langle S, T\rangle$ where S is from the

source language text and T is from the target language text. Then, for each such pair, we generate all phrase pairs $\langle s, t \rangle$ where s is a word subsequence of S of length i, minSrcLen $\leq i \leq$ maxSrcLen and t is a word subsequence of T of length j, minTrgLen $\leq j \leq$ maxTrgLen.

5.5.1.2 SVM Classifier

For classifying phrase pairs as parallel or non-parallel, we use an SVM classifier. Within the classifier, we use the following features as reported in previous work (Munteanu and Marcu 2005; Hewavitharana and Vogel 2011):

- ***lengthDifferenceInChar*** is the difference in the number of characters in the source and target phrases. We consider duplicates in the phrases when counting the characters.
- ***lengthDifferenceInWords*** is similar to the first feature but uses words instead of characters.
- ***sameEnding*** is 1 if source and target phrase have the same ending; otherwise, it is 0.
- ***numberOfWordsInPhrase*** is the number of words in the source phrase.
- ***firstWordTranslationScore*** indicates whether the first word in the source phrase is a translation of the first word in the target phrase. If this is the case, the translation probability is returned.
- ***lastWordTranslationScore*** indicates whether the last word in the source phrase is a translation of the last word in the target phrase. If this is the case, the translation probability is returned.
- ***translationCount*** is the number of source phrase words that have translations in the target one.
- ***translationRatio*** is the ratio of the count of source phrase words that have translations in the target phrase and the number of words in the source language.
- ***isHalfTranslated*** is 1 if at least half of the source phrase words have translations in the target phrase; otherwise, it is 0.
- ***longestTranslatedUnit*** is the count of words within the longest sequence of words that have all translations in the target phrase.
- ***longestNotTranslatedUnit*** is similar to the previous feature but considers words that do not have translations.
- ***translationPositionDistance*** captures the distance between the positions of source words and the position of their maximum likely translations in the target side. For example if the first word in the source phrase is the translation of the first word in the target phrase, then they have a translation position distance of 0. For each word in the source phrase, we compute its translation position distance, sum all the distances together and return it.

The first three features are independent of which language is taken as source and which as target. The feature *numberOfWordsInPhrase* is computed once for the source and once for the target phrase. The remaining nine features are direction

dependent and are computed in both directions, reversing which language is taken as the source and which as the target. Thus, in total, we have 21 features. To perform the translation of phrase words, we use GIZA++ dictionaries trained on parallel data (see section 'Phrase Extraction for Classifier Training and Testing').

Cognate-Based Methods for Translation Purposes

Dictionaries mostly fail to return translation entries for named entities (NEs) or specialised terminology. Because of this, we also use cognate-based methods to perform the mapping between source and target words or vice versa. We only apply the cognate-based methods for the *firstWordTranslationScore* and *lastWordTranslationScore* features. For these two features, it is easy to compare the first or the last words from both the source and target phrases. The score of the cognate methods becomes the translation score for the features. We adopt several string similarity measures that are described in Aswani and Gaizauskas (2010): (1) longest common subsequence ratio, (2) longest common substring, (3) dice similarity, (4) Needleman–Wunsch distance and (5) Levenshtein distance. Each of these measures returns a score between 0 and 1. We use a weighted linear combination of the scores to compute the final score. We learn the weights using linear regression over training data consisting of pairs of truly and falsely aligned city names available from Wikipedia.[16] For the truly aligned named entities, we assign a score of 1 and, for the falsely aligned ones, a score of 0. We take the cognate similarity score as the translation score only if it is above 0.7, a threshold that we set experimentally. The cognate methods assume that the source and target language strings being compared are drawn from the same character set. However, this is not the case for English and Greek. To be able to apply our cognate-based approach to Greek, we first map the Greek characters into English characters and apply the cognate metrics on the mapped characters. To learn the mappings, we used a list of Greek–English place name variants[17] and the GIZA++ tool. The input to GIZA++ is a list of aligned NEs (Greek and English) where each NE is split into single characters. The output of the tool is a dictionary with character mappings. We use these mappings to transliterate a Greek word into English characters and use the transliterated version for the cognate comparison. Note, since GIZA++ lists multiple entries as translation variants, we always select the one with the highest probability value.

5.5.1.3 Experiments

Data Sources

Our experiments involve the English–Greek (en–el), English–Latvian (en–lv) and English–German (en–de) language pairs. We train a separate classifier for each language pair. Therefore, for each language pair, a dataset consisting of parallel

[16] http://en.wikipedia.org/wiki/Names_of_European_cities_in_different_languages

[17] http://en.wikipedia.org/wiki/List_of_Greek_place_names

Table 5.22 Size of comparable corpora

Language pair	Document pairs	en sentences	Target sentences	en words	Target words
en–de	66K	623K	533K	14837K	6769K
en–el	122K	1600K	313K	27300K	8258K
en–lv	87K	1122K	285K	18704K	5356K

phrases is needed to train and test the SVM classifier. A second data source that is needed for our experiments is comparable corpora for the above-mentioned language pairs. From these, we generate pairs of phrases and judge them for parallelism using the trained classifier. Finally, the phrases judged as parallel by the classifier are used to attempt to improve a baseline SMT system.

Parallel Corpora

We used the JRC-Acquis parallel corpora (Steinberger et al. 2006) to prepare the parallel phrases used to train and test the SVM classifier. For each language pair, we split the corpus into two parts: a training set and a test set. The test set contains 10K parallel sentences. The training set contains 99K sentences for en–de, 423K for en–el and 53K sentences for en–lv.

Comparable Corpora

We used comparable corpora in English–Greek, English–Latvian and English–German language pairs. These corpora were collected from news articles using a lightweight approach that only compares titles and date of publication of two articles to judge them for comparability (Aker et al. 2012a). The corpora are aligned at the document level and are detailed in Table 5.22.

Phrase Extraction for Classifier Training and Testing
On both parallel training and testing datasets (see section 'Data Sources'), we separately applied GIZA++ to obtain the word alignment information used in our parallel phrase extraction method (see section 'Computing P, R, and $F1$'). Then, we ran the training example extraction method on each dataset to extract phrase pairs, setting minSrcLen = minTrgLen = 2 and maxSrcLen = maxTrgLen = 7. To train the classifier, we used 20K parallel and 20K non-parallel phrase pairs extracted from the training data. In testing, we used 500 parallel and 10K non-parallel phrase pairs extracted from the testing data. Note that the test set contains substantially more non-parallel than parallel data. This is to simulate the real-world scenario where the data from which parallel phrases have to be extracted will necessarily contain more non-parallel entries than parallel ones. It is also important to note that, in both the training and testing parallel phrase extraction steps, we used GIZA++ dictionaries obtained from the parallel training data, which excludes the 10K parallel sentences

Table 5.23 Phrase pairs extracted from comparable corpora

Language pair	Analysed sentence pairs	Analysed phrase pairs	Extracted phrase pairs
en–de	39659K	852327K	248K
en–el	33844K	1499169K	125K
en–lv	30788K	1919128K	106K

used in testing. We did this to ensure that feature extraction in testing is performed using a dictionary that has been built by a process that is blind to the test data.

Phrase Extraction from Comparable Corpora
We used the comparable corpora described in section 'Data Sources' and, for each language and each aligned document pair, extracted phrase pairs as described above in section 'Comparison with the State of the Art'. When generating training instances, we set minSrcLen = minTrgLen = 2 and maxSrcLen = maxTrgLen = 7. As in the training and testing steps described in the previous section, for feature extraction from the phrase pairs generated from the comparable corpora, we used the GIZA++ dictionary created from parallel sentences in the training data. Table 5.23 gives details about the phrases extracted from the comparable corpora.

We also ran a performance test to evaluate the speed of parallel phrase extraction. We took 1000 comparable document pairs from the en–de data and recorded the time it took to process them. We recorded ~44 minutes processing time on a single desktop machine with a 2.4 GHz processor and 4GB memory: 99% of the processing time was spent on feature extraction and the remaining 1% for phrase pairing and SVM classifier. Note that since the document pairs are independent of each other, multiple processes could be run in parallel on different sets of document pairs, which could significantly reduce processing time.

Results
To test the performance of our approach, we performed two different evaluations: classifier evaluation using Information Retrieval (IR) metrics and SMT performance using BLEU.

Classifier Evaluation

In this evaluation, we measure the performance of our classifier using precision, recall, *F*-measure and accuracy (Manning et al. 2008). Note that we use $F_{0.5}$, which puts more emphasis on precision than on recall. We sought to optimise SVM classifier performance for our task by finding the SVM-margin distance boundary that maximises $F_{0.5}$. During training, the SVM classifier determines a maximum margin hyperplane between the positive and negative examples. During classification, the distance to this boundary is used to classify instances: any instance that has negative distance (distance < 0) to the boundary is treated as a negative example; otherwise, it is treated as positive (distance ≥ 0). We shift the boundary between negative and positive examples to a new value that maximises the $F_{0.5}$ metric. To do

Table 5.24 Classifier's performance on phrases extracted from the test data

Language pair	Recall	Precision	$F_{0.5}$-measure	Accuracy
en–de	45	86	73	97
en–el	63	81	77	97
en–lv	59	84	77	97

this, we determine the maximal negative and maximal positive distance from the classification results, go from the negative value towards the maximal positive value in increments of 0.1 and record the boundary value that leads to the maximum $F_{0.5}$. To learn the new boundary, we used held-out training data containing 500 parallel and 10K non-parallel phrases. Note that this held-out training data is different from the testing data (see section 'Data Sources') but has the same size. Finally, we run the classifier with the new boundary on the testing data. The results are shown in Table 5.24.

From Table 5.24, we can see that the classifiers for each language pair perform reasonably well on the testing data. They all achieve an accuracy score above 97%, though note that always picking the majority class (non-parallel) gives 95% accuracy given the deliberate skew in the test data. The precision score obtained from each classifier is above 81%, showing good performance in identifying correct parallel phrases. In general, the recall scores are low, in the neighbourhood of 50%. However, given the potentially very large quantities of comparable text pairs available, recall is not a primary concern.

To identify the sources of misclassifications, we manually checked the en–de phrases from the test set that were classified incorrectly. The first source of problems is due to the existence of productive compounds in German and negatively affects recall. For example the classifier classifies the following parallel phrases as non-parallel. The features that we use within the classifier do not capture morphological elements within compound words and thus fail to match: for example *tiergesundheitszeugnisse* with *veterinary certificates* or *umweltkriterien* with *ecological criteria*.

1. *der tiergesundheitszeugnisse für die—veterinary certificates for the*
2. *zur festlegung überarbeiteter umweltkriterien—establishing revised ecological criteria*

   The second problem is due to feature extraction and causes a decrease in precision. The following phrases are non-parallel examples classified by the classifier as parallel. The reason for the misclassification is that while the words in the English phrase can be entirely mapped to those in the German phrase, the phrases are not parallel, because they differ either in the number or in the order of constituents.
3. *parlaments und des rates zur einführung—the council and the*
4. *die kommission erstattet dem europäischen parlament und—european parliament and of the council*

In example 3, all words of the English phrase have translations in the German phrase (both *the*'s are mapped to *des*, *council* is mapped to *rates* or *parlaments* and *and* is mapped to *und*). In example 4, we have a similar picture. The words *european parliament* are mapped to *europäischen parlament*, *and* to *und*, *the* to *die* or *dem* and *council* to *kommision*. The problem arises from the fact that, in example 3, the English word *council* translates into both German *Rat* and *Parlament*. Thus, two German noun phrases (NPs) are covered by one in English, so that the English phrase is not an adequate translation of the German one. In example 4, the problem lies in the order of the constituents, which results in the two phrases not being parallel. The English phrase contains a coordination of two NPs, while, in the German phrase, the coordinating conjunction *und* is at the end of the phrase and serves to link either the entire phrase or the second NP (*dem europäischen parlament*) to a further constituent not extracted as a part of this phrase.

BLEU Evaluation for SMT

In the BLEU evaluation, we tested the impact of the phrases extracted from the comparable corpora on improving the performance of the baseline SMT systems. We trained a baseline decoder for each language pair using the entire JRC-Acquis corpus for that language pair, which consists of the training and test data used for our phrase extraction system. We then injected the extracted phrases[18] into the baseline training data and re-trained a new decoder, which we call an extended decoder. As SMT test data, we used 612 parallel sentences manually generated from news articles. The English and German sentences both have 14K words in total. The Latvian sentences contain around 13K words, while the Greek sentences contain 15K words. To construct these test sets, we used English as the pivot language. We selected 612 English sentences from different news articles and then manually translated them into German, Greek and Latvian. For each language pair, a professional translator was hired to perform the translation. Note that these articles are not included in the comparable corpora summarised in Table 5.22.

From the results shown in Table 5.25, we can see that all extended decoders significantly outperform the baseline systems.[19] This shows that the phrases extracted from the comparable corpora are indeed of usable quality. In Table 5.25, we also see that the en–el BLEU scores are much higher than the others. We think that this is a result of the large size of the en–el parallel training data made available by JRC, which we used to train the en–el decoder. As described in section 'Data Sources', the en–el parallel corpus is more than 4 times bigger than the en–de corpus and 8 times bigger than the en–lv parallel corpus. For the language with the least training data, Latvian, the classifier still significantly outperforms the baseline. This

[18]These phrases are extracted with the SVM margin that maximises the *F*-measure, see the 'Classifier evaluation' subsection for details.

[19]Koehn (2004) reports that an increase of 1% in BLEU score is a significant improvement.

Table 5.25 BLEU scores on the SMT testing data

Language pair	BL BLEU score	Extended BLEU score
en–de	15.97	18.05
en–el	28.30	29.37
en–lv	10.24	12.23

is an encouraging result, which shows that though the amount of parallel data is important for SMT performance, our method for phrase extraction from comparable data provides a viable way to significantly improve SMT performance in cases where parallel data is sparse.

5.5.2 Parallel Phrase Mining with PEXACC

PEXACC has already been introduced in Sect. 5.4.2.1, where the segmentation of the source and target sentences into phrases has been described. Here, we will focus on the evaluation of the extracted parallel phrases from the artificially created comparable English–Romanian corpus described in section 'Computing *P*, *R*, and *F*1'. The assumption on which we based our entire evaluation process is that *if PEXACC has a specific (measurable) accuracy on a (random) pair of parallel documents, that accuracy should not significantly degrade if we introduce noise (in quantifiable ratios to the existing parallel data) in the source and target documents and randomly permute the sentences in each document*. To test that assumption, we needed to construct a Gold Standard (GS) of mapped phrases from a pair of (clean) parallel English–Romanian documents:

- Given a reference pair of parallel English–Romanian documents (tested with 100 sentences per document, randomly selected from different domains).
- Apply GIZA++ to obtain a word alignment from the source sentences to the target sentences.
- For each word-aligned source sentence and target sentence pair, break them using the PEXACC fragmentation routine (see Sect. 5.4.2.1) and align the resulting text fragments based on word alignments such that links of words from a source fragment do not point outside the boundaries of a target fragment.

For instance, given the English sentence 'In addition to schools and universities, the drive is on for libraries, museums and similar institutions ...' and the Romanian translation 'În plus față de școli și universități, se acționează pentru ca bibliotecile, muzeele și instituții similare . . .', Fig. 5.3 displays the PEXACC fragmentation style using dotted lines. Along with GIZA++ generated word alignments (see the arrows from the English words to the Romanian equivalents), we are able to automatically generate GS phrase mappings 'In addition' ⇔'În plus față', 'to schools' ⇔ 'de școli', 'and universities,' ⇔ 'și universități', etc. The quest is on then to apply PEXACC onto the same pair of parallel documents but with added noise (random

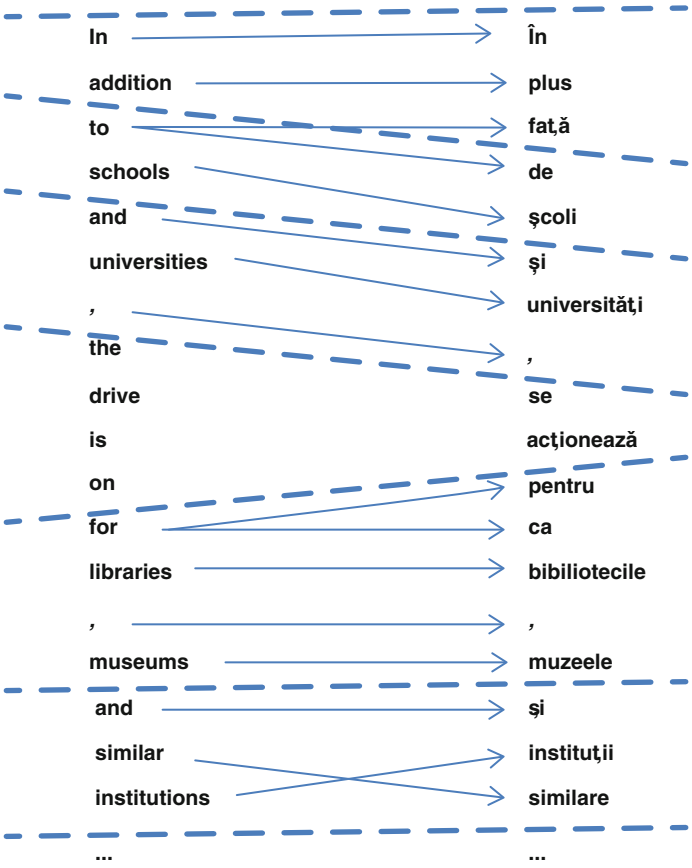


Fig. 5.3 PEXACC fragmentation example in English and Romanian. GS will contain pairs of phrases delimited by the dotted lines and supported by GIZA++ generated word alignments (drawn as arrows from the English words to the Romanian equivalents)

sentences added in the same proportion to the source document and the target document) and see to what extent we can cover the GS and with what precision we can generate parallel mapped phrases.

We have to note the following deficiencies of this automatically generated GS:

- Word-alignments generated by GIZA++ are not perfect, and, as such, there are correct phrase mappings that PEXACC finds but that are not present in the GS as the supporting word-alignments were missing/wrong.
- The GS was generated from a pair of parallel documents that are word-aligned at sentence level. But PEXACC may also find correct phrase mappings with phrases belonging to sentences that are not paired; these (correct) phrase mappings will obviously not be present in the GS. Thus, in order to compute a fair precision with respect to the given GS, we are going to consider, as the set of generated results,

Table 5.26 PEXACC performance on the parallel English–Romanian document pair, using a JRC Acquis-based GIZA++ extracted dictionary; 3 relevance feedback loops (the maximal values for each category *P*, *R* and *F* are bolded)

	Iteration 1		Iteration 2		Iteration 3	
0.1	*P*: 0.534	530	*P*: 0.549	532	*P*: 0.55	532
	R: 0.752		*R*: **0.761**		*R*: **0.761**	
	F: 0.624		*F*: 0.638		*F*: 0.639	
0.3	*P*: 0.634	429	*P*: 0.647	437	*P*: 0.645	439
	R: 0.708		*R*: 0.721		*R*: 0.721	
	F: 0.669		*F*: 0.682		*F*: 0.680	
0.5	*P*:**1**	228	*P*:**1**	239	*P*:**1**	241
	R: 0.509		*R*: 0.518		*R*: 0.526	
	F: 0.674		*F*: 0.682		*F*: **0.689**	

the set of all the source phrases[20] that PEXACC found. If for a given source phrase, there is a target phrase mapped to it such that the pair is found in GS, then we give a precision point and a recall point to PEXACC. In addition to that, we also experimentally observed that, for English–Romanian, all phrase pairs with a parallelism probability of over 0.5 are in fact correct even if they are not found in the GS.[21] In this case, we will also give PEXACC a precision point (but not a recall point) if the detected phrase pair has at least 0.5 as its parallelism probability (given by Eq. (5.6) of Sect. 5.4.1.4).

Tables 5.26 and 5.27 report on the base line performance of PEXACC: running on a pair of parallel clean documents that do not contain any added noise. Table 5.26 presents the run using an English–Romanian GIZA++ dictionary extracted from the JRC Acquis corpus (Steinberger et al. 2006), and Table 5.27 presents the same run but using a very large (over 9.5 million entries at wordform level) English to Romanian dictionary extracted from all our parallel corpora and enriched with a WordNet-based dictionary derived from the conceptual alignments between the Princeton WordNet (Fellbaum 1998) and the Romanian WordNet (Tufiş et al. 2008). There are three parallelism thresholds for which we computed the precision (*P*), the recall (*R*) and the *F*-measure (*F*) of the algorithm: 0.1, 0.3 and 0.5. After each phrase extraction phase (called 'an iteration'), a GIZA++ dictionary is trained on the output of the algorithm (considering all pairs of phrases with a parallelism probability of at least 0.5), and the resulting dictionary is incorporated into the main dictionary (see the 4th step of PEXACC in Sect. 5.4.2.1). Before 'Iteration 1', we have only the main dictionary. For each iteration, we also list the number of unique English–Romanian phrase pairs that PEXACC found, next to each of the *P*, *R* and *F* cell.

Studying Tables 5.26 and 5.27 comparatively, we can observe the following facts:

[20] And, if it is a set, no source phrase is repeated.

[21] The probability threshold over which all generated parallel pairs is correct is dependent on the type of document pairs. For the English-Romanian pair of parallel documents on which we tested, at least 0.5 is guaranteed to indicate perfect parallelism (we have determined that by manually inspecting the output).

Table 5.27 PEXACC performance on the parallel English–Romanian document pair, using a (very large) reference dictionary; three relevance feedback loops (the maximal values for each category *P*, *R* and *F* are bolded)

	Iteration 1		Iteration 2		Iteration 3	
0.1	*P*: 0.723	570	*P*: 0.713	571	*P*: 0.713	571
	R: **0.81**		*R*: **0.81**		*R*: **0.81**	
	F: 0.764		*F*: 0.758		*F*: 0.758	
0.3	*P*: 0.782	522	*P*: 0.766	526	*P*: 0.761	528
	R: 0.783		*R*: 0.788		*R*: 0.777	
	F: 0.782		*F*: 0.777		*F*: 0.77	
0.5	*P*:**1**	396	*P*:**1**	390	*P*:**1**	391
	R: 0.695		*R*: 0.672		*R*: 0.677	
	F: **0.82**		*F*: 0.804		*F*: 0.807	

- We can obviously improve the extraction accuracy by using a better (larger and more accurate) dictionary (see Table 5.27), but, in that case, training new dictionaries will not improve our subsequent extraction steps (in Table 5.27, the best result is obtained in the first iteration) due to the fact that the new translation equivalents pairs are very rare. This is the explanation of the fact that we cannot achieve 100% recall: no matter how large one dictionary is, it will always be incomplete with respect to new data. Figure 5.3 contains an example where the phrase 'the drive is on' is the equivalent of the Romanian 'se acționează'; the translation pair 'drive, acționează' is a new translation pair missing from our huge dictionary.
- On the other hand, training intermediary GIZA++ dictionaries certainly helps to discover new translation pairs (see Table 5.26 where better results are obtained with each new iteration) when using a rather small (just over 200,000 entries at wordform level) dictionary. Since we cannot rely on the existence of accurate and large dictionaries for every language pair, we need to adopt this 'extract, learn and loop' strategy.

Table 5.28 contains the results of running PEXACC on our pair of parallel documents to which we have added (to each individual document in fact) noise in proportion of 1:1, meaning that, for each existing sentence in the document, another random one was added. This noise addition modified the status of our document pair from 'parallel' to 'strongly comparable'. After the noise sentences were added, a

Table 5.28 PEXACC performance on the strongly comparable English–Romanian document pair (noise ratio 1:1), using a JRC Acquis-based GIZA++ extracted dictionary; three relevance feedback loops (the maximal values for each category *P*, *R* and *F* are bolded)

	Iteration 1		Iteration 2		Iteration 3	
0.1	*P*: 0.246	1110	*P*: 0.275	1118	*P*: 0.276	1118
	R: 0.73		*R*: **0.74**		*R*: **0.74**	
	F: 0.368		*F*: 0.401		*F*: 0.402	
0.3	*P*: 0.296	886	*P*: 0.325	918	*P*: 0.323	924
	R: 0.686		*R*: 0.7		*R*: 0.7	
	F: 0.413		*F*: 0.443		*F*: 0.442	
0.6	*P*: **1**	185	*P*: **1**	232	*P*:**1**	237
	R: 0.332		*R*: 0.393		*R*: 0.411	
	F: 0.498		*F*: 0.565		*F*: **0.583**	

Table 5.29 PEXACC performance on the strongly comparable English–Romanian document pair (noise ratio 2:1), using a JRC Acquis-based GIZA++ extracted dictionary; 3 relevance feedback loops (the maximal values for each category *P*, *R* and *F* are bolded)

	Iteration 1		Iteration 2		Iteration 3	
0.1	*P*: 0.146	1509	*P*: 0.17	1525	*P*: 0.165	1526
	R: 0.73		*R*: **0.74**		*R*: **0.74**	
	F: 0.244		*F*: 0.277		*F*: 0.27	
0.3	*P*: 0.18	1162	*P*: 0.207	1207	*P*: 0.198	1215
	R: 0.686		*R*: 0.7		*R*: 0.694	
	F: 0.286		*F*: 0.32		*F*: 0.308	
0.7	*P*:**1**	110	*P*: **1**	154	*P*:**1**	154
	R: 0.234		*R*: 0.265		*R*: 0.265	
	F: 0.38		*F*: **0.42**		*F*: **0.42**	

random permutation of the sentences in each document was generated to ensure that the order in which the parallel sentences appear does not influence the outcome of PEXACC.

After running the PEXACC phrase extractor tool on the modified documents, we noticed that the parallelism probability above which all extracted pairs were correct (perfectly parallel) increased to 0.6. This happened due to the fact that the extractor encountered pairs of phrases in which bad translation equivalents exist, and, despite the fact that they do not have large translation probabilities, their number and disposition in each of the phrases in the pair fool the similarity measure.

Although all the extracted pairs over the 0.6 threshold are in fact parallel, there are many pairs over 0.5 that are also perfect parallel pairs: 'A new era' ⇔ 'O nouă eră', score 0.58, 'to the hospital.' ⇔ 'spre spital.', score 0.52, etc. But, because these pairs do not exist in our initial GS, we have no means to count them as precision points. Finally, we have to stress the fact that many correct pairs over 0.6 still cannot be found in GS. With these considerations in mind, one should judge the lower precision/recall of PEXACC on the noise-induced comparable pair of documents versus the parallel pair of the same documents.

The important thing to notice about Table 5.28 is that the recall—when considering all the pairs over the lowest accepted parallelism probability of 0.1—does not significantly decrease (a 2.1% decrease) when compared to the baseline in Table 5.26. This fact confirms that the only limitation of PEXACC in finding all relevant parallel pairs resides in the dictionary used and not in the order and/or amount of sentences in a document or the 'comparability' level of the document pair. This finding is obvious if one thinks about how PEXACC actually works: *by trying all combinations of source and target phrases and scoring each combination individually*. It cannot skip a pair no matter how much noise one adds to each document in the pair. But it fails in another respect: the value of parallelism probability that indicates true parallelism does not stay the same when we go from parallel documents to comparable documents and to weakly comparable documents.

Table 5.29, once more, confirms the fact that the recall is not significantly affected with the addition of noise. Here we ran PEXACC on a noise-altered version of our parallel document pair containing noise in the proportion of 2:1.

References

Aker, A., Kanoulas, E., & Gaizauskas, R. (2012a). A light way to collect comparable corpora from the Web. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)* (pp. 21–27), Istanbul, Turkey.

Aker, A., Feng, Y., & Gaizauskas, R. (2012b). Automatic bilingual phrase extraction from comparable corpora. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, IIT Bombay, Mumbai, India.

Aswani, N., & Gaizauskas, R. (2010). English-Hindi transliteration using multiple similarity metrics. In *Proceedings of the 7th Language Resources and Evaluation Conference (LREC 2010)*, Valletta, Malta.

Borman, S. (2009). *The expectation maximization algorithm*. A short tutorial. http://www.seanborman.com/publications/EM_algorithm.pdf

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics, 19*(2), 263–311.

Ceauşu, A. (2009). Statistical machine translation for Romanian. PhD Thesis, Romanian Academy (in Romanian).

Chen, S. F.(1993). Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics* (pp. 9–16), Columbus, OH.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, June 2005 (pp. 263–270), Ann Arbor, MI.

Fellbaum, C. (Ed.) (1998) WordNet: An electronic lexical database. Cambridge, MA: MIT Press.

Fung, P., & Cheung, P. (2004). Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)* (pp. 57–63), Barcelona, Spain.

Gale, W. A., & Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics, 19*(1), 75–102.

Gao, Q., & Vogel, S. (2008). Parallel implementations of a word alignment tool. In *Proceedings of ACL-08 HLT: Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, June 20, 2008 (pp. 49–57), Ohio State University, Columbus, OH.

Hewavitharana, S., & Vogel, S. (2011). *Extracting parallel phrases from comparable data*. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web (BUCC 2011)* (pp. 61–68), Portland, OR.

Ion, R. (2012). PEXACC: A parallel sentence mining algorithm from comparable corpora. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)* (pp. 2181–2188), May 21–27, 2012, Istanbul, Turkey.

Ion, R., Ceauşu, A., & Irimia, E. (2011a). An expectation maximization algorithm for textual unit alignment. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora (BUCC 2011)* (pp. 128–135), June 24th, 2011, Portland, OR.

Ion, R., Zhang, X., Su, F., Paramita, M., & Ştefănescu, D. (2011b). *Report on Multi-Level Alignment of Comparable Corpora*. Technical report no. D2.2 of the ACCURAT Project (http://www.accurat-project.eu/).

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)* (pp. 388–395), Barcelona, Spain.

Koehn, P. (2005). *Europarl: A parallel corpus for statistical machine translation*. In *Proceedings of the Tenth Machine Translation Summit*, September 12–16, 2005 (pp. 79—86), Phuket, Thailand.

Koehn, P., Och, F., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational*

Linguistics on Human Language Technology (pp. 48–54), May 27–June 1, 2003, Edmonton, Canada.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Cowan, B., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL Companion Volume Proceedings of the Demo and Poster Sessions* (pp. 177–180), Prague, Czech Republic.

Manning, C., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval* (Vol. 1). Cambridge: Cambridge University Press.

Munteanu, D. S., & Marcu, D. (2002). Processing comparable corpora with bilingual suffix trees. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* (pp. 289–295), July 6–7, 2002, University of Pennsylvania, Philadelphia, PA

Munteanu, D. S., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (pp. 160–167), July 07–12, 2003, Sapporo, Japan.

Och, F. J., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics, 29*(1), 19–51.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics, 30*(4), 417–449.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, July 7–12 2002 (pp. 311–318), University of Pennsylvania, Philadelphia, PA.

Quirk, C., Udupa, R., & Menezes, A. (2007). Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of the MT Summit XI* (pp. 321–327), September, 2007, Copenhagen, Demark.

Rauf, S. A., & Schwenk, H. (2011). Parallel sentence generation from comparable corpora for improved SMT. *Machine Translation, 25*(4), 341–375.

Skadiņa, I., Aker, A., Giouli, V., Tufiş, D., Gaizauskas, R., Mieriņa, M., et al. (2010). A collection of comparable corpora for under-resourced languages. In *Proceedings of the Fourth International Conference Baltic HLT 2010. Frontiers in Artificial Intelligence and Applications* (Vol. 219, pp. 161–168), IOS Press.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA 2006): Visions for the Future of Machine Translation* (pp. 223–231), Cambridge, MA.

Snover, M., Madnani, N., Dorr, B., & Schwartz, R. (2009). Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation* (pp. 259–268). Association for Computational Linguistics, Athens, Greece.

Ştefănescu, D., Ion, R., & Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. In *Proceedings of the16th Conference of the European Association for Machine Translation (EAMT 2012)* (pp. 137–144), May 28–30, 2012, Trento, Italy.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., et al. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006),* May 24–26, 2006, Genoa, Italy.

Steinberger, R., Eisele, A., Klocek, A., Pilos, S., & Schlüter, P. (2012). DGT-TM: A freely Available Translation Memory in 22 Languages. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, May 21–27, 2012, Istanbul, Turkey.

Stolcke, A. (2002). SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference of Spoken Language Processing (ICSLP 2002)* (pp. 901–904), September 2002, Denver, CO.

Talvensaari, T., Pirkola, A., Järvelin, K., Juhola, M., & Laurikkala, J. (2008). Focused web crawling in the acquisition of comparable corpora. *Information Retrieval, 11*(5), 427–445.

Thi Ngoc Diep, D., Besacier, L., Castelli, E. (2010). A fully unsupervised approach for mining parallel data from comparable corpora. In *Proceedings of the 14th Annual Conference of the European Association for Machine Translation (EAMT 2010)*, May 27–28, 2010, Saint-Raphaël, France.

Tillmann, C. (2009). A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* (pp. 225–228), Suntec, Singapore, August 4th, 2009.

Tsvetkov, Y., & Wintner, S. (2010). Automatic acquisition of parallel corpora from websites with dynamic content. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)* (pp. 3389–3392), Valletta, Malta, May 2010.

Tufiş, D., Ion, R., Ceauşu, A., & Ştefănescu, D. (2006). Improved lexical alignment by combining multiple reified alignments. In *Proceedings of the11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)* (pp. 153–160), Trento, Italy, April 3–7 2006.

Tufiş, D., Ion, R., Bozianu, L., Ceauşu, A., & Ştefănescu, D. (2008). Romanian wordnet: Current state, new applications and prospects. In A. Tanacs, D. Csendes, V. Vincze, C. Fellbaum, & P. Vossen (Eds.), *Proceedings of 4th Global WordNet Conference, GWC-2008, January 2008* (pp. 441–452). Hungary: University of Szeged.

Zhang, Y., Wu, K., Gao, J., & Vines, P. (2006). Automatic acquisition of Chinese-English parallel corpus from the web. In *Proceedings of 28th European Conference on Information Retrieval ECIR 2006*, April 10–12, 2006, London.

Chapter 6
Training, Enhancing, Evaluating and Using MT Systems with Comparable Data

Bogdan Babych, Yu Chen, Andreas Eisele, Sabine Hunsicker, Mārcis Pinnis, Inguna Skadiņa, Raivis Skadiņš, Gregor Thurmair, Andrejs Vasiļjevs, Mateja Verlic, and Xiaojun Zhang

Abstract This chapter describes how semi-parallel and parallel data extracted from comparable corpora can be used in enhancing machine translation (MT) systems: what are the methods used for this task in statistical and rule-based machine translation systems; what kinds of showcases exist that illustrate the usage of such enhanced MT systems. The impact of data extracted from comparable corpora on MT quality is evaluated for 17 language pairs, and detailed studies involving human evaluation are carried out for 11 language pairs. At first, baseline statistical machine translation (SMT) systems were built using traditional SMT techniques. Then they were improved by the integration of additional data extracted from the comparable corpora. Comparative evaluation was performed to measure improvements. Comparable corpora were also used to enrich the linguistic knowledge of rule-based machine translation (RBMT) systems by applying terminology extraction technology. Finally, SMT systems were adjusted for a narrow domain and included domain-specific knowledge such as terminology, named entities (NEs), domain-specific language models (LMs), etc.

Chapter editor: Inguna Skadiņa

B. Babych
University of Leeds, Leeds, UK
e-mail: b.babych@leeds.ac.uk

Y. Chen · A. Eisele · S. Hunsicker · X. Zhang
Deutsche Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Saarbrücken, Germany

M. Pinnis · I. Skadiņa (✉) · R. Skadiņš · A. Vasiļjevs
Tilde, Riga, Latvia
e-mail: inguna.skadina@tilde.lv

G. Thurmair
Linguatec, München, Germany

M. Verlic
Zemanta, Ljubljana, Slovenia

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_6

6.1 Introduction

Building a statistical machine translation (SMT) system requires a large amount of parallel data for model training. Reasonably good results can be achieved when the domain of the training corpus is close to the test data.

There are only a few parallel corpora publicly available for the lesser spoken languages of Europe. Several large-scale highly multi-lingual parallel language resources, such as the JRC-Acquis corpus (Steinberger et al. 2006), the DGT-TM (Steinberger et al. 2012) and DCEP corpus (Najeh et al. 2014), are made available by the European Commission's Joint Research Centre (JRC) and other European Union organisations (Steinberger et al. 2014). Different corpora are presented in the OPUS collection (Tiedemann 2009, 2012). SETimes (Tyers and Alperen 2010) is a parallel corpus from a multi-lingual news website into English and eight South-East European Languages (Albanian, Bulgarian, Croatian, Greek, Macedonian, Romanian, Serbian and Turkish).

For many under-resourced languages, multi-lingual comparable resources are widely available. Data extracted from comparable resources can be useful for machine translation. While methods on how to use parallel corpora in MT are well studied, methods and techniques for comparable corpora have not been thoroughly investigated.

The research in the field of the application of comparable corpora to the task of SMT has shown that adding extracted aligned parallel lexical data (additional phrase tables and their combination) from comparable corpora to the training data of an SMT system improves the system's performance in view of untranslated word coverage (Hewavitharana and Vogel 2008; Xu et al. 2006; Zhang 2011). It has also been demonstrated that language pairs with little parallel data can benefit the most from exploitation of comparable corpora (Lu et al. 2010).

Xu et al. (2006) exploit comparable data to extract parallel corpus. The proposed approach breaks documents into segments using pre-defined anchor words and then align these segments. In order to avoid errors in alignments, they present an advanced approach to extract the parallel sentences recursively by partitioning a bilingual document into two pairs. For Chinese–English data, this method produced translation results comparable to those of a state-of-the-art sentence aligner. A combination of the two approaches lead to a better translation performance.

Munteanu and Marcu (2006) achieved significant performance improvements from large comparable corpora of news feeds for English, Arabic and Chinese over a baseline MT system trained on existing available parallel data. The authors stated that the impact of comparable corpora on SMT performance is 'comparable to that of human translated data of similar size and domain'.

Irvine and Callison-Burch (2013) used comparable corpora to improve accuracy and coverage of phrase-based MT built on small amounts of parallel data. They showed that adding translations of low-frequency words from comparable corpora improves performance beyond what is achieved by inducing translations for out-of-

vocabulary words alone and that data from comparable corpora improves BLEU score (Papineni et al. 2002).

Most of the experiments are performed with widely used language pairs, such as French–English (Abdul-Rauf and Schwenk 2009, 2011), Arabic–English (Abdul-Rauf and Schwenk 2011) or English–German (Ştefănescu et al. 2012), while possible exploitation of comparable corpora for machine translation tasks is less studied for under-resourced languages (e.g. Skadiņa et al. 2012).

In this chapter, we analyse the impact of data extracted from comparable corpora on the machine translation task (both data-driven and rule-based) for under-resourced languages and narrow domains. Section 6.2 describes experiments to improve SMT systems trained on available parallel data by integration of additional data from comparable corpora for application in the general domain translation task. Section 6.3 proposes a methodology for how to assess changes in translation quality for systems enhanced with data extracted from comparable corpora and describes human evaluation results for eleven language pairs. Section 6.4 focusses on MT adaptation for a particular domain with the help of domain data extracted from comparable corpora. The last three sections deal with use cases. Section 6.5 analyses German–English MT adaptation to the automotive domain for both (rule-based and SMT) approaches. Section 6.6 analyses the role of machine translation in Web authoring, while Sect. 6.7 discusses the application of MT systems, enriched with data from comparable corpora, in computer-aided translation.

6.2 Enriching General Domain SMT Systems with Data from Comparable Corpora

In this section, we describe experiments to improve SMT systems trained on available parallel data (we call them baseline systems) by integration of additional data from comparable corpora for application in the general domain translation task.

6.2.1 Data Used for Experiments

The following publicly accessible parallel corpora were used to set up baseline SMT systems for the experiments: JRC: JRC-Acquis, DGT: DGT-TM (Steinberger et al. 2012), SETimes,[1] Europarl, and News Commentary.[2] Table 6.1 shows the size of the training data that was used to train the baseline systems.

[1] http://www.setimes.com

[2] The News Commentary corpus is from the training data released for the shared tasks of the last few workshops for statistical machine translation (SMT).

Table 6.1 Size of corpora for baseline systems

Language Pair	Corpora	Size (lines)
English–Latvian	DGT, JRC	2,305,674
English–Lithuanian	DGT, JRC	2,339,905
English–Estonian	DGT, JRC	2,239,791
English–Slovenian	DGT, JRC	2,190,704
German–Romanian	DGT, JRC	615,336
Latvian–Lithuanian	DGT, JRC	974,161
Lithuanian–Romanian	DGT, JRC	940,461
English–Greek	SETimes	169,337
English–Croatian	SETimes	157,950
English–Romanian	SETimes	171,573
Greek–Romanian	SETimes	175,019
German–English	Europarl, Newscommentary	1,639,893

We conducted three groups of directions in our experiments. The first group uses JRC and DGT for training and the second group uses SETimes. Although the data combining JRC and DGT is fairly large in size, the domain of the data is rather limited to legislation/law. The systems based on such a data set perform poorly on general translation tasks of other open domains, in spite of the high translation quality for in-domain tests reported in previous literature. Therefore, we still consider these language pairs under-resourced. The second group is the opposite. This group of baseline systems is based on the SETimes corpus, which covers a relatively broad range of topics and is much smaller in size than JRC or DGT. The third group includes only German–English as a control group. We used both Europarl and News Commentary for this group. This dataset has a presumably open domain and large size. This setup allows us to have more contrastive studies on the effect of using comparable corpora, as the set up for German–English has been used for state-of-the-art systems.

As for language model (LM) training, we use the target portion of the corresponding parallel data.

To enrich the baseline SMT systems, we use data extracted from comparable corpora collected by tools described in Chap. 3. We distinguish between the data extracted from news corpora (News) and Wikipedia articles corpora (Wiki).

The ACCURAT toolkit (Pinnis et al. 2012a) was used to extract semi-parallel sentences from the aligned comparable corpora. Table 6.2 gives the statistics about the extracted data. The amount of data varies a lot between language pairs and also between the two comparable corpora.

We used the News corpus to adapt the language models. The amount of data is reported in Table 6.3.

We tune all models on the same development set (Table 6.4) to get comparable results. The tuning is performed using minimal error rate training (MERT; Och 2003).

Table 6.2 Statistics of the extracted semi-parallel data from comparable corpora

	Number of lines	
Language Pair	News	Wiki
English–Latvian	112,398	116,240
English–Lithuanian	33,219	179,578
English–Estonian	19,048	128,939
English–Slovenian	67,508	5418
German–Romanian	10,227	–
Latvian–Lithuanian	7163	29,370
Lithuanian–Romanian	9470	–
English–Greek	6641	45,646
English–Croatian	36,663	31,048
English–Romanian	23,820	45,771
Greek–Romanian	1783	–
German–English	13,782	–

Table 6.3 Statistics of monolingual comparable corpora

Language	Size (lines)
Croatian	180,908
German	1,485,774
Greek	1,267,731
English	2,235,282
Estonian	711,147
Latvian	789,178
Lithuanian	1,061,713
Romanian	1,815,170
Slovenian	470,782

Table 6.4 Statistics about development data

Language Pair	Name of development set	Length (in lines)
English–Latvian	Tilde	1000
English–Lithuanian	Tilde	1000
English–Estonian	Tilde	1000
English–Greek	SETimes	600
English–Croatian	SETimes	600
Croatian–English	SETimes	600
English–Romanian	SETimes	600
Romanian–English	SETimes	600
English–Slovenian	mtserver	1000
Slovenian–English	mtserver	1000
German–English	WMT-dev 2008	2051
German–Romanian	RACAI	3000
Romanian–German	RACAI	3000
Greek–Romanian	SETimes	600
Romanian–Greek	SETimes	600
Lithuanian–Romanian	DGT-dev	3000
Latvian–Lithuanian	Tilde	1000

Additionally, we make use of the target language tuning texts to interpolate the language models as described in the next subsection.

6.2.2 Methodology

When improving SMT systems, we need to look at the two models used in translation: the translation model (TM) and the language model (LM). The comparable data can be used to adapt both models.

6.2.2.1 Mixture Translation Model

Including additional parallel corpora as training data to an SMT system usually yields an improvement to a certain extent. However, the additional texts could also introduce errors that do not exist in the original model. This case is especially more likely to happen when the parallel texts are not translations of each other: for example when we have misaligned sentences in the comparable corpora. On the other hand, due to various reasons, the added data might not be dominant enough among the other sources of training corpora to help the SMT system to recover from the errors in the baseline system. Therefore, in addition to a single translation model built from both the parallel corpora and the comparable data as a whole, we experimented with mixture models that distinguish texts from different sources.

The mixture models, introduced by Xu et al. (2007), start from individual models that are generated separately using the sets of texts from different sources. The most straightforward way is to divide the data into two subsets: the original parallel corpora versus the aligned texts that were extracted from the comparable corpus. Such a partition may be very close to the baseline model when the sizes of the two subsets differ too much, as it would lead to a mixture model that relies on the larger subset. Thus, in order to emphasise and better control the contribution of parallel and comparable data to the final translation, we choose to further divide the original parallel data into separate corpora, from each of which we generate a different translation model. This approach also allows us to understand the influence of each individual corpus (parallel or comparable) in the SMT system, and it is especially important when the parallel corpora used in the baseline systems are from very different domains.

As a state-of-the-art word alignment algorithm such as GIZA++ tends to perform poorly for a limited amount of data, we generate the word alignments for the mixture model by training over the combination of all the training data, that is the parallel data alongside the extracted sentence pairs from the comparable corpus in order to find sufficient alignment points that are useful for constructing a translation model. Then, after the second step, the word alignments are split into segments corresponding to the individual corpus.

We construct the individual translation models from the word alignments for each corpus. The models are then sorted by the size of the corresponding training corpora, given the fact that the probabilistic estimation over a larger set of data is usually more reliable.

The other models are appended to the largest model in this sorted order such that only phrase pairs that were never seen previously are included. Lastly, we add new features (in the form of additional columns) to the phrase table of the final translation model to indicate each phrase pair's origin. Each new column corresponds to one model, including the original model. If a phrase table entry appears in a model, its feature value in the corresponding column is 2.718; otherwise, it is 1.

Table 6.5 shows a few sample entries from the phrase table of a mixture model created in our experiments for English–Latvian translation. The first five columns are the probabilistic scores estimated in the standard phrase-based SMT training, including the inverse phrase translation probability $\varphi(f|e)$, the inverse lexical weighting $lex(f|e)$, the direct phrase translation probability $\varphi(e|f)$, the direct lexical weighting $lex(e|f)$ and the phrase penalty which is always $e^1 = 2.718$. Following the scheme of defining the phrase penalty, we added three additional columns to the phrase table, corresponding to the three individual models which have been sorted by size. In this example, the first column refers to the JRC model, the second column refers to the DGT model and the last column is for the extracted comparable corpus. The values in these three columns are either 2.718 or 1, indicating whether the phrase pairs exist in the individual models. For example the last three columns for the phrase pair '*economic approaches*'-'*ekonomiskas metodes*' are 1, 2.718, and 1. This means that this pair is originally from the DGT model and does not appear in the other two.

In the mixture model, segments repeated by many sources are considered more probable for translation. On the other hand, unique pieces from some sources may lead us to valuable information, such as terminologies from a particular domain in the comparable corpus. The former case corresponds to phrase pairs with very high probabilities, whereas the latter is still included in the model.

Table 6.5 Sample entries from the phrase table of a mixture model for English–Latvian

Source phrase (e)	Target phrase (f)	Probabilistic scores	Origin markers
Economic, political	Ekonomiskās, politiskās	0.079 0.266 0.011 0.011 2.718	2.718 2.718 2.718
Economic, social	Ekonomiskajā, sociālajā	0.119 0.048 0.001 0.001 2.718	2.718 2.718 1
Economic downturn	Ekonomikas lejupslīdi	0.120 0.134 0.017 0.016 2.718	1 2.718 2.718
Economic subjects	Ekonomiskajos priekšmetos	0.406 0.555 0.051 0.001 2.718	1 2.718 1
Economic approaches	Ekonomiskas metodes	0.241 0.004 0.241 0.001 2.718	1 2.718 1

6.2.2.2 Interpolating Language Models

To make the best use of the fact that our language models have been trained on different texts, we want to combine them into one and adapt the n-gram probabilities accordingly. Although, for example, our baseline JRC and DGT language models are out of domain, we do not want to completely lose the information they contain. On the other hand, these models are big enough that they can overpower the influence of the new language model that has been trained on much smaller amounts of data. Here we need to adjust the n-gram probabilities so that they mirror what we would expect from our target domain.

Combination is done by optimising the perplexity of the interpolated language model on an in-domain development text in the target language. We then receive a lambda for each language model we used; we can adjust the probabilities for each n-gram. In this way, we combine the probabilities from the different language models into one (Schwenk and Koehn 2008).

The interpolated language model will then be used for the new SMT system.

6.2.3 Experiments with Data Extracted from Comparable Corpora

In total, we worked on seventeen language pairs: English–Latvian, English–Lithuanian, English–Estonian, English–Greek, English–Croatian, Croatian–English, English–Romanian, Romanian–English, English–Slovenian, Slovenian–English, German–English, German–Romanian, Romanian–German, Greek–Romanian, Romanian–Greek, Lithuanian–Romanian, Latvian–Lithuanian. Our main concern is to translate from English, but we also investigate a few language pairs that do not involve English and for which there is very little data available.

We trained state-of-the-art phrase-based models using 7-gram phrase-tables and 5-gram interpolated language models. For the training, we used the data described in Table 6.1, where the parallel data was used for the translation model and the target language text was used to generate the language model. In the case of the language pairs using DGT and JRC, as well as German–English, we interpolated the language models built on the two baseline corpora using the target side of our development set. This is the same set that we later optimised the SMT translation parameters on using Minimal Error Rate Training (MERT) and is listed in Table 6.4.

Then, for each language pair, we trained systems using the additional data described in Table 6.2. We use the same general settings for training the enriched models as we did for training the baseline models. We trained separate models for the data extracted from the News and the Wiki data to examine the influence of the different sorts of data.

For the interpolated model, we use the target side of both the baseline parallel data and the collected comparable corpus. The translation model is trained on the

extracted parallel data and the baseline corpora. We apply this approach to both the News and the Wiki extracted data. For the language model, we use the comparable News corpus for both News and Wiki experiments.

For the mixture model, we trained a phrase table on each individual corpus and then combined them into a single mixture translation model. For the language model, we used the interpolated language models.

All systems were tested on the same test set, which consists of 511 sentences from general domain text (Skadiņš et al. 2010). Table 6.6 lists the results for all experiments on interpolated language models and mixture models. Figures in bold indicate models that outperform the baseline. The best model for each language pair is denoted with an asterisk.

We see that not every approach works equally well for each language direction. The largest improvement in BLEU score can be noted for those language pairs that only used the SETimes corpus with less than 200,000 lines per language pair as the baseline corpus. The improvements are smaller for the language pairs using DGT/JRC. For some of the language pairs, we did not observe any improvement by adding the data, and thus, we further investigated English–Lithuanian pair. We describe these experiments in the next subsection.

Table 6.6 Evaluation results (BLEU scores) for all experiments

Language pair	Baseline	Interpolated LM		Mixture models
		News	Wiki	
English–Latvian	12.74	**13.20** (+.46)	**13.07**(+.33)	**13.25*** (+.51)
English–Lithuanian	12.66	12.21(−.45)	12.33 (−.33)	11.94 (−.71)
English–Estonian	10.44	**11.23*** (+.79)	**10.46** (+.02)	**10.88** (+.44)
English–Greek	19.06	**21.40** (+2.34)	**23.67* (+4.61)**	**20.61** (+1.55)
English–Croatian	10.91	10.36 (−.55)	**11.25** (+.34)	**11.45*** (+.54)
Croatian–English	20.78	20.31 (−.47)	**21.17** (+.39)	**21.91***(+1.13)
English–Romanian	17.89	**20.11*** (+2.22)	**20.00** (+2.11)	**19.08** (+1.19)
Romanian–English	21.54	**26.16** (+4.62)	**30.35*** (+8.81)	**25.27** (+3.73)
English–Slovenian	18.20	**18.68*** (+.48)	**18.66** (+.46)	17.70 (−.50)
Slovenian–English	26.28	**27.40** (+1.12)	**27.46*** (+1.18)	**27.31** (+1.03)
German–English	27.90	**28.62*** (+.72)	–	27.88 (−.02)
German–Romanian	9.66	**10.14*** (+.48)	–	8.37 (−1.29)
Romanian–German	10.22	9.56 (−.66)	–	9.97 (−.25)
Greek–Romanian	15.81	**17.25*** (+1.44)	–	**17.15** (+1.34)
Romanian–Greek	12.13	**13.59*** (+1.46)	–	**13.37** (+1.24)
Lithuanian–Romanian	9.91	9.24 (−.67)	–	4.67 (−5.24)
Latvian–Lithuanian	12.12	**12.69*** (+.57)	8.70 (−3.42)	**12.41** (+.29)

6.2.4 Staggered Experiments

The LEXACC tool, which is described in Chap. 5, assigns a score to each sentence pair extracted from comparable corpora, denoting how likely these two sentences are parallel. As such, the LEXACC score should allow us to predict how usable a particular chunk of data is, that is, will the use of this data increase translation quality.

To test this influence of the LEXACC score, we split up the extracted data. We want to check the effect of the score both in intervals and in a cumulative fashion. The hypothesis for the former is that data with a higher LEXACC score should be more helpful than data with a lower score. In the cumulative experiments, we choose different thresholds. As the score goes down, the less parallel the data will become, and more errors will be introduced into the translation model. But as the distribution of the data follows Zipf's law, we have very few items with a very high score, but, the lower the score, the more sentences LEXACC extracts. However, we also need to take into account how much data we have: for higher thresholds, LEXACC will only be able to extract small amounts of data. Here we are interested in the threshold that allows the maximal increase in translation quality for the amount of data used. This threshold may vary for different corpora which is an effect we also want to examine.

As we couldn't observe an improvement in translation quality in the experiments using the full data for English–Lithuanian, we treat this language in these experiments. Additionally, we examine English–Latvian and English–Romanian. We saw improvements in these two languages, but we are interested in seeing how much each part of the data contributes. We chose these languages, because they work with different baseline corpora. This allows us to see the effects of adding a small amount of data to a large out-of-domain corpus (DGT/JRC in the case of English–Latvian) and the effects of adding similar amounts of data to a small in-domain corpus (SETimes for English–Romanian).

6.2.4.1 English–Latvian

For English–Latvian, we examined both the interpolated language models and the mixture models. The problem with using mixture models is that the probabilities associated with the entries in the phrase table become less trustworthy on such a small set of data. Tables 6.7 and 6.8 give the amount of data (in sentence pairs) in the different intervals.

We did not investigate data with an LEXACC score of less than 0.1 (the default threshold of LEXACC is 0.1). We see that we have very little data with a score higher than 0.9, but we get more data for lower scores.

We used each chunk of the data to retrain the SMT model and evaluated it the same as the baseline and full enriched models. Tables 6.9 and 6.10 give the BLEU scores for those experiments. The baseline SMT system reached a BLEU score of 12.66. Experiments that perform worse than the baseline are marked in italics; the best experiment in each approach and corpus is marked in boldface.

Table 6.7 Statistics about interval experiments for English–Latvian

Interval	News	Wiki
>0.9	169	208
0.9–0.8	3226	1730
0.8–0.7	13,264	5791
0.7–0.6	12,735	6868
0.6–0.5	9009	7085
0.5–0.4	6914	8556
0.4–0.3	8720	13,902
0.3–0.2	15,325	26,669
0.2–0.1	43,036	45,431

Table 6.8 Statistics about cumulative experiments for English–Latvian

Cumulative	News	Wiki
>0.9	169	208
>0.8	3395	1938
>0.7	16,659	7729
>0.6	29,394	14,597
>0.5	38,403	21,682
>0.4	45,317	30,238
>0.3	54,037	44,140
>0.2	69,362	70,809
>0.1	112,398	116,240

Table 6.9 BLEU scores for interval experiments for English–Latvian

	Interpolated LM		Mixture models	
Interval	News	Wiki	News	Wiki
>0.9	13.48	13.73	12.97	13.48
0.9–0.8	13.60	13.57	**13.29**	13.36
0.8–0.7	13.15	13.57	12.71	13.29
0.7–0.6	**13.67**	**13.83**	12.76	13.23
0.6–0.5	13.49	13.50	12.84	12.91
0.5–0.4	13.54	13.57	12.78	**13.72**
0.4–0.3	13.31	13.39	12.80	13.61
0.3–0.2	12.77	13.40	12.99	13.44
0.2–0.1	*12.15*	*12.63*	12.84	12.86

Figure 6.1 illustrates the effect of the LEXACC score on the BLEU score. The data in the interval of [0.1,0.2] scores the worst results and doesn't even reach the BLEU score of the baseline (plotted for comparison purposes). As the LEXACC score increases, we can also see an increase in BLEU score. Using the interpolated language models, this development is rather steady. When we compare News to the Wiki-extracted data, the interpolated language models show similar trends.

According to the BLEU scores, the translation results using the mixture models seem less correlated to the LEXACC score, mostly due to the fact that the mixture models are very sensitive to the size of the data that is used to construct the additional phrase tables. Higher LEXACC thresholds indicate better quality of extracted sentence pairs. Meanwhile, these high scores also result in less extracted data. In

Table 6.10 BLEU scores for cumulative experiments for English–Latvian

	Interpolated LM		Mixture models	
Cumulative	News	Wiki	News	Wiki
>0.9	13.48	**13.73**	12.97	13.48
>0.8	13.50	13.34	13.77	12.90
>0.7	13.66	*12.56*	13.19	**13.49**
>0.6	**13.86**	13.55	**13.78**	12.97
>0.5	13.73	13.10	13.00	13.11
>0.4	13.68	13.30	13.41	12.90
>0.3	13.58	13.22	13.26	12.96
>0.2	13.74	13.46	13.75	13.15
>0.1	13.20	13.07	13.25	–

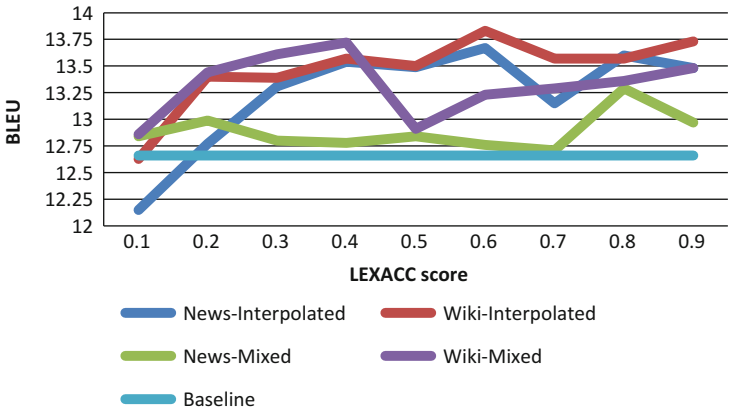


Fig. 6.1 BLEU scores for interval experiments for English-Latvian

general, the translation model constructed over a small amount of data tends to contain less useful phrase pair entries while having high probability estimation values. When combining a small model with high scores and a much larger model with much lower scores, one cannot avoid penalising the phrase pairs from the small model in order to use entries that exist in the other models, which are actually the majority of the combined model. Thus, in general, the tuning procedure seems to assign higher weights to the feature that represents the larger model. As a result, the additional data might not have as much influence on the final translation as we hope. It also explains why, in the experiment for Wiki data, the BLEU score drops significantly at the LEXACC interval [0.4,0.5], for which there are nearly 40% less sentence pairs than for [0.3,0.5]. The BLEU score increases again for higher LEXACC scores, as the size difference is smaller for the other cases. In practice, the probability estimation in the sub-models should all be normalised, but this would make it more difficult to compare results for different extracted data. Therefore, we chose to retain the probability scores in the sub-models.

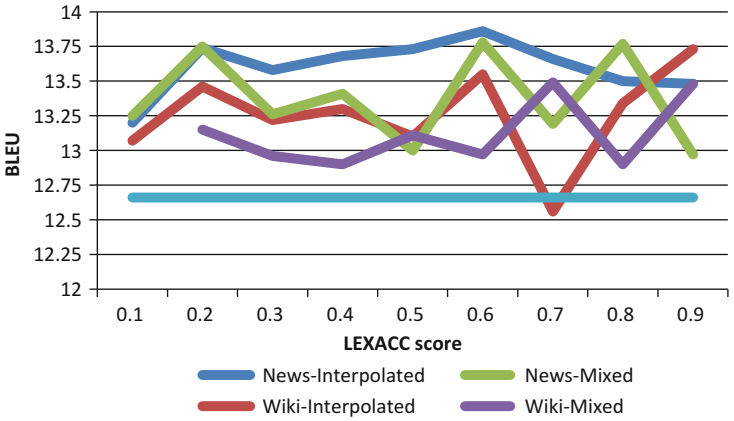


Fig. 6.2 BLEU scores for cumulative experiments for English–Latvian

The results for the cumulative experiments are not quite as clear. The effect of the LEXACC score on BLEU is plotted in Fig. 6.2. Here we see a lot of fluctuation. Although the best BLEU scores are comparable for three of the four experiment runs, they occur in different intervals. Especially interesting is the behaviour of the data with an LEXACC score of 0.7 and above. In News, this chunk leads to an improvement using the interpolated LMs, but the BLEU score drops by almost 0.6 for the mixture models which is a significant deterioration. The Wiki data behaves similarly, except that here the BLEU score of the interpolated LM drops even below the baseline performance. However, this data is the best performing for the mixture models

Figure 6.2 illustrates this point. We see a lot of ups and downs, although the data using a threshold of 0.6 seems to work reliably well for both models and both corpora.

6.2.4.2 English–Romanian

The training data for English–Romanian was very small, so our hypothesis was that this language direction was very sensitive to the quality of the newly added data. Whereas the DGT/JRC corpora are big enough to smooth out mistakes in the translation probabilities, the SETimes corpus is small enough that even the relatively small amount of extracted data can counteract the probabilities extracted from the original data: the English–Latvian baseline corpus consists of 2,305,674 lines, with 112,398/116,240 lines extracted from each comparable corpus, adding about 5% of the data to the baseline corpus. For English–Romanian, we only had 171,573 lines in the baseline, so the data from News (238,320 lines) and the Wiki corpus (45,771 lines) amount to 14% and 27%, respectively. Thus, the influence of the new data will be much higher than for the previous experiments.

Table 6.11 Statistics about interval experiments for English–Romanian

Interval	News	Wiki
>0.9	246	5807
0.9–0.8	2468	13,174
0.8–0.7	2221	6530
0.7–0.6	1511	3993
0.6–0.5	2021	3653
0.5–0.4	2636	3974
0.4–0.3	4024	3826
0.3–0.2	8693	4814

Table 6.12 Statistics about cumulative experiments for English–Romanian

Cumulative	News	Wiki
>0.9	246	5807
>0.8	2714	18,981
>0.7	4935	25,511
>0.6	6446	29,504
>0.5	8467	33,157
>0.4	11,103	37,131
>0.3	15,127	40,957
>0.2	23,820	45,771

For this language pair, we examined only the interpolated language models, as the results on the mixture models were too unsteady. Tables 6.11 and 6.12 give the amount of data in the different intervals.

The distribution of this data is especially interesting. In English–Latvian, the distribution followed Zipf's law, that is there was very little data for the high scores, but the lower the score, the more data was extracted. For English–Romanian, however, this only holds for News. The Wiki corpus behaves differently: here we have unusually many sentence pairs with a high score. This cannot simply be explained by the fact that Wiki articles are inherently more strongly comparable than news text, as then this would also have to hold for other language pairs. Manual inspection of the data suggests that many articles in the Romanian Wikipedia have been originally translated from the English Wikipedia. We consider this an anomaly.

The procedure of these experiments is the same as for the previous English–Latvian experiments. For each chunk of the data, we retrain the SMT models and compare it against the baseline, which was evaluated with a BLEU score of 17.89. Table 6.13 shows the results for the interval experiments, the best results are marked in boldface.

All systems outperform the baseline, but the overall tendency for improvement of BLEU is not as clear-cut as it was for the previous experiment (Fig. 6.3). Instead, we see that the improvement in BLEU varies a lot over of the intervals. For the Wiki corpus, which adds 25% to the original data, our assumption that higher LEXACC scores predict a higher increase in BLEU still holds, but, for the News data, we find that using the maximum amount of available data results in the highest gain. Here we must take into account the amount of data in each interval: although Wiki can offer

Table 6.13 BLEU scores for interval experiments for English–Romanian

Interval	Interpolated LM	
	News	Wiki
>0.9	19.45	19.08
0.9–0.8	18.74	19.63
0.8–0.7	19.28	**19.64**
0.7–0.6	19.81	18.79
0.6–0.5	20.03	19.13
0.5–0.4	20.04	19.29
0.4–0.3	**20.22**	19.30
0.3–0.2	19.92	18.30

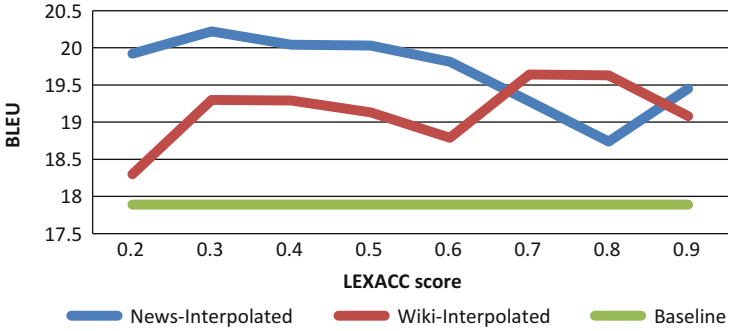


Fig. 6.3 BLEU scores for interval experiments for English–Romanian

Table 6.14 BLEU scores for cumulative experiments for English–Romanian

Cumulative	Interpolated LM	
	News	Wiki
>0.9	19.45	19.08
>0.8	19.04	19.59
>0.7	18.54	19.75
>0.6	18.71	20.03
>0.5	19.01	19.98
>0.4	19.85	**20.27**
>0.3	19.44	20.40
>0.2	**20.11**	20.00

us 13,000 additional lines in the interval of [0.9,0.8], there are only 2500 sentences in the same interval in the News corpus.

Table 6.14 shows the results for the cumulative experiments, the best results are marked in boldface. As for the interval experiments, all models improve over the baseline.

In Fig. 6.4, we see less variation than for English–Latvian, with rather obvious thresholds for the corpora. As for the interval experiments, we get the best results by using all of the available additional data for the News corpus, whereas the threshold for Wiki lies at 0.3. This is consistent with the best LEXACC performance, where we reach the best F1 score at a threshold of 0.36. Although these thresholds are close, we

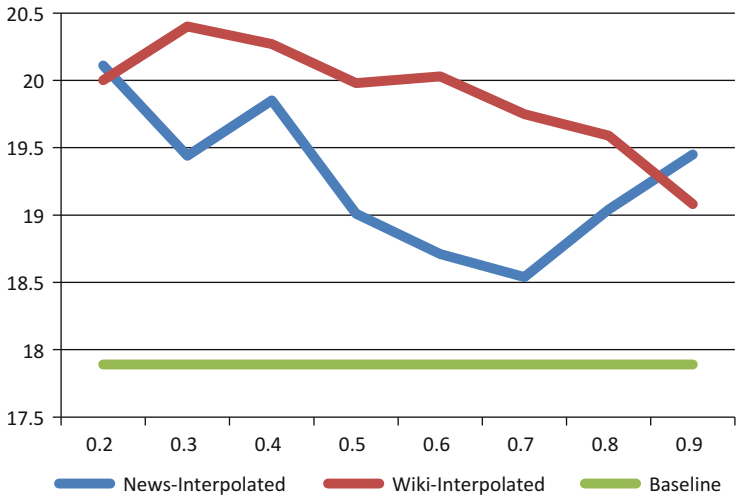


Fig. 6.4 BLEU scores for cumulative experiments for English–Romanian

see quite a difference between the different corpora: the News corpus improves by 0.7 BLEU points when using all the data, whereas the performance of the Wiki corpus drops by 0.3 BLEU points when using the same threshold. The BLEU scores for threshold 0.3 differ by almost one full BLEU score, a very significant difference. This can be explained by taking into account the amount of data (see Table 6.12); for this interval, we have almost three times as many sentences for Wiki than for News.

6.2.4.3 English–Lithuanian

As shown in Table 6.6, using the full data for English–Lithuanian did not result in an improvement of BLEU score. As we have seen a lot of variation in the BLEU scores for the individual chunks of the data, we decided to give English–Lithuanian the same treatment.

The size of the original baseline corpus consisting of DGT/JRC was 2,339,905 lines. We could add 33,219 lines to this from the News corpus (+1.42%) and 179,578 lines from Wiki (+7.67%). Splitting up the data into the individual chunks results in the amount of data shown in Tables 6.15 and 6.16.

The data again follows the distribution we would expect. The difference in size between the News and Wiki corpus is significant—in each section, we have about six times as much data for Wiki than for the News corpus.

The baseline produced a BLEU score of 12.66. Tables 6.17 and 6.18 present the BLEU scores for the respective interval and cumulative experiments, the best results are marked in boldface.

None of the interval experiments perform better than the baseline, but we can see that the Wiki data performs much better than the News data. In Fig. 6.5, we observe the general tendency that higher scoring intervals result in better BLEU scores, but

Table 6.15 Statistics about interval experiments for English–Lithuanian

Interval	News	Wiki
>0.9	28	1089
0.9–0.8	352	4265
0.8–0.7	1006	6450
0.7–0.6	1061	6307
0.6–0.5	1317	7656
0.5–0.4	1692	10,393
0.4–0.3	2495	17,628
0.3–0.2	5536	35,574
0.2–0.1	19,732	90,196

Table 6.16 Statistics about cumulative experiments for English–Lithuanian

Cumulative	News	Wiki
>0.9	28	1089
>0.8	380	5354
>0.7	1386	11,804
>0.6	2447	18,111
>0.5	3764	25,767
>0.4	5456	36,160
>0.3	7951	53,788
>0.2	13,487	89,562
>0.1	33,219	179,758

Table 6.17 BLEU scores for interval experiments for English–Lithuanian

	Interpolated LM	
Interval	News	Wiki
>0.9	**12.48**	**12.64**
0.9–0.8	12.00	12.49
0.8–0.7	12.47	12.40
0.7–0.6	12.47	12.53
0.6–0.5	12.33	12.37
0.5–0.4	12.46	12.00
0.4–0.3	12.01	12.26
0.3–0.2	12.04	12.34
0.2–0.1	12.13	11.87

the amount of data does not seem sufficient to push the enriched system over the baseline.

Using the interval, especially the small amounts available for the News corpus, did not yield an improvement in the system.

Most of the cumulative experiments also perform worse than the baseline (Fig. 6.6). It is interesting to note that the best-performing system, which also improves over the baseline, uses the same threshold we have already identified as optimal for English–Latvian, namely 0.6. This can be interpreted such that Lithuanian generally behaves similar to Latvian.

Table 6.18 BLEU scores for cumulative experiments for English–Lithuanian

	Interpolated LM	
Cumulative	News	Wiki
>0.9	12.48	**12.64**
>0.8	12.35	12.56
>0.7	12.35	12.34
>0.6	**12.94**	12.43
>0.5	11.90	12.41
>0.4	12.11	12.32
>0.3	12.45	12.25
>0.2	12.37	11.93
>0.1	11.21	12.33

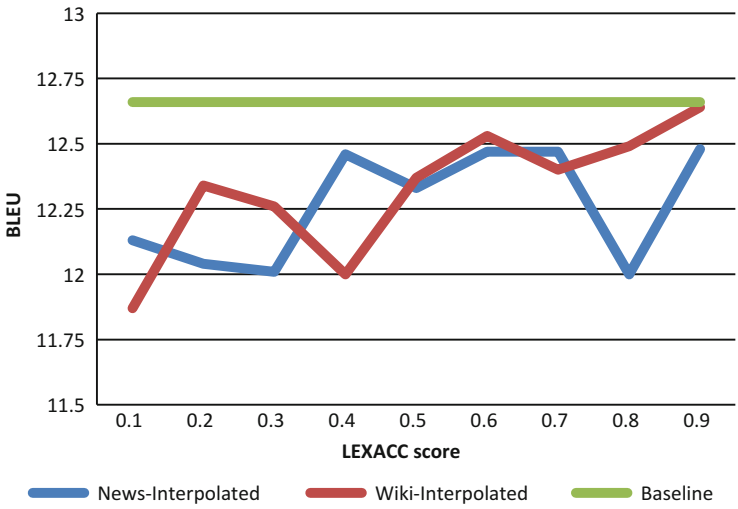


Fig. 6.5 BLEU scores for interval experiments for English–Lithuanian

It is worthwhile to note that the upper intervals get close to the performance of the baseline which leads us to believe that the amount of data extracted was simply too small to have a large enough impact on the baseline corpus.

6.3 Human Evaluation of MT Output

The human evaluation experiment is designed to measure the difference between the performance of the baseline MT systems built using only parallel data and those that were enhanced with sentences and phrases extracted from comparable corpora (CC). We developed a special evaluation scenario which takes into account the properties of the evaluated data.

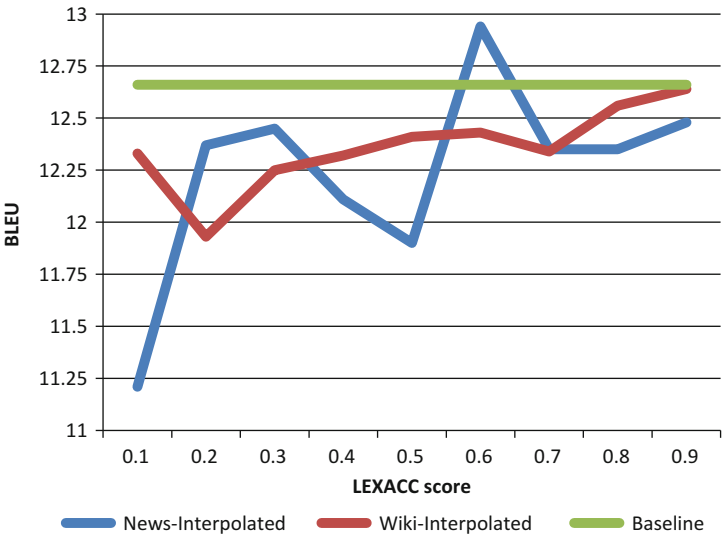


Fig. 6.6 BLEU scores for cumulative experiments for English–Lithuanian

Traditional measures of translation quality involve metrics such as adequacy (fidelity, i.e. the amount of information preserved in MT output compared with the gold-standard human translation), fluency (the degree of naturalness or well-formedness of a sentence according to the requirements of the target language, irrespective of the original sentence) or informativeness (responses to a multiple-choice questionnaire; White et al. 1994). However, we noted that none of these standard measures can adequately quantify the changes in translation quality for systems enhanced with resources based on comparable corpora.

6.3.1 Evaluation Methodology and the Interface

For our evaluation experiment, we developed a novel evaluation methodology, which captures differences between the baseline and the modified MT systems more directly and systematically. Specifically, we were interested in the following aspects of MT evaluation.

Firstly, we need to capture a general user intuition about the translation quality of evaluated sentences taken in context. The division between adequacy and fluency makes sense for non-translator users; however, our target audience—translation studies students or professional translators—are able to assess the relevant importance of adequacy and fluency for their specific post-editing or summarisation tasks. In this respect, it makes sense to collapse both evaluation measures onto a single scale, for which we obtain professional user ratings. In our scenario, translators were

asked to evaluate the *overall translation quality* of the sentences presented to them in the order that they normally appear in a text.

Secondly, we are also interested in a comparative aspect of evaluation, specifically—the differences between the baseline sentences compared with corresponding aligned CC-enhanced sentences. Traditional comparative-based metrics have two major shortcomings from this perspective: they do not place compared sentences onto any systematic scale, and they do not compare specific linguistic: for example lexical differences within the sentences. In our case, we need to tie the differences to an interpretable scale and focus the attention of evaluators on specific changes in otherwise similar sentences. In our scenario, not all sentences are different in the baseline and the enhanced output, and, if there are differences, they are usually minimal; it can be just one or two words or different morphological forms of words. We also cannot use here standard adequacy or fluency measures independently on the baseline and CC-enhanced MT output; this would miss such small differences, since granularity of the standard 5-point scale could be insufficient for capturing the changes.

Therefore, in our evaluation scenario, we combined the question about the general translation quality with the comparative evaluation task: lexical differences between the baseline and the enhanced versions were highlighted, and users were asked to rate the appropriateness of lexical choices for each of the highlighted words. Sentences without any differences were removed (which sometimes disrupted the intra-sentential context, but the number of such omission was small compared to the overall text size), and the order of presentation was randomised. The origin of the text was anonymised; users did not know whether the sentence was coming from the baseline or the CC-enhanced MT system.

Highlighting lexical differences is intended to focus the attention of evaluators on specific linguistic issues, and the numerical scale combined with the comparative framework allows us to adequately quantify the quality level, as well as relative and absolute improvement in translation quality.

The evaluation interface presented to users had the following form (Fig. 6.7).

6.3.2 Experiment Set-Up

System output was generated for the baseline and CC-enhanced MT systems for the following translation directions and domains: News domain: German–English (de-en); Romanian–English (ro-en); Slovenian–English (sl-en); Croatian–English (hr-en); Romanian–German (ro-de); Latvian–Lithuanian (lv-lt); English–Latvian (en-lv); English–Croatian (en-hr); English–Greek (en-el); German–Romanian (de-ro); Greek–Romanian (el-ro); Automotive domain: German–English (de-en) and English–Latvian (en-lv). An evaluation set of 511 sentences (circa 11,000 words) was used for all translation directions.

Evaluation packs for human evaluation were constructed using the following procedure: sentences different in the baseline versus CC-enhanced output were identified. Words different in the baseline versus CC-enhanced output were

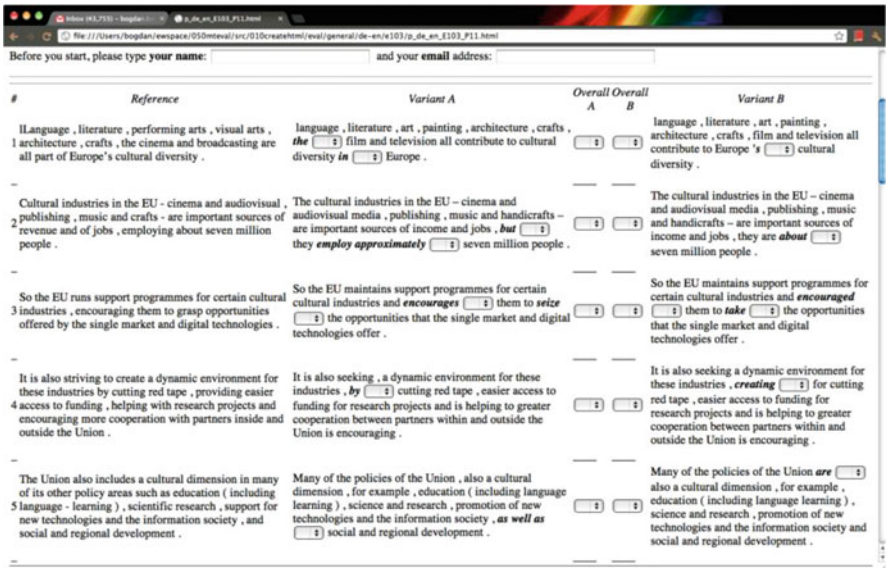


Fig. 6.7 Evaluation interface for professional translators

automatically highlighted; if several consecutive words were highlighted, all of them were evaluated together as a phrase. The order of presentation of the CC-enhanced versus baseline systems was randomised. Evaluation packs were presented to evaluators within a Web interface that automatically calculated submitted evaluation results (using CGI script).

The set of 120 sentences (those were the first 120 non-similar sentences out of the complete set of 511 sentences used for calculating BLEU scores) were typically used for the human evaluation experiment, with at least three independent judgments collected for each sentence and also—for each highlighted word or phrase that was different in the baseline versus CC-enhanced translation.

For each target language, we recruited at least three evaluators, most of them had backgrounds in translation (either professional translators, translation students or linguists), and obtained at least three independent scores for each of the compared sentences and lexical differences. Evaluators were asked to evaluate translation quality of the compared sentences and the quality of translation choices of the highlighted words or phrases.

For judging *overall quality,* evaluators were asked to rate each pair of sentences on a scale of 1 to 5: (1 = Translation is not at all good ... 5 = Translation is very good). For judging *lexical translation* choices, the judges were asked to use the same scale for rating translation quality of highlighted words and phrases: (1 = Very bad translation choice... 5 = Very good translation choice).

6.3.3 Human Evaluation Results

Evaluation results are presented in two groups: overall evaluation results (Table 6.19) and evaluation results for lexical differences (Table 6.20). Bold values denote the best results.

It can be seen from the table that improvement in translation quality is not observed for all translation directions. Even if the CC-enhanced system contains all the parallel data that is also present in the baseline MT, the addition of new comparable resources may cause degradation. The reason for this is that the data does not contain true translation equivalents which gives rise to spurious and wrong translations. These cases are less visible to automatic metrics like BLEU but are easily identified by human translators. Therefore, the important point is not only to be able to add more data to the system, but to control the quality of the data which is coming from comparable corpora.

Overall, the results show that the baseline translation quality is very low (1 or 2 on the 5-point translation quality scale on average). The quality of lexical translation choices (where translators were asked to focus on specific words or phrases that are different in the baseline and the enhanced output) is higher. Also, here CC-enhanced systems achieve greater improvement. This shows that the proposed evaluation methodology of focussing on lexical differences is more appropriate to the task of measuring improvements with CC-based data.

On average, across all translation directions, there is improvement in all four areas. The improvement was the smallest for overall translation quality in the

Table 6.19 Evaluation results for *overall translation quality*

Language pair	Baseline (average)	CC-enhanced	Human scores for improvement (%)
News			
de–en	2.269	2.1	–7.45
ro–en	**1.826**	**2.721**	**49.01**
sl–en	**1.869**	**2.025**	**8.35**
hr–en	2.175	2.199	1.10
ro–de	**1.692**	**1.846**	**9.10**
lv–lt	2.157	2.095	–2.87
en–lv	2.04	1.993	–2.30
en–hr	2.107	1.864	–11.53
en–el	2.212	2.362	6.78
de–ro	1.942	1.914	–1.44
el–ro	2.156	2.271	5.33
		Average	4.92
Automotive			
de–en	2.201	2.893	31.44
en–lv	2.177	2.5	14.84
		Average	23.14

Table 6.20 Evaluation results for *lexical choices*: baseline vs CC-enhanced MT

Language pair	Baseline	CC-enhanced	Human scores for improvement (%)
News			
de–en	2.773	2.774	0.04
ro–en	**1.819**	**3.377**	**85.65**
sl–en	2.642	2.867	8.52
hr–en	**2.66**	**2.905**	**9.21**
ro–de	2.351	2.376	1.06
lv–lt	2.614	2.587	−1.03
en–lv	2.564	2.618	2.11
en–hr	2.507	2.118	−15.52
en–el	3.026	3.271	8.10
de–ro	2.399	2.365	−1.42
el–ro	**2.757**	**3.458**	**25.43**
		Average	11.10
Automotive			
de–en	2.628	3.835	45.93
en–lv	2.604	2.956	13.52
		Average	29.72

News domain: 4.92% over the baseline, then lexical improvement in the News domain was 11.1% on average.

In the automotive domain, there is a much higher and consistent improvement for both evaluated systems and in both aspects: overall and lexical quality, in comparison to the broad domain. The improvement for automotive domains was 23.14% and 29.72% for overall and lexical translation quality, respectively.

For the broader News domain, improvement or deterioration depends on the translation direction. Translation into English is always improved. All cases of degradation are for translation into more morphologically complex languages, such as Croatian. The mechanism for this fact is not known and requires further investigation. The results point out that the biggest benefit of CC-enhanced data is achieved for narrow domains and for MT into morphologically simpler languages like English.

6.4 MT Adaptation for Under-Resourced Domains

This section focusses on a very practical aspect of statistical machine translation (SMT)—how a general out-of-domain SMT system can be tailored to a particular domain using data extracted from an in-domain comparable corpus. Particularly, we are dealing with domain-specific terminology and named entities (NEs). We extract terms and named entities from initial parallel training data. These terms and named entities are used to collect a comparable corpus from the Web. Then, we extract

parallel terms from the collected comparable corpus, and finally, we integrate them in the SMT system. The changes in the quality of the adapted SMT system are evaluated in respect to a general out-of-domain baseline system. This section is based on the publication by Pinnis and Skadiņš (2012).

6.4.1 Initial Extraction and Alignment of Terms and Named Entities

The first step in our SMT system adaptation technique is acquisition of in-domain term pairs. Bilingual terminology will allow making the SMT system term-aware and will allow finding better translation candidates for narrow-domain translation tasks. To acquire the term pairs, we use bilingual comparable corpora from the Web.

In order to find important domain specific documents on the Web, we use the small amount of available parallel data sentences (up to two or three thousand parallel sentences) and extract seed terms and named entities for a focussed narrow domain Web crawl. Terms and named entities are monolingually tagged in the parallel in-domain data. For terms, we use the *Tilde's Wrapper System for CollTerm* (*TWSC*) (Pinnis et al. 2012b) and for named entities—*TildeNER* (Pinnis 2012) for Latvian and *OpenNLP*[3] for English. In parallel, a *Moses* phrase table is created from the in-domain parallel data.

Then, the monolingually tagged terms and NEs (in our experiment, 542 unique English and 786 unique Latvian units in total) are bilingually aligned using the *Moses* phrase table. At first, we try to find all symmetric term and named entity phrases in the phrase table that have been monolingually tagged in both languages. We allow only full phrase table entry and term or named entity alignments; that is, a phrase is considered valid only if all tokens from the phrase are identical to tokens of the corresponding term or named entity. In order to also allow inflective form alignments, all tokens of all terms, named entities and phrases are stemmed prior to alignment. This allows finding more translation candidates in cases when some inflective forms have not been tagged as terms, but others have.

Then, we also align terms and named entities that have been tagged by only one of the monolingual taggers. If a phrase is aligned in the phrase table with multiple phrases from the other language, we select the translation candidate that has the highest averaged (source-to-target and target-to-source) translation probability within the phrase table. This step allows finding terms and NEs, which have been missed by one of the monolingual taggers, thus increasing the amount of extracted term and named entity phrases. The alignment method on the in-domain parallel data produced 783 bilingually aligned term and NE phrases.

[3] Apache OpenNLP (available at: http://opennlp.apache.org/).

6.4.2 Comparable Corpora Collection

The second step in our SMT system adaptation technique requires collection of bilingual in-domain comparable corpora from the Web. We use the bilingual terms and NEs that were extracted from the parallel in-domain data as seed terms for focussed monolingual crawling of two monolingual narrow domain Web corpora with the *Focussed Monolingual Crawler* (FMC), which is described in Chap. 3. By using bilingually aligned seed terms, we ensure that the crawled corpora will be comparable and in the same domain for both English and Latvian languages. As the aligned seed terms may also contain out-of-domain or cross-domain term and NE phrases, we apply a ranking method based on reference corpus statistics; more precisely, we use the inverse document frequency (IDF) (Spärck Jones 1972) scores of words from general (broad) domain corpora (e.g. the whole Wikipedia and current news corpora) to weigh the specificity of a phrase. We rank each bilingual phrase using the following equation:

$$R(p_{\text{src}}, p_{\text{trg}}) = \min\left(\sum_{i=1}^{|p_{\text{src}}|} \text{IDF}_{\text{src}}(p_{\text{src}}(i)), \sum_{j=1}^{|p_{\text{trg}}|} \text{IDF}_{\text{trg}}(p_{\text{trg}}(j))\right), \tag{6.1}$$

where p_{src} and p_{trg} denote phrases in the source and target languages and IDF_{src} and IDF_{trg} denote the respective language IDF score functions that return an IDF score for a given token. The ranking method was selected through a heuristic analysis so that specific in-domain term and named entity phrases would be ranked higher than broad-domain or cross-domain phrases. This technique also allows filtering out phrase pairs where a phrase may have a more general meaning in one language but a specific meaning in the other language. After applying a threshold on the ranks, 614 phrase pairs were kept in the seed term list for corpora collection.

In addition to the seed terms, FMC requires seed URLs. In total, 55 English and 14 Latvian in-domain seed URLs were manually collected.

When the seed terms and seed URLs were acquired, a 48-hour focussed monolingual Web crawl was initiated for both languages. The collected English and Latvian corpora were filtered for duplicates, broken into sentences, and tokenised. The statistics of the collected corpora are given in Table 6.21.

Both monolingual corpora were aligned in the document level using the *DictMetric* (Su and Babych 2012) tool described in Chap. 2, which scores document pair comparability and aligns document pairs that exceed a specified comparability score threshold. Executing *DictMetric* on narrow domain comparable corpora may cause over-generation of document pairs; that is, every document from one language can be paired with many documents from the other language. Therefore, we filtered the document alignments so that each Latvian document would be paired with the top three comparable English documents and vice versa, thus creating 81,373 document pairs. The comparable corpus statistics after document level alignment are given in Table 6.22.

Table 6.21 Monolingual automotive domain corpora statistics

Language	Unique documents	Sentences	Tokens	Unique sentences	Tokens in unique sentences
English	34,540	8,743,701	58,526,502	1,481,331	20,134,075
Latvian	6155	1,664,403	15,776,967	271,327	4,290,213

Table 6.22 English-Latvian automotive comparable corpus statistics

Language	Unique documents	Unique sentences	Tokens in unique sentences
English	24,124	1,114,609	15,660,911
Latvian	5461	247,846	3,939,921

6.4.3 *Extraction of Term Pairs from Comparable Corpus*

Once the bilingual comparable corpus is collected, the third step is to extract translated term pairs. Both parts (the Latvian and the English documents), similarly as in the first step, are monolingually tagged with *TWSC*. In this step, we only tag terms as the precision of named entity mapping without a phrase table is well below 90% and would create unnecessary noise in the extracted data for SMT adaptation. Then, by using the document alignment information of the comparable corpus, we map terms bilingually using the *TerminologyAligner* (*TEA*) (Pinnis et al. 2012b) tool with a translation confidence score threshold of 0.7 (with a precision of 90% and higher). In total, 369 in-domain term pairs were extracted from the bilingual comparable corpus.

6.4.4 *Baseline System Training*

We start with the creation of an English–Latvian baseline system using the following data:

- A relatively large out-of-domain parallel corpus. We used the publicly available DGT-TM (Steinberger et al. 2012) English-Latvian parallel corpus (release of 2007). The corpus consists of 804,501 unique parallel sentence pairs and 791,144 unique Latvian sentences. The Latvian part is used for language modelling.
- A small amount of in-domain parallel sentences (up to two or three thousand parallel sentences). In our experiments, we have selected the automotive domain (more precisely, service manuals) as the target domain. The in-domain data are split in two sets—tuning and evaluation. The tuning set and the evaluation set consist of 1745 and 872 unique sentence pairs from the automotive domain. All systems were tuned with minimum error rate training (MERT, Bertoldi et al. 2009) using the in-domain tuning set and evaluated on the evaluation set.

Table 6.23 Baseline system evaluation results

Case sensitive	BLEU	NIST	TER	METEOR
No	10.97	3.9355	89.75	0.1724
Yes	10.31	3.7953	90.40	0.1301

For MT system training, we use the *LetsMT!* (Vasiļjevs et al. 2012) Web-based platform for SMT system creation. The *LetsMT!* platform is built upon the state-of-the-art *Moses* (Koehn et al. 2007) SMT *Experiment Management System* (*EMS*).

Evaluation results for the baseline system using different automatic evaluation methods (BLEU (Papineni et al. 2002), NIST (Doddington 2002), TER (Snover et al. 2006), and METEOR (Banerjee and Lavie 2005)) are given in Table 6.23.

6.4.5 SMT System Adaptation

Following domain adaptation methods suggested in earlier research (Koehn and Schroeder 2007; Lewis et al. 2010; Xu et al. 2007),we start the SMT adaptation task by adding an in-domain language model built using the Latvian monolingual comparable corpus that was collected in the second step. We built the SMT system (named *Int_LM*) using two language models (a general and an in-domain model). Both language models have different weights determined with system tuning (MERT). The in-domain monolingual language model increases SMT quality to 11.3 BLEU points (a relative increase of only 3.0% over the baseline system). We also trained an SMT system (named *In-domain_LM_only*) using only the in-domain language model. The experiment achieved 11.16 BLEU points, which is an increase over the baseline system but also a decrease over the *Int_LM* system. This was expected, as MERT has tuned the in-domain language model to be more important, while the in-domain language model may not contain some general language phrases that are in the broad domain corpus (thus, also interpolation of the two models achieves a higher score).

We continue our experiments by adding the translated term pairs (in total 610) that were extracted from the in-domain tuning set to the parallel data corpus and the corresponding Latvian translations to the in-domain monolingual corpus, from which the SMT system is trained. This simple addition of in-domain term translations to the SMT system (named *Int_LM+T_Terms*) increased the quality to 12.93 BLEU points (a relative increase of 17.8% over the baseline system). After also adding term pairs extracted from the comparable corpus collected from the Web (in total 369 new pairs), the quality of the system (named *Int_LM+T&CC_Terms*) increased to 13.5 BLEU points (a relative increase of 23.1% over the baseline system).

Considering also term banks as possible translated term resources, we extracted 6767 unique in-domain automotive term pairs from EuroTermBank (Rirdance and Vasiljevs 2006).[4] Then, we trained an SMT system (named *Int_LM+ETB_Terms*)

[4]EuroTermBank (http://www.eurotermbank.com/).

with the same parameters as the *Int_LM+T_Terms* system. The system achieved 11.26 BLEU points, which is a decrease in comparison with the *Int_LM* system and much worse than *Int_LM+T&CC_Terms* (the best thus far performing system). The reason for the decrease is fairly simple—term banks, in many cases, provide multiple translation candidates for a single term. This causes ambiguities in the translation model and can result in the selection of the wrong translation hypothesis. To solve this issue (at least partially), the term pairs from the term bank would have to be semantically disambiguated in respect to the required domain so that only the correct in-domain pairs would be used in the SMT system training.

Recent results in MT system adaptation (Ştefănescu et al. 2012) suggest that pseudo-parallel sentence pairs extracted from in-domain comparable corpora and used for SMT system training can significantly improve SMT system quality. Using the same pseudo-parallel sentence extraction tool LEXACC, we extracted 6718 and 678 unique sentence pairs with two parallelism confidence score thresholds of 0.51 and 0.35. These sentence pairs were then added to the available parallel data and the in-domain monolingual corpus. The results after training the SMT systems (named *Int_LM+LEXACC_0.35* and *Int_LM+LEXACC_0.51*) show a decrease in BLEU points (10.75 and 11.08 respectively) in comparison with the *Int_LM* system. After manual analysis of the MT output for *Int_LM+LEXACC_0.35* in comparison with the baseline system, it was evident that the translation quality has decreased because of non-parallel sentence alignments in the LEXACC extracted sentence pairs that cause in-domain term phrase pairs to receive lower weights (translation probability scores) in the translation model. Although in-domain terms in the pseudo-parallel sentences are in many cases paired with correct translations, they are often also paired with incorrect translations, thus creating noise for the translation model. This is not to say that the pseudo-parallel sentences in general do not help to improve SMT quality but rather that, for very narrow and under-resourced domains, where it is difficult to find strongly comparable in-domain corpora in the Web, the results can lower translation quality because of incorrect term translation hypothesis.

So far in our experiments, only the in-domain language model helps to distinguish in-domain translation hypotheses from broad (general) domain hypotheses. Therefore, in the next step, we transformed the *Moses* phrase table of the translation model to an in-domain term-aware phrase table. We do this by adding a sixth feature to the default 5 features that are used in *Moses* phrase tables. The 6th feature receives the following values:

- ‘1’ if a phrase on both sides (in both languages) does not contain a term pair from a bilingual term list. If a phrase contains a term on one side (in one language) but not on the other, it receives the value ‘1’, as such situations indicate about possible out-of-domain (wrong) translation candidates.
- ‘2’ if a phrase contains a term pair from the term list on both sides (in both languages).

In order to find out whether a phrase in the phrase table contains a given term or not, phrases and terms are stemmed prior to comparison. This allows finding inflected forms of term phrases even if those are not given in the bilingual term

list. The sixth feature identifies phrases containing in-domain term translations and allows filtering out out-of-domain (wrong) translation hypotheses in the translation process.

With the described methodology, we transformed phrase tables of the systems *Int_LM+T_Terms* (using the 610 tuning data term pairs) and *Int_LM+T&CC_Terms* (additionally using the 369 term pairs from the comparable corpora) to term-aware phrase tables. After tuning with MERT, two new systems were created. The *Int_LM+T_Terms+6th* system achieves 13.19 BLEU points, and the *Int_LM+T&CC_Terms+6th* system achieves 13.61 BLEU points (a relative increase of 24.1% over the baseline system and the highest measured increase in this experiment). Although the increase in translation quality over the systems without the 6th feature is relatively small, the translations show better translation hypothesis selection for in-domain terminology.

Complete results of the previously described automotive domain systems are shown in Table 6.24 ('*CS*' stands for '*Case-Sensitive*' evaluation).

To show that improvements in SMT quality are also consistent when using larger corpora, we trained a new English–Latvian baseline system (*Big_Baseline*) using 5,363,043 parallel sentence pairs for translation model training and 33,270,743 monolingual Latvian sentences for the language model training. The system was tuned using the same tuning set and evaluated on the same evaluation set as before. The adapted systems (*Big_Int_LM+T&CC_Terms* and *Big_Int_LM+T&CC_Terms+6th*) were built exactly as the *Int_LM+T&CC_Terms* and *Int_LM+T&CC_Terms+6th* systems from the previous experiment. The results (in Table 6.25) show a relative BLEU increase of 8.8% and 14.9% over the baseline for the system without the 6*th* feature and with the 6*th* feature, respectively. As more data creates higher ambiguity, the 6*th* feature allows increasing the results significantly more than in the previous experiment. This shows the potential of the method when applied on larger corpora.

The results of the experiments show that integration of terminology within SMT systems, even with simple techniques (adding translated term pairs to the parallel data corpus or adding an in-domain language model), can achieve improvement in SMT system quality by up to 23.1% over the baseline system. Transformation of translation model phrase tables into term-aware phrase tables can boost the quality up to 24.1% over the baseline system, mostly because of wrong translation candidate filtering in the translation process.

The experiments also show that the usage of pseudo-parallel sentence pairs extracted from weakly comparable narrow-domain corpora and term pairs acquired from term banks without a sophisticated term sense disambiguation, and semantic analysis of the source text may not result in increased SMT quality due to the added noise in in-domain translation hypotheses.

Table 6.24 English–Latvian automotive domain SMT system adaptation results

System	BLEU	BLEU (CS)	NIST	NIST (CS)	TER	TER (CS)	METEOR	METEOR (CS)
Baseline	10.97	10.31	3.9355	3.7953	89.75	90.40	0.1724	0.1301
Int_LM	11.30	10.61	3.9606	3.8190	89.74	90.34	0.1736	0.1312
In-domain_LM_only	11.16	10.52	3.9447	3.8074	89.31	89.92	0.1726	0.1305
Int_LM+T_Terms	12.93	12.12	4.2243	4.0598	88.58	89.32	0.1861	0.1418
Int_LM+T&CC_Terms	13.50	12.65	4.2927	4.1105	88.86	89.70	0.1878	0.1443
Int_LM+ETB_Terms	11.26	10.52	3.9456	3.7882	89.43	90.04	0.1737	0.1290
Int_LM+LEXACC_0.35	10.75	10.09	3.7935	3.6682	90.31	90.86	0.1646	0.1229
Int_LM+LEXACC_0.51	11.08	10.28	3.9132	3.7709	90.23	90.78	0.1706	0.1286
Int_LM+T_Terms+6th	13.19	12.36	4.2657	4.0962	88.84	89.62	0.1876	0.1439
Int_LM+T&CC_Terms+6th	13.61	12.78	4.3514	4.1747	88.54	89.32	0.1920	0.1469

Table 6.25 English–Latvian automotive domain big SMT system adaptation results

System	BLEU	BLEU (CS)	NIST	NIST (CS)	TER	TER (CS)	METEOR	METEOR (CS)
Big_Baseline	15.85	15.00	4.84	4.69	73.80	75.12	0.2098	0.1651
Big_Int_LM +T&CC_Terms	17.24	16.12	5.00	4.83	72.16	73.59	0.2163	0.1717
Big_Int_LM +T&CC_Terms +6th	18.21	17.08	5.15	4.96	70.22	71.62	0.2191	0.1747

6.5 MT Adaptation to a Narrow Domain in Case of Resource-Rich Languages

The objective of this contribution is to evaluate improvements achieved by using data from comparable corpora for tuning Machine Translation systems to narrow domains for languages that are usually classified as resource-rich. The language direction chosen was German to English, and the automotive domain, in particular the sub-domain on transmission/gearbox technology, was selected as an example for a narrow domain. In order to assess the effect of domain adaptation on MT systems with different architecture, both data-driven (SMT) and knowledge-driven (RBMT) systems were evaluated.

6.5.1 Evaluation Objects: Narrow-Domain-Tuned MT Systems

The evaluation objects are two versions of an MT system: a baseline version, without domain tuning, and an adapted version, with domain tuning. Their comparison shows whether or not domain adaptation can improve MT quality.

The evaluation objects were created as follows:

1. For the baseline systems, on the RBMT side, the system of Linguatec's '*Personal Translator*' PT (V.14) was used which is a rule-based MT system based on the IBM slot-filler grammar technology (Aleksić and Thurmair 2011). It was taken as out of the box and installed on a standard PC. On the SMT side, a baseline Moses system with standard parallel data (Europarl, JRC, etc.), which was presented in Sect. 6.2.3, and some initial comparable corpus data as collected in ACCURAT (Skadiņa et al. 2010) were used.
2. For adaptation of the baseline systems, data was collected from the automotive domain. This data was collected by crawling sites of automotive companies that are active in the transmission field (like ZF, BASF, Volkswagen and others). This data was strongly comparable. It was then aligned and cleaned manually. Some sentence pairs were set aside for testing, and the rest were given to the two

systems for domain adaptation as development and test sets. The resulting narrow-domain automotive corpus has about 42,000 sentences for German-to-English.

For the SMT system, domain adaptation was done by adding these sentences to the training and development sets and building a new SMT system.

In the case of rule-based technology, domain adaptation is more complicated as it involves terminology creation which is the main means of adaptation. Therefore, the following steps were taken:

- Creation of a phrase table with GIZA++ and MOSES; for this, the phrase tables of the SMT adapted system were taken; phrase tables of only in-domain data were also built but turned out to be not as efficient as the ones from baseline plus in-domain data.
- Extraction of bilingual terminology candidates from these phrase tables using the P2G (Phrase-Table-to-Glossary) tool; this resulted in a list of about 25,000 term candidates.
- Preparation of these candidates for dictionary import, including creation of part-of-speech and gender annotations, removal of already existing entries, resolution of conflicts in transfers, etc.; the final list of imported entries was about 7100 entries.
- Creation of a special 'automotive' user dictionary which can be added to the system dictionary in cases where texts from the automotive domain are translated.

This procedure is described in detail by Thurmair and Aleksić (2012).

The result of these efforts was four test systems for German-to-English, tuned for the automotive domain with the same adaptation data:

- *SMT-base:* DFKI-baseline system trained with only baseline data
- *SMT-adapted:* DFKI-adapted system trained with baseline plus in-domain data
- *RBMT-base:* PT-baseline as the out-of-the-box RBMT system
- *RBMT-adapted:* trained with an additional 'automotive' dictionary.

6.5.2 Evaluation Data

For evaluation, a set of sentence pairs was extracted from the collected strongly comparable automotive corpora. In total, about 1500 sentences were taken for tests, with one reference translation each.

The sentences represent 'real-life' data; they were not cleaned or corrected, just like the training data. So they contain spelling mistakes, segmentation errors and other types of noise. This fact, of course, affects the translation quality for the adapted systems.

6.5.3 *Evaluation Methodology*

Several methods can be applied for the evaluation of MT results. **Automatic** comparison (called BLEU in Fig. 6.8) is the predominant paradigm in the world of SMT. So BLEU (Papineni et al. 2002) and/or NIST (Doddington 2002) scores can be computed for different versions of MT system output.

While such scores seem to measure inner-system quality changes with some degree of reliability, they do not seem to measure translation quality (Babych and Hartley 2008), do not conform to the judgment of human evaluators (Hamon et al. 2006), and are sensitive towards an SMT system architecture in disfavour of rule-based approaches. Therefore, projects like WMT do not use them as the only measure of quality any more (Callison-Burch et al. 2009; Bojar et al. 2018) but also ask for human judgment.

Comparative evaluation (called COMP in Fig. 6.8) is possible between two systems as well as between two versions of the same system. It simply asks whether or not one translation is better/equal/worse than the other.

While this approach can find which of two systems has an overall better score, it cannot answer the question of what the real quality of the two systems is: '*Equal'* can mean that both sentences are perfect or that both are unusable.

Therefore, **absolute** evaluation (called ABS in Fig. 6.8) is required to determine the quality of a given translation. This procedure looks at one translation of a source sentence at a time and determines its accuracy (how much content has been transported to the target language) and fluency (how correct/grammatical is the produced target sentence), following the FEMTI paradigm (King et al. 2003).

Post-editing evaluation (called POST in Fig. 6.8) reflects the task-oriented aspect of evaluation (Popescu-Belis 2008). It measures the distance of an MT output to a human (MT-post-edited) output, either in terms of time (answering the question of how productive a system can be as compared, e.g. to a human-only translation) or in terms of the keystrokes needed to produce a human-corrected translation from an MT-raw translation (HTER: Snover et al. 2006, 2009).

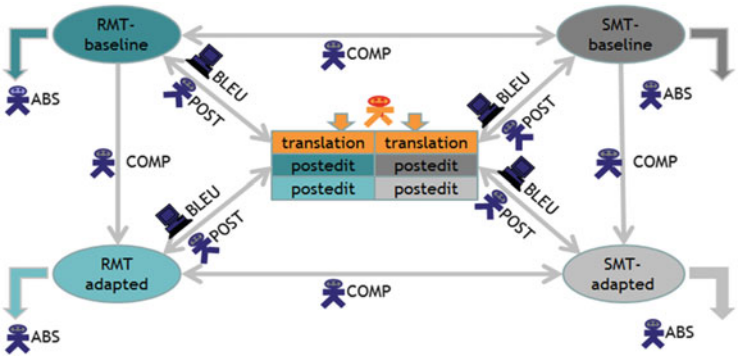


Fig. 6.8 Evaluation options

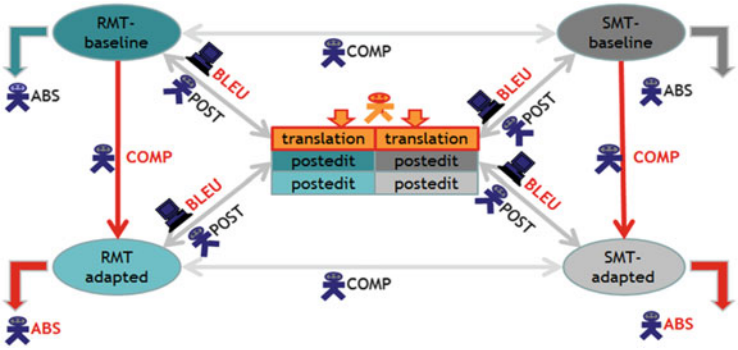


Fig. 6.9 Evaluation in narrow domain task

Post-editing evaluation adds reference translations to the evaluation process.

In our narrow domain task, the following evaluation methods were used, cf. Fig. 6.9:

- Automatic evaluation of the four systems (SMT-baseline and SMT-adapted, RBMT-baseline and RBMT-adapted) using BLEU and NIST scores.
- Comparative evaluation of the pairs (SMT-baseline versus SMT-adapted and RBMT-baseline versus RBMT-adapted); this would produce the core information of how much the systems can improve.
- Absolute evaluation of the systems (SMT-adapted and RBMT-adapted), to gain insight into the translation quality and, consequently, the potential acceptance of such systems for real-world use.

Other forms of evaluation were not included in the evaluation task. However, to have a complete picture, the other ABS and COMP directions were evaluated as well, but with less effort (1 tester only).

6.5.4 Evaluation Tools

To perform the evaluations, a special toolset was created for the non-automatic tasks. The toolset is called '*Sisyphos-II*' (for details see Chap. 8: Appendix) and consists of three components:

- 'ABS' to support absolute evaluation, using two four-point scales. For adequacy, the options are {*full content conveyed* | *major content conveyed* | *some parts conveyed* | *incomprehensible*}. For fluency, the options are {*grammatical* | *mainly fluent* | *mainly nonfluent* | *rubble*}.
- 'COMP' to support comparative evaluation of two MT outputs, using a four-point scale. Comparison options are {*first translation better* | *both equally good* | *both equally bad* | *second translation better*}.

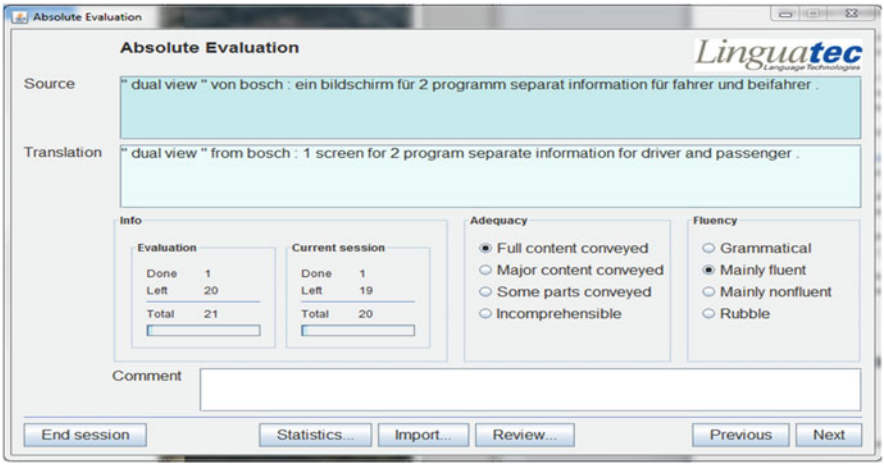


Fig. 6.10 Screen shot of evaluation tool (ABS)

- 'POST' to support post-editing evaluation, by measuring the post-editing time from the first display of the sentence until the pressing of the [Save] button (in seconds) and allowing HTER computing.

The tools are stand-alone tools that can be given, for example, to a freelance translator. Evaluation data is presented to the users by a special GUI in random order, and evaluation results are collected in an XML file which is the basis for evaluation.

An example screenshot of the tool is shown in Fig. 6.10. Each time a 4-point scale is presented, users select one of the options in both areas.

6.5.5 Evaluation Results

Three evaluators were used to do the translations, all of them good speakers of English with a bit of MT background. Each of them evaluated a random subset of the 1500 sentence test set, consisting of at least 500 sentences for each of the COMP evaluations (SMT-adapted versus SMT-baseline and RBMT-adapted versus RBMT-baseline) and at least 300 sentences for each ABS evaluation (SMT-adapted and RBMT-adapted). More than 5000 evaluation points were collected this way.

6.5.5.1 Automatic Evaluation

The automatic evaluation for the German–English pair was done on the basis of BLEU scores. The results are shown in Table 6.26.

Table 6.26 BLEU scores for SMT and RBMT

	SMT	RBMT
Baseline	17.36	16.08
Adapted	22.21	17.51
Improvement	**4.85**	**1.43**

For both systems, there is an increase in BLEU; it is more moderate for the RBMT than for the SMT system. However, it is known that BLEU is biased towards SMT systems.

6.5.5.2 Comparative Evaluation

For the German–English pair, three testers were used, all of them good speakers of English with a bit of MT background.

Of the 1500 test sentences, three testers inspected randomly selected subsets, in total about 2000 sentences. As the tool does not offer identical sentences for evaluation, these cannot be differentiated for 'equally good' versus 'equally bad'. If these two categories are merged into one ('*equal*'), the following results were achieved (Table 6.27).

The data shows that the domain adaptation results in an improvement of 5% for both types of systems. It is a bit more (5.1%) for the SMT than for the RBMT (4.7%). The result is consistent among the testers: all of them state an improvement of the adapted versions, and all of them see a higher improvement for the SMT than for the RBMT.

It may be worthwhile to notice that in the RBMT evaluation, a large proportion of the test sentences (nearly 60%) came out identical in both versions, and the changes were rather small (17% of the sentences). In the SMT system, nearly no sentence came out unchanged, and the variance in comparison was between 36% and 51% (depending on the testers).

In a sideline evaluation, a comparison was made between the baseline versions of SMT and RBMT and their adapted versions (Table 6.28).

The result shows that the RBMT quality is considered significantly better than the SMT quality. The main reason for this seems to be that the SMT German–English frequently eliminates verbs from sentences: for example *Silber wird in der Medizin seit Jahrhunderten wegen seiner antimikrobiellen Wirkung geschätzt und eingesetzt. => silver in medicine centuries for its antimicrobial effect and*. This effect has already been observed with other SMT outputs.

It should be noted, however, that the distance between the systems is smaller in the adapted versions than in the baseline versions (by 3%).

6.5.5.3 Absolute Evaluation

The absolute evaluation was done to assess how usable the resulting translation would be after the system was adapted. A total of 1100 sentences, randomly selected from the 1500 test base, were inspected by three testers. The adequacy and fluency

Table 6.27 Comparative evaluation baseline versus adapted for SMT and RBMT

	SMT					RMT				
	Total inspected	Base better	Both equal	Adapted better	Improvement (%)	Total inspected	Base better	Both equal	Adapted better	Improvement (%)
Tester 1	1049	235	514	300	6.20	1501	91	1237	173	5.46
Tester 2	510	130	228	152	4.31	503	33	417	53	3.98
Tester 3	501	82	319	100	3.59	501	34	418	49	2.99
Total	2060	447	1061	552	**5.10**	2505	158	2072	275	**4.67**

Table 6.28 Comparative Evaluation SMT / RBMT, baseline and adapted

Total inspected	SMT better	Both equal	RBMT better	In percent
501	47	170	284	47.3
489	38	203	260	44.3

was measured for each sentence on a scale of 1–4. Table 6.29 gives the result (lower average scores mean better quality).

It can be seen that testers evaluate the SMT somewhat between 'mainly' and 'partially' fluent/comprehensible and the RBMT close to 'mainly' fluent/comprehensible. If the percentage level of 1/2 of the evaluations is taken, the SMT adequacy is rated with 36.6% and fluency with 53.04%, while both adequacy (64.97%) and fluency (77.50%) are significantly higher in RBMT. All testers agree in their evaluation and have similar average results. The better score for RBMT may result from the 'missing verb' problem mentioned above.

It could be worthwhile to mention that the often-heard opinion that SMT produces more fluent output than RBMT cannot be corroborated with the evaluation data here: the RBMT output is clearly considered to be more fluent than the SMT output (1.8 vs. 2.3).

An absolute evaluation was also done for the two baseline systems, however with only one tester. The results are given in Table 6.30.

The figures indicate that system adaptation improves the accuracy of both of the SMT (from 2.86 baseline to 2.62 adapted), and it seems to reduce the fluency of the RBMT (from 1.48 baseline to 1.80 adapted); a further error analysis would be required to find out why. The other results (RBMT accuracy and SMT fluency) seem unchanged.

As far as the inter-rater agreement is concerned, the test set-up made it difficult to compute it: all testers used the same test set but tested only a random subset of it. So there are only a few data points common to all testers (only 20 in many cases). For those, only weak agreement could be found (with values below 0.4 in Cohen's kappa, Table 6.31). However, all testers show consistent behaviour in the evaluation and came to similar overall conclusions, as has been explained above.

6.5.6 Conclusion

Figure 6.11 gives all evaluation results. The main conclusion is that all evaluation methods indicate an improvement of the adapted versions over the baseline versions.

Automatic evaluation:

- For SMT, the BLEU score increases from 17.36 to 22.21.
- For RBMT, the BLEU score increases from 16.08 to 17.51.

Comparative evaluation:

- For SMT, an improvement of 5.1% was found.
- For RBMT, an improvement of 4.67% was found.

Table 6.29 Absolute evaluation for SMT-adapted and RBMT-adapted systems

	Inspected	Adequacy						Fluency					
		1: full	2: most	3: partial	4: none	Average	% of 1+2	1: fluent	2: mainly	3: partly	4: none	Average	% of 1+2
		SMT adapted											
Tester 1	500	89	119	284	8	2.42	41,60	87	163	238	12	2.35	50,00
Tester 2	302	52	48	156	46	2.65	33,11	97	97	93	15	2.09	64,24
Tester 3	301	59	37	77	128	2.91	31,89	116	25	31	129	2.57	46,84
Total	1103	200	204	517	182	2.62	**36,63**	300	285	362	156	2.34	**53,04**
		RMT adapted											
Tester 1	501	210	127	150	14	1.94	67,27	197	189	100	15	1.87	77,05
Tester 2	300	106	99	80	15	2.01	68,33	164	89	42	5	1.63	84,33
Tester 3	301	149	25	55	72	2.17	57,81	180	35	34	52	1.86	71,43
Total	1102	465	251	285	101	2.02	**64,97**	541	313	176	72	1.80	**77,50**

Table 6.30 Absolute evaluation of the baseline systems

	Inspected	Adequacy						Fluency					
		1: full	2: most	3: partial	4: none	average	% of 1+2	1: fluent	2: mainly	3: partly	4: none	average	% of 1+2
SMT-baseline	301	57	51	69	124	2.86	**35,88**	136	22	46	97	2.35	**52,49**
RMT baseline	301	165	15	61	60	2.05	**59,80**	222	37	18	24	1.48	**86,05**

Table 6.31 Kappa for inter-tester agreement

	SMT COMP	RMT COMP	SMT-ABS adequacy	SMT-ABS fluency	RMT-ABS adequacy	RMT-ABS fluency
Records inspected	1189	1102	846	846	851	851
Common data points	115	39	21	21	21	21
Common evaluation	46	11	5	4	3	3
Kappa	0.38	0.26	0.22	0.18	0.17	0.11

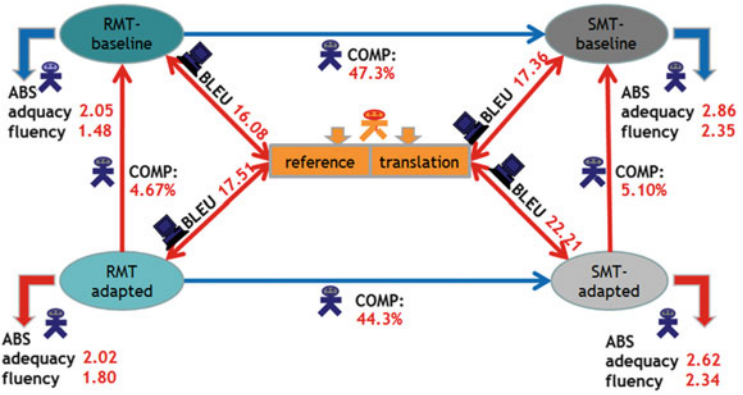


Fig. 6.11 Evaluation summary (BLEU, COMP, ABS)

Absolute evaluation:

- For SMT, adequacy improved from 2.86 to 2.62, and fluency improved slightly from 2.35 to 2.34.
- For RBMT, adequacy improved from 2.05 to 2.02, and only fluency decreased from 1.48 to 1.8.

The improvement is more significant for the SMT system than for the RBMT. This may be due to the fact that the RBMT baseline system has better COMP and ABS scores, though lower BLEU scores, than the SMT baseline.

For SMT improvement, Pecina et al. (2012) report improvements between 8.6 and 16.8 BLEU (relative) for domain adaptation. Our results here are in line with these findings.

6.6 Application of Machine Translation (MT) in Web Authoring

Authoring is defined as a process of creating and editing documents, especially multimedia documents, for other end-users. Authoring systems or tools are software packages used for creating and packaging content and have been applied in multimedia (Bulterman and Hardman 2005; Scherp and Boll 2005; Deltour and Roisin 2006), e-learning (Watson et al. 2010; Capuano et al. 2009), adaptive e-learning (Bontchev and Vassileva 2009), mobile learning (Mugwanya and Marsden 2010), tutoring (Escudero and Fuentes 2010), interactive digital storytelling (Müller et al. 2010) and lately digital gaming (Mehm et al. 2012). Their purpose is to assist less technically skilled users to produce multimedia, structure the content without special expertise and speed up the process of content creation by streamlining and automating common tasks.

Authoring tools are also widely used in professional publishing for a variety of publication types such as books, documentations, reports, articles or presentations. The simplest authoring tools are simple preformatted Word document templates or templates with added scripting. More sophisticated publishing systems, platforms and desktop publishing applications enable advanced functionalities and may support different roles in the system, for example writers and editors, and even provide the ability to collaborate between these roles. Web authoring is a sub-domain of authoring and, in its broadest sense, means authoring content online.

The way people use the Web has changed considerably. Web users are provided with new means of communication (blogs, tweets, instant messaging), collaboration (wiki, forums), and sharing of multimedia content. From the late 1990s, user-oriented Web authoring started changing the publishing game. Previously, special technical knowledge was required to create and publish content on the Web, but the emergence of Web authoring tools has brought publishing closer to the wider public. Web authoring has facilitated content creation by non-technical users: all a user needs is an Internet connection and a browser, everything else is available online. Furthermore, Web authoring tools started providing simple interfaces that guide and assist at every step of content creation by enabling users to develop websites in desktop publishing (or similar) format by generating underlying HTML code for the layout based on the user's design. Users can typically toggle between graphical design and HTML code.

Web authoring systems that are accountable for most user-generated content are popular blogging platforms (such as WordPress[5] and Blogger[6]), micro-blogging platforms (such as Twitter[7] and Tumblr[8]) and wiki platforms (Désilets et al. 2006)

[5]WordPress: http:www.wordpress.com

[6]Blogger: http://www.blogger.com/

[7]Twitter: http://twitter.com

[8]Tumblr: http://www.tumblr.com

based on the MediaWiki[9] open source project. We consider social media networks as a part of Web authoring because users can create multimedia profile pages, post text, images, video or a combination of all of them and create content in this way.

The availability and popularity of Web authoring tools have affected many areas, including translation of online content. The amount of data has expanded drastically, the number of languages in which online content is produced has increased, and even English content is frequently written in *casual* English. Use of Web authoring (and the Web in general) has shifted from producers to users who connect and exchange ideas and opinions, but the language barrier still remains. Garcia (2009) stated that *the amount of content contributed by producers and users exceeds translation industry's capacity to cope* and that the translation industry *cannot keep pace with an environment that puts a premium on cheapness and speed*.

Today, researchers working in the field of natural language processing, specifically information retrieval and machine translation (MT), are faced with the seemingly low-hanging fruit of the large amounts of online data that is provided through user-oriented media (social networking sites, blogging and microblogging services). In reality, this comes with a price, bringing new issues they have not faced before.

To illustrate the vastness of online content, we provide figures published by the most popular services: Tumblr claims to have 277 million blogs in 16 languages and 128.7 billion posts[10]; bloggers at WordPress.com write in 120 languages, publish 41.7 million new posts and 60.5 million new comments per month, and 409 million people view more than 15.5 billion pages each month; Twitter supports more than 35 languages and has more than 320 million active users[11]; Wikipedia has 5 million content articles in English only, is available in 291 languages, and more than 19,000 articles are added to it every day.[12] These numbers give us a perspective on the amount of content that users generate. Researchers have to deal with scaling-up demands and the robustness required by the need to understand *casual written English, which often does not conform to rules of spelling, grammar and punctuation* (Clark and Araki 2011). Online content is ever more resembling conversation with loose grammar rules, intentional or unintentional misspellings, acronyms and jargon which affects the accuracy of natural language processing, information retrieval and translation.

6.6.1 The Role of Translation and MT in Web Authoring

Web authoring is a multi-language online environment. With the aid of Web authoring, we now have the means to search for information globally or locally,

[9]MediaWiki: http://www.mediawiki.org/

[10]https://www.tumblr.com/about, accessed in January, 2016.

[11]https://about.twitter.com/company, all numbers approximate as of September 30, 2015.

[12]http://stats.wikimedia.org/EN/ReportCardTopWikis.htm, accessed in January, 2016.

while social media tools help us to distribute it to various communities that speak other languages and, in this way, to broaden readership and/or the pool of customers. Social tools also allow us to build and support international communities or networks communicating in languages other than English or our mother tongue.

Yet there is still one important obstacle in the way of reaching the full potential: the lack of quality translation, especially for under-resourced languages and narrow domains.

The role and the importance of translation in Web authoring can be viewed from different perspectives, and we will focus on two that are closely related to Web authoring, namely

- Content creator perspective, which includes companies or organisations publishing content and public publishing user-generated content via social media tools and platforms (blogging, microblogging, wikis, forums).
- Content consumer perspective which involves readers searching for information or opinions.

From the content creator perspective, we distinguish between localisation and internationalisation. In this context, localisation is considered as a translation from English (or any other major world language) into other languages, including under-resourced languages and minority languages; for internationalisation, the direction of translation is exactly the opposite: translating from a (minor) language into one of the major languages. While localisation is more typical for larger companies, internationalisation is more frequent for smaller companies or bloggers wanting a larger audience. For example large international companies are expanding their business to other countries and want to localise their Web pages, product documentation or user support. On the other hand, small (local) companies want to reach out and provide their content, especially Web sites, in one of the more frequently used languages.

From the end-user perspective, translation plays an important role in discovering new knowledge, finding information about products, people and events. Translation direction can range from English to any language, between any language pair, even from an under-resourced one to another under-resourced one.

Quality is an important factor in the translation in Web authoring but is not the only one. When readers are just interested in the broader meaning of the text (so-called *gisting*), the quality is less important than the speed of translation and its accessibility to the public.

The traditional translation model includes the aid of computer translation software and is carried out by professional translators or bi-lingual experts. Most translation software was based on translation memories and terminology databases, thus being less suitable for the needs of translation in Web authoring. The translation model has changed with the rise of ‘software as a service’. Translation services, particularly the ones based on MT, are now more affordable, available to the general public, and suitable for integration into authoring tools, more easily than ever before.

From the content consumer perspective, MT (as a service) seems to be the only good option for translating user-generated content. In general, readers do not

understand more than two or three languages, cannot afford a human translator and do not want to buy expensive translation software. Content producers see MT as important for similar reasons; they want translation to be as fast and as cheap as possible.

Translation quality is a major factor in Web authoring, but it is not the most important factor in some cases. When readers are just interested in the broader meaning of the text (so-called gisting), the quality is less important than the speed of translation, public availability and price—especially if the service is free.

The reasons observed by Hutchins (2003) regarding why machine translation is needed are still valid today, also for the domain of Web authoring. We added the lack of support for under-resourced languages to the following list of reasons:

- The amount of generated content is too large for human translators.
- The demand for increase in the volume and speed of translation throughput (translation needed now, not in a few days from now) is growing.
- Top quality translation is not always needed, neither is human assistance/post-editing.
- People communicate and generate content in a large number of under-resourced languages, which are usually not supported by traditional translation models or are not easily accessible.

An additional reason might also be the need for integration of translation service into other tools used for research, exploration and discovery. For example, corporations that use multi-lingual online collaboration environments need translation tools that seamlessly integrate into collaborative tools, such as chats and support forums, for more effective use of online content (replacing the need to copy-paste content into one of the freely available MT services) and appealing to an even broader population of users.

While translation services are valuable standalone products, they are more valuable if they can be integrated to complement the functionalities in other tools that users work with, such as Web browsers, document editors, phone applications and Web authoring tools and platforms.

6.6.2 Characteristics and Requirements for Translation in Web Authoring

In terms of demographic factors (such as geographic location, age, gender, household income or levels of education), Web authoring is widespread mostly due to the emergence of new online tools and platforms that make Web authoring easier than ever before. Translation in Web authoring is needed, and it has to meet the requirements set by its users. Web users are very demanding—they want translation to be fast and free.

The role of human translators in Web translation has changed: the old traditional translation model of translate-edit-proofread, involving human (professional) translators, has been replaced with other, more flexible models—not collaborative models, but rather MT-assisted models (Garcia 2009).

A number of MT-assisted models are already implemented in translation services, such as Google Translate or Microsoft Translator, and are being widely used. However, before using them, we have to consider the following question: is MT really the answer to every translation problem in Web authoring? Considering the current state of MT, the answer to this question is not encouraging. Several factors have to be considered before deciding to use MT:

- Role of the user: content consumer or content producer.
- Volume of material: the larger the volume, the more prohibitive the cost of human translation becomes.
- Frequency with which material changes: it may be less practical to continually use human translators for material that changes frequently.
- Domain and purpose of content and its translation: informational, persuasive, legal, etc. The more important it is that the translation is accurate and fluent, the less likely it is that MT should play a role, at least not without post-editing.
- Speed of translation: MT will always provide faster results.
- Languages involved: related languages and languages that are very commonly used will translate the best. For some language pairs, it might be hard to find a human translator and be much easier to use Google Translate, even if the translation quality is not good.

Balancing these factors is an important part in making the decision of whether to apply MT or not. The translation quality of MT tools depends on the domain and the languages involved, and therefore, it is important to choose the tool that is best suited for the problem, as not all tools produce good results with all language pairs.

Current translation techniques that are applied to Web authoring depend on the type of platform and the content. Popular content management systems (CMS) allow editing of multi-lingual content in parallel at the time of writing or soon afterwards. In most cases, authors translate text themselves on the fly. For Web authoring platforms, such as WordPress, several plug-ins provide the functionality of parallel text editing of multiple languages.

The collaborative translation model is used for wiki projects such as Wikipedia and Wikitravel. They both provide content in multiple languages, and translation is performed by multiple (anonymous) bi-lingual authors/editors. However, some might not truly consider this to be multi-lingual translation, because neither Wikipedia nor Wikitravel provides (exactly) the same content in different languages. Research by Désilets et al. (2006) has refuted the commonly held assumption that Wikipedia contents are parallel, they claim that '*[t]hese sites are in fact a collection of parallel communities that produce content about overlapping sets of topics in different languages, with little if any synergy across languages*'. There is also a

project translatewiki.net[13] which is a wiki localisation platform for translation communities, language communities and free and open source projects. The platform incorporates translation memory from the translate toolkit,[14] Yandex Translate[15] and Microsoft Translator which assist in collaborative translations.

Crowdsourcing translation is a similar model to the collaborative one, but it is not limited to the wiki environment. Facebook crowdsourced its translation, and according to the results, this could not be performed any faster or better even if it had applied the usual localisation processes (Garcia 2009). Twitter is using a similar approach by inviting its users to help with localisation of the platform. Some of the other 'big players' have also used crowdsourcing to translate the parts of their content that they considered to be suitable for this kind of translation. For example Google used crowdsourcing to translate its interface into many minority languages. It also uses the '*Suggest a better translation*' feature in Google Translate through which crowds contribute to improvements of its SMT engine (Garcia 2009).

6.6.2.1 MT in Web Authoring

Traditional translation models involving professional translators and the workflow of using only translation memories are less suitable for fast growing (in terms of the volume of publications) and expanding (in terms of new languages) Web authoring domain, especially if compared with MT systems.

Today's widely spread use of MT on the reader side of Web authoring can be credited mostly to Google Translate and Microsoft Bing Translator. They are typically used at the time when content is consumed. Readers have the option of using a Web browser with an integrated translation service available via a toolbar or installation of the tool as an extension for their favourite Web browser. When readers visit a certain website, if the content language differs from the default language set in the browser, it is either automatically translated or translated on demand by pressing a button on the toolbar. Some bloggers put special translation widgets directly on their blogs, so readers do not have to install toolbars and can use the translation widget instead.

The situation is similar for content creators. For example bloggers can use several translation plug-ins which are available for the most popular blogging platforms. These plug-ins usually use Google Translate or Bing Translator to translate text in the Web editor, and they put it directly back in the editor so that the author can post-edit it before publishing. The microblogging platform Twitter has integrated Bing Translator API into their Web–user interface to provide machine translation between more than 40 language pairs.

[13]Translatewiki project: http://translatewiki.net/wiki/

[14]Translate Toolkit & Pootle: http://translate.sourceforge.net/wiki/

[15]Yandex Translate: http://company.yandex.com/technologies/translation.xml

Wiki projects are a special case in regard to machine translation. Wikipedia took the initiative in the form of the Wikipedia Machine Translation Project.[16] As Wikipedia is a multi-lingual resource, the 'Wikipedia consensus is that an unedited machine translation, left as a Wikipedia article, is worse than nothing.'

6.6.2.2 Translating User-Generated Content

Web authoring covers online content by professionals and amateurs. The latter is also known as user-generated content. It is usually produced in a more conversational manner, most of it is in poor or non-standard quality, it can be produced by non-native speakers, native speakers can non-deliberately introduce typos or deliberately stray from spelling norms to achieve special linguistic goals or effects (Jiang et al. 2012).

Carrera et al. (2009) acknowledged that user-generated content is suitable for MT, but most such content usually remains untranslated. Jiang et al. (2012) built a number of statistical SMT engines for a Middle East–based social networking provider based on user-generated content and identified several problems in the process.

Flournoy and Rueppel (2010) describe how MT could be used in Adobe for translating user-generated content either for a community translation initiative, in which MT output can be presented as pre-translations for the members of the community, or for translating valuable resources such as Q&A, tutorials and product reviews. While high-quality MT is preferred in both cases, it is not required.

Evaluation of MT is a separate research field, and we will not delve far into it. In many studies including the one by Hovy et al. (2002), the following aspects of translation quality are taken into consideration: fluency (lexically and syntactically well-formed sentences), fidelity (translation does not change the meaning/semantics of the input), price, system extensibility and coverage (specialisation of the system to the domains of interest). More recent research studies about translating user-generated content were mainly interested in fidelity. Fidelity is measured on a limited scale by human judges rating how well a system's output expresses the content of the same portion of the source text or even ideal human translations (Hovy et al. 2002). Mitchell and Roturier (2012) conducted a pilot study, based on a previous study by Roturier and Bensadoun (2011), that examined the perceived quality of MT in terms of comprehensibility among members of an online community forum and the ways users interact with the MT content. Even though the study had a low response rate, the results have shown that the MT output was *comprehensible slightly more often than not*.

Translation direction in user-generated content is primarily from English to other languages; otherwise, it varies and can include any language pair. Open-source MT

[16] https://meta.wikimedia.org/wiki/Machine_translation

translation attempts are an opportunity for minor languages, and the objective behind is also to 'de-minorise' translation (Forcada 2006).

From the usage of MT in Web authoring, we can conclude that it is mostly used and useful for obtaining a general understanding of content. If it is used for content creation, then the content is post-edited, because the quality of MT is not good enough.

6.6.2.3 Defining Requirements for Using MT in Web Authoring

When defining requirements for MT in Web authoring, we have to consider both content characteristics and the factors that we mentioned at the beginning of this section. Major factors affecting the quality of MT in Web authoring are

- **Domain specificity:** MT systems based on texts in one domain perform badly in another domain; may work well for general translations but not for specific ones, or vice versa; work well for EU-related documents, but perform really bad for general translations.
- **Lack of resources**: SMT systems rely solely on quantitative information extracted by systems trained on vast amounts of data. What if there are no vast amounts of data for systems to be trained on, as in cases of under-resourced languages or narrow domains?
- **Casual English**: problems include rapidly changing out-of-dictionary slang, short-forms and acronyms, punctuation errors or omissions, phonetic spelling, misspelling for verbal effect and other intentional misspelling and recognition of out-of-dictionary named entities. Use of casual English in social media poses a problem: casual media needs pre-processing before translation, but this might not prove to be feasible for bloggers (Clark and Araki 2011).

Requirements for MT on translating cross-language social media were described by Carrera et al. (2009) in the context of social media analysis. They noted that an MT system would need to be designed for

- Large-scale, real-time translation
- Preservation of meaning (which should be good enough for gisting)
- Robustness, especially in light of errors in linguistic formalisation

Flournoy and Rueppel (2010) provide additional requirements valid for translating user-generated content:

- Low to medium translation quality is required.
- MT has to be able to deal with various subject matters.
- There is no need for special security (no need for non-disclosure agreements as in the case of formal documents with business secrets).
- The most frequently occurring language pairs are EN→XX, but others can also occur, such as XX→YY.
- Input is of varied, uncontrolled quality.

Almost all researchers agree on the biggest issue that all MT systems are facing: the quality of translation output. If we ignore the fact that most MT systems prior to Google Translate were either rule-based or assisted by translation memories, one of the more important causes for poor quality is the discrepancy between the corpora that MT systems are trained on and the texts that MTs are used on. For example, Google Translate works best for short subject–verb–object sentences, such as driving directions, simple instructions or simple scientific sentences. It also does quite well for gisting of websites, but is unlikely to provide adequate translations for short-lived colloquialisms, new words or word plays. Using Google Translate to directly translate social media content without post-editing is not recommended. The same goes for legal drafts, descriptions of medical equipment, political texts, safety applications and legal documents.

6.6.3 MT Systems Enhanced with Comparable Corpora in Web Authoring: A Use Case

The quality of translation services for under-resourced languages and narrow domains still falls behind the quality for more widely used language pairs (e.g. English, German, French, Arabic, Chinese) and more general domains. MT systems enhanced with comparable corpora aim to close this gap and improve the quality of translation for these under-resourced languages and narrow domains.

Comparable corpora are easier to obtain than parallel corpora, but, in comparison to other comparable corpora in major languages, they are still not in abundance. Content from narrow domains faces a similar situation: translation services trained on general texts produce poor results when used on texts from narrow domains and make it hard to train a quality SMT due to the lack of parallel corpora.

The blogosphere is a good example that combines both of the above-mentioned issues, which were addressed in the ACCURAT project. We evaluated the use of MT systems enhanced with comparable corpora (henceforth CC-enhanced MT) for Web authoring in a use case involving blog posts in Slovenian, Croatian and German. CC-enhanced MT was used as an intermediate step between content written in one of the under-resourced source languages and Zemanta's recommendation engine that is available via a Web service. Currently, the recommendation service works only for texts in English and does not return good results for texts in other languages, so this was an opportunity to use translation before the recommendation step returns more relevant related contents.

6.6.3.1 Evaluation Process and Datasets

As a blogger writes a blog post, Zemanta's recommendation engine analyses the text and suggests related contents that the blogger can use to enrich the blog post. Our goal was to find out whether using MT before sending text to the recommendation

engine results in better suggested related articles. Although the recommendation engine returns related articles, images and keywords, we focussed on related articles only.

Evaluation of results was done in Zemanta's internal evaluation system by two human evaluators. We collected blog posts and online news articles in Slovenian, Croatian and German, 100 texts per language (Table 6.32), and put them through the recommendation engine to obtain the 10 best suggested related articles per text. Texts in the source language were translated using two translation methods—baseline and CC-enhanced MT—and translations were sent to the recommendation engine to get suggestions again.

The evaluation cycle is illustrated in Fig. 6.12. After the recommendation engine returned suggested related articles, two human evaluators assessed each suggested article, from the blogger's perspective, and assigned a score to it ranging from zero (will not use) to three (definitely will use). These scores were used to calculate the precision@10 metric which considers only the top ten relevant documents with the highest precision score.

Table 6.32 Evaluation sets of texts

Language pair	Number of files	Avg. text length (words)
Slovenian–English	100	238.8
German–English	100	242.7
Croatian–English	100	202.7

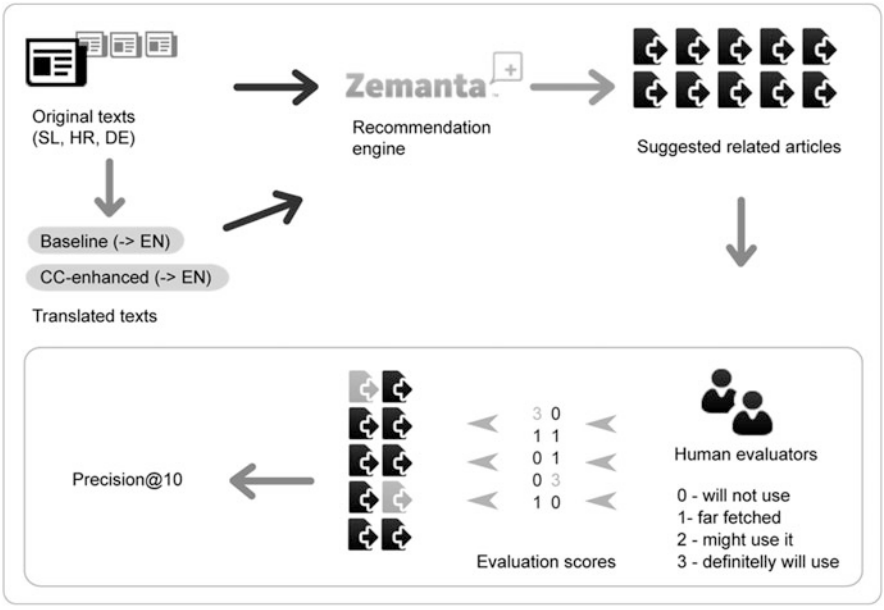


Fig. 6.12 Evaluation process used in the use case

The length of blog posts and online news articles can vary from a few sentences to full-length and detailed reviews. Datasets contain texts of length between 200 and 300 words. This is a typical length of a blog post. This amount of text is enough for the recommendation engine to return valid suggestions and for a translation service to provide translations in a reasonable amount of time. Texts included topics from business/economy, politics, technology, sports and living.

6.6.3.2 Results and Discussion

We calculated precision@10 for all language pairs in three different batches and summarised the average precision in the tables below. Batches are labelled *Original*, *Baseline* and *CC-enhanced*. In the first batch (*Original*), untranslated texts were fed directly into the recommendation engine; in the second batch (*Baseline*), texts were first translated using baseline MT and then fed into the recommendation engine. In the last and third batch (*CC-enhanced*), texts were translated into English using CC-enhanced SMT and then again fed into the recommendation engine. The inter-rater agreement for two human evaluators was moderate which was rather expected due to the high level of subjectivity in the blogosphere.

Table 6.33 shows the average precision for different language pairs. For the Slovenian–English language pair, usage of baseline MT in comparison to original texts improved precision by 11%, and usage of CC-enhanced SMT improved it by 15%.

We can also see that using translation for German texts shows even greater improvement: nearly 20% for baseline MT and 24% for CC-enhanced SMT in comparison to original texts.

Unfortunately, we were not able to use CC-enhanced SMT for Croatian texts, and therefore, we only have precision for baseline MT. Using this translation method improved results by 11%.

We tested the hypothesis that results obtained by using the recommendation engine on MT translated texts does not differ significantly from results obtained by using the engine on untranslated (original) texts using the unpaired t-test. We tested both translation methods, and the difference for all translation pairs was significant on 95% confidence level. Table 6.34 contains mean, SD and P-values for both translation methods. Although the CC-enhanced method improved precision for Slovenian and German, the difference between both translation methods is not significant.

Table 6.33 Average precision for different language pairs

Dataset	Slovenian–English	German–English	Croatian–English
Original	0.153	0.141	0.201
Baseline	0.265	0.344	0.314
CC-enhanced	0.299	0.379	–

Table 6.34 Mean, SD and P-value for language pairs and translation methods

Value	Baseline			CC-enhanced	
	Slovenian	German	Croatian	Slovenian	German
Mean	0.265	0.344	0.313	0.301	0.381
SD	0.245	0.248	0.312	0.243	0.298
P-value	0.0006	0.0001	0.0110	0.000	0.000

Table 6.35 Average, minimum and maximum translation times for baseline method and CC-enhanced method

	Avg translation time (s)		Min time (s)		Max time(s)	
Language pair	Baseline	CC-enhanced	Baseline	CC-enhanced	Baseline	CC-enhanced
Slovenian–English	111.98	133.99	61.56	30.95	365.05	423.88
German–English	172.71	186.98	92.16	122.06	273.62	304.94
Croatian–English	78.92	–	31.70	–	122.47	–

Table 6.36 Percentage of translated words for baseline translation method and CC-enhanced method

	Avg. words			% translated words	
Language pair	Original	Baseline	CC-enhanced	Baseline	CC-enhanced
Slovenian–English	238.8	232.0	225.2	59	76
German–English	242.7	209.8	217.1	73	74
Croatian–English	202.7	183.8	–	73	–

Although we were more concerned with the criteria of fidelity, we also measured the translation time for whole datasets (Table 6.35) and the percentage of translated words for 10 randomly translated texts per translation method and language pair (if available), as summarised in Table 6.36. While average translation times were roughly similar for Slovenian and German, the difference for minimum and maximum time was quite large, but we did not investigate it any further at this point.

Next, we analysed ten randomly selected translated texts per translation method and a language pair (if available) for percentage of translated words. Results are summarised in Table 6.36.

Interestingly, the percentage of translated words when using the CC-enhanced method for Slovenian texts increased by 17% which could indicate that using comparable corpora can improve translation for under-resourced languages. The reason why we were interested what amount of text was actually translated was because of the way that the recommendation engine works. It is based on keyword search and named entity recognition, and, if these are not translated, the results might not be good, that is actually relevant to the original text.

Part of the engine is also based on statistic approaches in order to recognise new trending concepts and named entities that can appear in blog posts and news

overnight. The training and learning cycle of a machine translation service has to be short enough to be able to incorporate them into a translation model so that they get properly translated. Because the CC-enhanced method also depends on news crawling, extracting parallel phrases and training translation workers on this data, the learning cycle is longer than ideal (which would be daily integration of new concepts), but it might still be fast enough to be useful when used for Web authoring.

6.6.4 Conclusion

After investigating the importance of translation and specifically MT for Web authoring, we came to the conclusion that translation is much needed and desired on all levels of Web authoring (professional and amateur) and from all perspectives (content creators or content consumers). The quality of MT output is, with some exceptions, still not high enough to be used without human intervention and post-editing, and this is even truer for texts in under-resourced languages and narrow domains. Users in Web authoring use MT output mostly for gisting or as a basis for post-editing.

We described characteristics of Web authoring and user-generated content as well as several requirements that have to be met before successfully applying MT to Web authoring problems.

In our use case, we have shown that MT works well as an intermediate layer between content in under-resourced languages and Web services such as a recommendation engine for related articles which supports only the English language. The recommendation engine returned better suggestions, that is more of the articles were actually related when MT was used to translate texts before feeding them into the recommendation engine.

Even though there are still several obstacles on the path of full utilisation of MT in Web authoring, it already benefits users by helping them bridge the language gap when they are either searching for information, participating in social media networks or enriching their blog posts that are written in an under-resourced language.

6.7 Systems for Computer-Aided Translation

Although the quality of MT systems has been criticised a lot, due to a growing pressure on efficiency and cost reduction, MT receives more and more interest from the localisation industry.

Different aspects of post-editing and machine translatability have been researched since the 1990s (a comprehensive overview has been provided by O'Brien (2005)). Several productivity tests have been performed in translation and localisation industry settings at Microsoft (Schmidtke 2008), Adobe (Flournoy and Duran 2009), Autodesk (Plitt and Masselot 2010), and Tilde (Skadiņš et al. 2011). In all these tests,

authors report productivity increase. However, in many cases, they also indicate significant performance differences in the various translation tasks. Increase of the error score for translated texts is also reported.

As the localisation industry experiences a growing pressure on efficiency and performance, some developers have already integrated MT in their computer-assisted translation (CAT) products: for example, SDL Trados, ESTeam TRANSLATOR and Kilgrey memoQ.

In this section, we demonstrate that, for language pairs and domains where there is not enough parallel data available,

1. In-domain comparable corpora can be used to increase translation quality.
2. If comparable corpora are large enough and can be classified as strongly comparable, then the trained SMT systems applied in the localisation process increase the productivity of human translators.

We present our work on English–Latvian SMT system adaptation to the IT domain: building a comparable corpus, extracting semi-parallel sentences and terminological units from the comparable corpus and adapting the SMT system to the IT domain with the help of the extracted data. We describe evaluation results demonstrating that data extracted from comparable corpora can significantly increase the BLEU score over a baseline system. Results from the application of the adapted SMT system in a real-life localisation task are presented, showing that SMT usage increased the productivity of human translators by 13.6%. This section is based on the publication by Pinnis et al. (2013).

6.7.1 Collecting and Processing a Comparable Corpus

For our experiment, we used an English–Latvian comparable corpus containing texts from the IT domain: software manuals and Web crawled data (consisting of IT product information, IT news, reviews, blogs, user support texts including software manuals, etc.). The corpus was acquired in an artificial fashion in order to simulate a strongly comparable narrow domain corpus (i.e. a corpus containing overlapping content in a significant proportion).

To get more data for our experiments, we used two different approaches in the creation of a comparable corpus. Thus, the corpus consists of two parts. The first part contains documents acquired from different versions of software manuals of a productivity software suite split into chunks of less than 100 paragraphs per document and aligned at document level with the *DictMetric* tool, which is decribed in Chap. 2. As a very large number of alignments were produced, we filtered document pairs so that, for each source and target language document, there were no more than the top three alignments (for both languages separately) included.

The second part consists of an artificially created strongly comparable corpus from parallel data that is enriched with Web crawled non-comparable and weakly comparable data. The parallel data was split into random chunks from 40 to 70 sentences per

document and randomly polluted with sentences from the Web crawled data from 0 to 210 sentences. The Web corpus sentences were injected in random positions in English and Latvian documents separately, thus heavily polluting the documents with non-comparable data. The Web crawled data was collected using the *Focussed Monolingual Crawler* (FMC), which is decribed in Chap. 3. The Web corpus consists of 232,665 unique English and 96,573 unique Latvian sentences. The parallel data contained 1,257,142 sentence pairs before pollution.

The statistics of the English–Latvian comparable corpus are given in Table 6.37. Note that the second part of the corpus accounts for 22,498 document pairs.

The parallel sentence extractor *LEXACC*, which is decribed in Chap. 5, was used to extract semi-parallel sentences from the comparable corpus. Before extraction, texts were pre-processed—split into sentences (one sentence per line) and tokenised (tokens separated by a space).

Because the two parts of our corpus differ in terms of comparable data distribution and the comparability level, different confidence score thresholds were applied for extraction. The threshold was selected by manual inspection of extracted sentences so that most (more than 90%) of the extracted sentence pairs would be strongly comparable or parallel.

Table 6.38 shows information about data extracted from both parts of the corpus using the selected thresholds.

We applied the ACCURAT Toolkit to acquire in-domain bilingual term pairs from the comparable corpus following the process thoroughly described in Pinnis et al. (2012b). At first, the comparable corpus was monolingually tagged with terms, and then terms were bilingually mapped. Term pairs with the confidence score of mapping below the selected threshold were filtered out. In order to achieve a precision of about 90%, we selected the confidence score threshold of 0.7. The statistics of both the monolingually extracted terms and the mapped terms are given in Table 6.39.

The term pairs were further filtered so that for each Latvian term, only those English terms having the highest mapping confidence scores would be preserved. We used the Latvian term to filter term pairs, because Latvian is a morphologically richer language and multiple inflective forms of a word, in most cases, correspond to a single English word form (although this is a 'rude' filter, it increases the precision of term mapping to well over 90%).

Table 6.37 Comparable corpus statistics

English documents	Latvian documents	Number of aligned document pairs	Number of aligned document pairs after filtering
27,698	27,734	385,574	45,897

Table 6.38 Extracted semi-parallel sentence pairs

Corpus part	Threshold	Unique sentence pairs
First part	0.6	9720
Second part	0.35	561,994

Table 6.39 Term tagging and mapping statistics

	Unique monolingual terms		Mapped term pairs	
Corpus part	English	Latvian	Before filtering	After filtering
First part	127,416	271,427	847	689
Second part	415,401	2,566,891	3501	3393

As can be seen in Table 6.39, only a small part of the monolingual terms were mapped. However, this amount of mapped terms was sufficient for SMT system adaptation as described below. It should also be noted that, in our adaptation scenario, translated single-word terms are more important than multi-word terms as the adaptation process of single-word terms partially covers also the multi-word pairs that have been missed by the mapping process.

6.7.2 Building SMT Systems

We used the LetsMT! platform (Vasiļjevs et al. 2012) based on the Moses tools (Koehn et al. 2007) to build three SMT systems: the baseline SMT system (trained on publicly available parallel corpora), the intermediate adapted SMT system (in addition, data extracted from the comparable corpus was used) and the final adapted SMT system (in-domain terms integrated). All SMT systems have been tuned with minimum error rate training (MERT) (Bertoldi et al. 2009) using in-domain (IT domain) randomly selected tuning data containing 1837 unique sentence pairs.

For the English–Latvian baseline system, the DGT-TM parallel corpora of two releases (2007 and 2011) were used. The corpora were cleaned in order to remove corrupt sentence pairs and duplicates. As a result, for training of the baseline system, a total of 1,828,317 unique parallel sentence pairs were used for translation model training, and a total of 1,736,384 unique Latvian sentences were used for language model training.

In order to adapt the SMT system for the IT domain, the extracted in-domain semi-parallel data (both sentence pairs and term pairs) were added to the parallel corpus used for baseline SMT system training. The whole parallel corpus was then cleaned and filtered with the same techniques as for the baseline system. The statistics of the filtered corpora used in SMT training of the adapted systems (intermediate and final) are shown in Table 6.40.

Table 6.40 shows that there was some sentence pair overlap between the DGT-TM corpus and the comparable corpora content. This was expected as DGT-TM covers a broad domain and may contain documents related to the IT domain. For language modelling, however, the sentences that overlap in general domain and in-domain monolingual corpora have been filtered out from the general domain monolingual corpus. Therefore, the DGT-TM monolingual corpus statistics between the baseline system and the adapted system do not match.

Table 6.40 Training data for adapted SMT systems

	Parallel corpus (unique pairs)	Monolingual corpus
DGT-TM (2007 and 2011) sentences	1,828,317	1,576,623
Sentences from comparable corpus	558,168	1,317,298
Terms form comparable corpus	3594	3565

After filtering, a translation model was trained from all available parallel data, and two separate language models were trained from the monolingual corpora:

- Latvian sentences from the DGT-TM corpora were used to build the general domain language model.
- The Latvian part of the extracted semi-parallel sentences from the in-domain comparable corpus was used to build the in-domain language model.

To make in-domain translation candidates distinguishable from general domain translation candidates, the phrase table of the domain adapted SMT system was further transformed to a term-aware phrase table (Pinnis and Skadiņš 2012) by adding a sixth feature to the default five features used in Moses phrase tables. The following values were assigned to this sixth feature:

- '2', if a phrase in both languages contained a term pair from the list of extracted term pairs.
- '1', if a phrase in both languages did not contain any extracted term pair; if a phrase contained a term only in one language, but not in both, it received '1' as this case indicates possible out-of-domain (wrong) translation candidates.

In order to find out whether a phrase contained a given term or not, every word in the phrase and the term itself was stemmed. Finally, the transformed phrase table was integrated back into the adapted SMT system.

6.7.3 Automatic and Comparative Evaluation

The evaluation of the baseline and both adapted systems was performed with four different automatic evaluation metrics: BLEU, NIST, TER, and METEOR on 926 unique IT domain sentence pairs. Both case-sensitive and case-insensitive evaluations were performed. The results are given in Table 6.41.

The automatic evaluation shows a significant performance increase of the improved systems over the baseline system in all evaluation metrics. Comparing two adapted systems, we can see that making the phrase table term-aware (*Final adapted system*) yields further improvement over intermediate results after just adding data extracted from comparable corpora (*Intermediate adapted system*). This is due to better terminology selection in the fully adapted system. As terms comprise only a certain part of texts, the improvement is limited.

For the system comparison, we used the same test corpus as for automatic evaluation and compared the baseline system against the adapted system. Figure 6.13

Table 6.41 Automatic evaluation results

System	Case-sensitive?	BLEU	NIST	TER	METEOR
Baseline	No	11.41	4.0005	85.68	0.1711
	Yes	10.97	3.8617	86.62	0.1203
Intermediate adapted system	No	56.28	9.1805	43.23	0.3998
	Yes	54.81	8.9349	45.04	0.3499
Final adapted system	No	56.66	9.1966	43.08	0.4012
	Yes	55.20	8.9674	44.74	0.3514

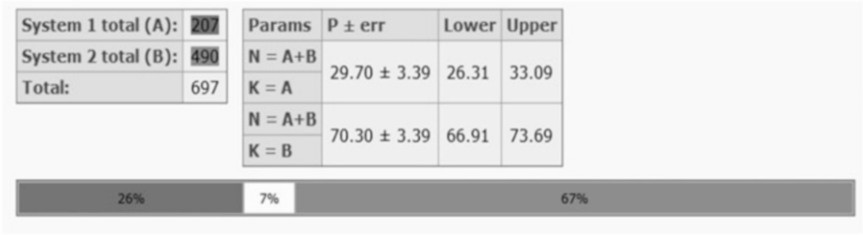


Fig. 6.13 System comparison by total points (System 1—baseline, System 2—adapted system)

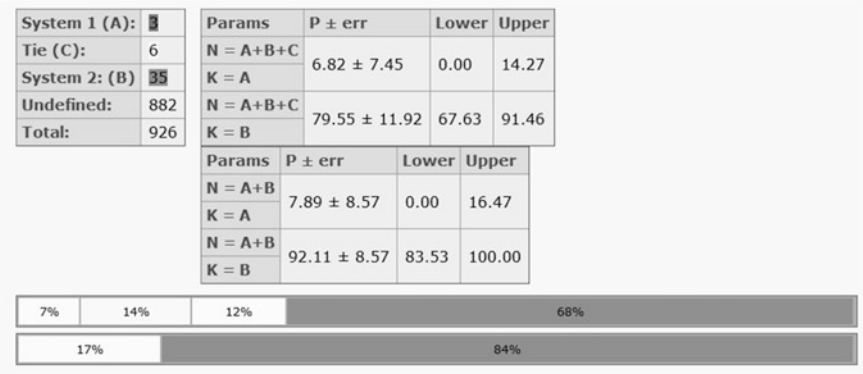


Fig. 6.14 System comparison by count of the best sentences (System 1—baseline, System 2—adapted system)

summarises the human evaluation results using the evaluation method described in Skadiņš et al. (2010). From 697 evaluated sentences, output of the improved SMT system was chosen as a better translation in 490 cases (70.30 ± 3.39%), while users preferred the translation of the baseline system in 207 cases (29.70 ± 3.39%). This allows us to conclude that for IT domain texts, the adapted SMT system provides better translations than the baseline system.

Figure 6.14 illustrates the evaluation on sentence level: we can reliably say that the adapted SMT system provides a better translation for 35 sentences, while users preferred the translation of the baseline system for only 3 sentences. It must be noted

that, in this figure, we present the results only for those sentences for which there was a statistically significant preference to the first or second system by the evaluators.

6.7.4 Evaluation in Localisation Task

The main goal of this evaluation task was to evaluate whether integration of the adapted SMT system in the localisation process allows increasing the output of translators in comparison to the efficiency of manual translation. We compared productivity (words translated per hour) in two real life localisation scenarios:

- Translation using only translation memories (TMs).
- Translation using suggestions of TMs and the SMT system that is enriched with data from the comparable corpus.

6.7.4.1 Evaluation Set-Up

For tests, 30 documents from the IT domain were used. Each document was split into two parts. The length of each part of a document was 250 to 260 adjusted words on average, resulting in 2 sets of documents with about 7700 words in each set.

Three translators with different levels of experience and average performance were involved in the evaluation cycle. Each of them translated 10 documents without SMT support and 10 documents with integrated SMT support. The SDL Trados translation tool was used in both cases.

The results were analysed by editors who had no information about the techniques used to assist the translators. They analysed average values for translation performance (translated words per hour) and calculated an error score for translated texts. The individual productivity of each translator was measured and compared against his or her own productivity. The average productivity for all of the translators has been calculated using the following formula (6.2):

$$\text{Productivity (scenario)} = \frac{\sum_{\text{Text}=1}^{N} \text{Adjusted words(Text, scenario)}}{\sum_{\text{Text}=1}^{N} \text{Actual time(Text, scenario)}}. \tag{6.2}$$

Usage of MT suggestions in addition to TMs increased the productivity of the translators on average from 503 to 572 words per hour (see Table 6.42). There were significant differences in the results of different translators from a performance increase by 35.4% to decreased performance by 5.9% for one of the translators. Analysis of these differences requires further studies, but they are most likely caused by working patterns and skills of individual translators.

Table 6.42 Results of productivity evaluation

Translator	Scenario	Actual productivity	Productivity increase or decrease (%)	Standard deviation of productivity
Translator 1	TM	493.2	35.39	110.7
	TM +MT	667.7		121.8
Translator 2	TM	380.7	13.02	34.2
	TM +MT	430.3		38.9
Translator 3	TM	756.9	−5.89	113.8
	TM +MT	712.3		172.0
Average	TM	503.2	13.63	186.8
	TM +MT	571.9		184.0

Table 6.43 Quality grades based on error scores

Superior	Good	Mediocre	Poor	Very poor
0. . .9	10. . .29	30. . .49	50. . .69	>70

According to the standard deviation of productivity in both scenarios (186.8 without MT support and 184.0 with MT support), there were no significant performance differences in the overall evaluation. However, each translator separately showed higher differences in translation performance when using the MT translation scenario.

Editors also calculated an error score for every translation task by counting identified errors and applying a weighted multiplier based on the severity of the error type:

$$\text{ErrorScore} = \frac{1000}{n} \sum_i w_i e_i, \tag{6.3}$$

where n is the number of words in the translated text, e_i is the number of errors of type i, w_i is a coefficient (weight) indicating the severity of type i errors. Depending on the error score, the translation is assigned a translation quality grade (*Superior*, *Good*, *Mediocre*, *Poor*, or *Very poor*) (Table 6.43).

6.7.4.2 Results

The overall error score (shown in Table 6.44) increased for one out of three translators. Although the total increase in the error score for all translators combined was from 24.9 to 26.0 points, it still remained at the quality evaluation grade ‘Good’.

Table 6.44 Localisation task error score results

Translator	Scenario	Accuracy	Language quality	Style	Terminology	Total error score
Translator 1	TM	6.8	8.0	6.8	1.6	23.3
	TM +MT	9.9	14.4	7.8	4.1	36.3
Translator 2	TM	8.2	10.1	11.7	0.0	30.0
	TM +MT	3.8	11.7	7.6	1.5	24.6
Translator 3	TM	4.6	9.5	7.3	0.0	21.4
	TM +MT	3.0	8.3	6.0	0.8	18.1
Average	TM	6.5	9.3	8.6	0.5	24.9
	TM +MT	5.4	11.4	7.1	2.1	26.0

6.7.5 Discussion

The results of our experiment demonstrate that it is feasible to adapt SMT systems for a particular domain with the help of comparable data and integrate such SMT systems for highly inflected under-resourced languages into the localisation process.

The use of the English-Latvian domain-adapted SMT suggestions (trained on comparable data) in addition to the translation memories led to the increase of translation performance by 13.6% while maintaining an acceptable ('*Good*') quality of the translation. However, our experiments also showed a relatively high difference in translator performance changes (from −5.89% to +35.39%), which suggests that the experiment should be carried out with more participants for more justified results. It would also be useful to further analyse the correlation between the regular productivity of a translator and the impact on productivity by adding MT support.

Error rate analysis shows that in overall usage of MT suggestions decreased the quality of translation in two error categories (language quality and terminology). At the same time, this degradation is not critical, and the result is still acceptable for production purposes.

References

Abdul-Rauf, S., & Schwenk, H. (2009). On the use of comparable corpora to improve SMT performance. *Proceedings of the 12thConference of the European Chapter of the Association for Computational Linguistics* (pp. 16–23), Athens, Greece.

Abdul-Rauf, S., & Schwenk, H. (2011). Parallel sentence generation from comparable corpora for improved SMT. *Machine Translation, 25*(4), 341–375.

Aleksić, V., & Thurmair, Gr. (2011). Personal Translator at WMT 2011. *Proceedings of the WMT Edinburgh*, UK.

Babych, B., & Hartley, A. (2008). Sensitivity of automated MT evaluation metrics on higher quality MT output: BLEU vs task-based evaluation methods. *Proceedings of LREC*, Marrakech.

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association of Computational Linguistics (ACL 2005)*, June 2005, Michigan.

Bertoldi, N., Haddow, B., & Fouet, J. B. (2009). Improved minimum error rate training in moses. *The Prague Bulletin of Mathematical Linguistics, 91*, 7–16.

Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Koehn, P., & Monz, C. (2018). *Findings of the 2018 Conference on Machine Translation (WMT18)* (pp. 272–303). WMT (shared task) 2018.

Bontchev, B., & Vassileva, D. (2009). Courseware authoring for adaptive e-learning. *Proceedings of the 2009 International Conference on Education Technology and Computer (ICETC '09)* (pp. 176–180). IEEE Computer Society, Washington, DC.

Bulterman, D. C. A., & Hardman, L. (2005). Structured multimedia authoring. *ACM Transactions on Multimedia Computing, Communication and Applications, 1*, 89–109.

Callison-Burch, Ch., Koehn, Ph., Monz, Ch., & Schroeder, J. (2009). Findings of the 2009 workshop on statistical machine translation. *Proceedings of the 4th Workshop on SMT*, Athens.

Capuano, N., Pierri, A., Colace, F., Gaeta, M., & Mangione, G. R. (2009). A mash-up authoring tool for e-learning based on pedagogical templates. *Proceedings of the First ACM International Workshop on Multimedia Technologies for Distance Learning (MTDL '09)* (pp. 87–94). ACM, New York, NY.

Carrera, J., Beregovaya, O., & Yanishevsky, A. (2009). *Machine Translation for Cross-Language Social Media*. Accessed April 23, 2013 from http://www.promt.com/company/technology/pdf/machine_translation_for_cross_language_social_media.pdf

Clark, E., & Araki, K. (2011). Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English. *Procedia – Social and Behavioral Sciences, 27*, 2–11.

Deltour, R., & Roisin, C. (2006). The limsee3 multimedia authoring model. *Proceedings of the 2006 ACM Symposium on Document Engineering (DocEng '06)* (pp. 173–175). ACM, New York, NY.

Désilets, A., Gonzalez, L., Paquet, S., & Stojanovic, M. (2006). Translation the Wiki Way. *The Conference Wiki of the 2006 International Symposium on Wikis*. Odense, Denmark.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proceedings of the Second International Conference on Human Language Technology Research (HLT 2002)* (pp. 138–145), San Diego.

Escudero, H., & Fuentes, R. (2010). Exchanging courses between different Intelligent Tutoring Systems: A generic course generation authoring tool. *Knowledge-Based Systems, 23*(8), 864–874.

Flournoy, R., & Duran, C. (2009). Machine translation and document localization at Adobe: From pilot to production. *Proceedings of the Twelfth Machine Translation Summit*, Ottawa, Canada.

Flournoy, R., & Rueppel, J. (2010). One technology: Many solutions. *AMTA 2010: The Ninth Conference of the Association for Machine Translation in the Americas*, Denver, CO, 6p.

Forcada, M. (2006). Open-source machine translation: An opportunity for minor languages. *5th SALTMIL Workshop on Minority Languages* (pp. 1–7).

Garcia, I. (2009). Beyond translation memory: Computers and the professional translator. *The Journal of Specialised Translation, 12*, 199–214.

Hamon, O., Popescu-Belis, A., Choukri, K., Dabbadie, M., Hartley, A., Mustafa El Hadi, W., et al. (2006). CESTA: First conclusions of the technolangue mt evaluation campaign. *Proceedings of the LREC*, Genova, Italy.

Hewavitharana, S., & Vogel, S. (2008). Enhancing a statistical machine translation system by using an automatically extracted parallel corpus from comparable sources. *Proceedings of the Workshop on Comparable Corpora, LREC'08* (pp. 7–10).

Hovy, E., King, M., & Popescu-Belis, A. (2002). Principles of context-based machine translation evaluation. *Machine Translation, 17*(1), 43–75.

Hutchins, J. (2003). *Machine translation and computer-based translation tools: What's available and how it's used*. A New Spectrum of Translation Studies. University of Valladolid.

Intel Corporation. (2012). *Enabling Multilingual Collaboration through Machine Translation (IT@Intel White Paper).* Accessed March 30, 2013 from http://www.intel.com/content/www/us/en/it-management/intel-it-best-practices/enabling-multilingual-collaboration-through-machine-translation.html

Irvine, A., & Callison-Burch, Ch. (2013). Combining bilingual and comparable corpora for low resource machine translation. *Proceedings of the Eighth Workshop on Statistical Machine Translation* (pp. 262–270).

Jiang, J., Way, A., & Haque, R. (2012). Translating user-generated content in the social networking space. *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA-2012)*, San Diego, CA.

King, M., Popescu-Belis, A., & Hovy, E. (2003). FEMTI: Creating and using a framework for MT evaluation. *Proceedings of MT Summit*, New Orleans.

Koehn, P., & Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., et al. (2007). Moses: Open source toolkit for statistical machine translation. *Annual Meeting of the Association for Computational Linguistics (ACL), Demonstration Session*.

Lewis, W., Wendt, C., & Bullock, D. (2010). Achieving domain specificity in SMT without overt siloing. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*.

Lu, B., Jiang, T., Chow, K., & Tsou, B. K. (2010). Building a large English-Chinese parallel corpus from comparable patents and its experimental application to SMT. *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora: From Parallel to Non-parallel Corpora* (pp. 42–48), Valletta, Malta.

Mehm, F., Reuter, C., Göbel, S., & Steinmetz, R. (2012). Future trends in game authoring tools. *Entertainment Computing-ICEC 2012* (Vol. 7522, pp. 536–541),Springer, Heidelberg.

Mitchell, L., & Roturier, J. (2012). Evaluation of machine-translated user generated content: A pilot study based on user ratings. *Proceedings of the 16th EAMT Conference*, 28–30 May 2012, Trento, Italy.

Mugwanya, R., & Marsden, G. (2010). Mobile learning content authoring tools (MLCATs): A systematic review. *Proceedings E-Infrastructures and E-Services on Developing Countries – Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (pp. 20–31).

Müller, W., Iurgel, I., Otero, N., & Massler, U. (2010). Teaching English as a second language utilizing authoring tools for interactive digital storytelling. *ICIDS'10 Proceedings of the Third Joint Conference on Interactive Digital Storytelling* (pp. 222–227).

Munteanu, D., & Marcu, D. (2006). Improving machine translation performance by ex-ploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Najeh, H., Kolovratnik, D., Vaeyrynen, J., Steinberger, R., & Varga,D. (2014). DCEP-digital corpus of the European parliament. *Proceedings of LREC 2014 (Language Resources and Evaluation Conference)* (pp. 3164–3171).

O'Brien, S. (2005). Methodologies for measuring the correlations between post-editing effort and machine translatability. *Machine Translation, 19*(1), 37–58.

Och, F. J. (2003) Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (Vol. 1, pp. 160–167).

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. *Proceedings of ACL-2002: 40th Annual meeting of the Association for Computational Linguistics* (pp. 311–318).

Pecina, P., Toral, A., Papavassiliou, V., Prokopidis, P., & van Genabith, J. (2012). Domain adaptation of statistical machine translation using web-crawled resources: A case study. *Proceedings of the EAMT 2012*, Trento, Italy.

Pinnis, M. (2012). Latvian and lithuanian named entity recognition with TildeNER. *Proceedings of LREC 2012*, 21–27 May, 2012, Istanbul, Turkey.

Pinnis, M., & Skadiņš, R. (2012). MT Adaptation for Under-Resourced Domains –What Works and What Not. *Baltic HLT2012*.

Pinnis, M., Ion, R., Ştefănescu, D., Su, F., Skadiņa, I., Vasiļjevs, A., et al. (2012a). Toolkit for multi-level alignment and information extraction from comparable corpora. *Proceedings of ACL 2012, System Demonstrations Track*, Jeju Island, Republic of Korea, 8–14 July 2012.

Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., & Gornostay, T. (2012b). Term extraction, tagging and mapping tools for under-resourced languages. *Proceedings of the 10th Conference on Terminology and Knowledge Engineering*, Madrid, Spain.

Pinnis, M., Skadiņa, I., & Vasiļjevs, A. (2013). Domain adaptation in statistical machine translation using comparable corpora: Case study for english latvian it localisation. *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics CICLING 2013*.

Plitt, M., & Masselot, F. (2010). A Productivity Test of Statistical Machine Translation Post-Editing in a Typical Localisation Context. *The Prague Bulletin of Mathematical Lin-Guistics, 93*, 7–16.

Popescu-Belis, A. (2008). Reference-based vs. task-based evaluation of human language technology. Proceedings of LREC.

Rirdance, S., & Vasiljevs, A. (Eds.). (2006). *Towards consolidation of European terminology resources. Experience and recommendations from EuroTermBank project*. Riga: EuroTermBank Consortium.

Roturier, J., & Bensadoun, A. (2011). Evaluation of MT systems to translate user generated content. *Proceedings of Machine Translation Summit XIII* (pp. 244–251), Xiamen, China.

Scherp, A., & Boll, S. (2005). Context-driven smart authoring of multimedia content with xSMART. *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA '05)* (pp. 802–803). ACM, New York, NY.

Schmidtke, D. (2008). *Microsoft office localization: Use of language and translation technology*. Available at: http://www.tm-europe.org/files/resources/TM-Europe2008-Dag-Schmidtke-Microsoft.pdf

Schwenk, H., & Koehn, P. (2008). Large and diverse language models for statistical machine translation. *IJCNLP2008*.

Skadiņa, I., Aker, A., Giouli, V., Tufis, D., Gaizauskas, R., Mieriņa, M., et al. (2010). A collection of comparable corpora for under-resourced languages. *Proceedings of the Fourth International Conference Baltic HLT 2010, Frontiers in Artificial Intelligence and Applications*(Vol. 219, pp. 161–168), IOS Press.

Skadiņa, I., Aker, A., Mastropavlos, N., Su, F., Tufiş, D., Verlič, M., et al. (2012). Collecting and using comparable corpora for statistical machine translation. *Proceedings of LREC'12* (pp. 438–445), Istanbul, Turkey, 21–27 May 2012.

Skadiņš, R., Goba, K., & Šics, V. (2010). Improving SMT for baltic languages with factored models. *Proceedings of the Fourth International Conference Baltic HLT 2010* (pp. 125–132), October 7–8, 2010, Riga, Latvia.

Skadiņš, R., Puriņš, M., Skadiņa, I., & Vasiļjevs,A. (2011). Evaluation of SMT in localization to under-resourced inflected language. *Proceedings of the 15th International Conference of the European Association for Machine Translation EAMT 2011* (pp. 35–40), May 30–31, 2011, Leuven, Belgium.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. Proceedings of Association for Machine Translation in the Americas.

Snover, M., Madnani, N., Dorr, B., & Schwartz, R. (2009). Fluency, adequacy, or HTER? Exploring different human judgments with a tunable MT metric. *Proceedings of WMT09*.

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Ştefănescu, D., Ion, R., & Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT 2012)* (pp. 137–144), Trento, Italy.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., TufisD., et al. (2006). The jrcacquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation*.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., & Schlüter, P. (2012). DGT-TM: A freely available translation memory in 22 languages. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, Istanbul, 21–27 May 2012.

Steinberger, R., Ebrahim, M., Poulis, A., Carrasco-Benitez, M., Schlüter, P., Przybyszewski, M., et al. (2014). An overview of the European Union's highly multilingual parallel corpora. *Language Resources and Evaluation Journal (LRE), 48*(4), 679–707.

Su, F., & Babych, B. (2012). Measuring comparability of documents in non-parallel corpora for efficient extraction of (semi-) parallel translation equivalents. *Proceedings of the EACL'12 Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRBMT) and Hybrid Approaches to Machine Translation (HyTra)* (pp. 10–19), Avignon, France, 23–27 April 2012.

Thurmair, Gr., & Aleksić, V. (2012). Creating term and lexicon entries from phrase tables. *Proceedings of the EAMT 2012*,Trento, Italy.

Tiedemann, J. (2009). News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, & R. Mitkov (Eds.), *Recent Advances in Natural Language Processing* (Vol. V, pp. 237–248). Amsterdam/ Philadelphia: John Benjamins.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*.

Tyers, F., & Alperen, M. (2010). South-East European Times: A parallel corpus of Balkan languages. *Proceedings of Workshop "Exploitation of Multilingual Resources and Tools for Central and (South) Eastern European Languages"*.

Vasiļjevs, A., Skadiņš, R., & Tiedemann, J. (2012). LetsMT!: A cloud-based platform for do-it-yourself machine translation. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL2012)* (pp. 43–48), Jeju, Republic of Korea, 10 July 2012, System Demonstrations.

Watson, C., Li, F. W. B., & Lau, R. W. H. (2010). A pedagogical interface for authoring adaptive e-learning courses. *Proceedings of the Second ACM International Workshop on Multimedia Technologies for Distance Learning (MTDL '10)* (pp. 13–18). ACM, New York, NY.

White, J., O'Connell, T., & O'Mara, F. (1994). The ARPA MT evaluation methodologies: Evolution, lessons, and future approaches. *Proceedings of the 1st Conference of the Association for Machine Translation in the Americas* (pp. 193–205). Columbia.

Xu, J., Zens, R., & Ney, H. (2006) Partitioning parallel documents using binary segmentation. *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation* (pp. 78–85), New York City, NY, June 2006.

Xu, J., Deng, Y., Gao, Y., & Ney, H. (2007) Domain dependent machine translation. *Proceedings of the Machine Translation Summit XI*, Copenhagen, Danmark, September 2007.

Zhang, X. (2011). Two-level parallel text extraction from comparable corpora. Diploma thesis of Univeristy of Saarland.

Chapter 7
New Areas of Application of Comparable Corpora

Reinhard Rapp, Vivian Xu, Michael Zock, Serge Sharoff, Richard Forsyth, Bogdan Babych, Chenhui Chu, Toshiaki Nakazawa, and Sadao Kurohashi

Abstract This chapter describes several approaches of using comparable corpora beyond the area of MT for under-resourced languages, which is the primary focus of the ACCURAT project. Section 7.1, which is based on Rapp and Zock (Automatic dictionary expansion using non-parallel corpora. In: A. Fink, B. Lausen, W. Seidel, & A. Ultsch (Eds.) Advances in Data Analysis, Data Handling and Business Intelligence. Proceedings of the 32nd Annual Meeting of the GfKl, 2008. Springer, Heidelberg, 2010), addresses the task of creating resources for bilingual dictionaries using a seed lexicon; Sect. 7.2 (based on Rapp et al., Identifying word translations from comparable documents without a seed lexicon. *Proceedings of LREC 2012*, Istanbul, 2012) develops and evaluates a novel methodology of creating bilingual dictionaries without an initial lexicon. Section 7.3 proposes a novel system that can extract Chinese–Japanese parallel sentences from quasi-comparable and comparable corpora.

Chapter editors: Bogdan Babych and Inguna Skadiņa

R. Rapp
University of Mainz, Mainz, Germany

V. Xu
Beijing Foreign Studies University, Beijing, China

M. Zock
CNRS, Marseille, France

S. Sharoff · R. Forsyth · B. Babych (✉)
University of Leeds, Leeds, UK
e-mail: b.babych@leeds.ac.uk

C. Chu · T. Nakazawa · S. Kurohashi
Graduate School of Informatics, Kyoto University, Kyoto, Japan

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_7

7.1 Automatic Dictionary Expansion Using a Seed Lexicon and Non-parallel Corpora

Automatically generating bilingual dictionaries from parallel, manually translated texts is a well-established technique that works well in practice. However, parallel texts are a scarce resource. Therefore, it is desirable to also be able to generate dictionaries from pairs of comparable monolingual corpora. For most languages, such corpora are much easier to acquire, and it is often easier to do so in considerably larger quantities. In this section, we present the implementation of an algorithm which successfully exploits such corpora. Based on the assumption that the co-occurrence patterns between different languages are related, it expands a small base lexicon. For improved performance, it also realises a novel interlingua approach. That is if corpora of more than two languages are available, the translations from one language to another can be determined not only directly but also indirectly via a pivot language.

7.1.1 Motivation: Improving Algorithms and Boosting Performance via Cross-Language Transitivity

Until some years ago, English has been the primary language of the Internet. However, with web contents expanding from the initial and mainly technical topics to topics of almost any aspect of life, there is a tendency for web publishers to adopt the native tongue of the intended audience. This leads to a significant increase of web pages written in languages other than English.

This reduces the share of web pages written in English, and, from time to time, it becomes harder for native English speakers to find relevant information in their native language—this situation is familiar to speakers of all other languages. As a consequence of the web becoming increasingly multilingual, as well as of globalisation in general, the need for affordable dictionaries is growing. To be able to optimally exploit the information on the web, dictionaries between all language pairs would be desirable. However, with 6800 living languages, of which 600 exist in written form, this is not very realistic. But even if we only consider the 100 main languages, which cover 95% of the world's population, there are still 4450 possible language pairs (9900 directions) requiring dictionaries.

The need for dictionaries between a large number of language pairs makes self-learning systems an interesting option. Such systems are able to automatically extract raw versions of dictionaries from translated texts. However, the required parallel texts are a scarce resource.[1] Despite all efforts to mine parallel texts from

[1]Examples are the parallel corpora derived from the proceedings of the European parliament (Armstrong et al. 1998; Koehn 2005) and the JRC-Aquis corpus (Steinberger et al. 2006).

pairs of monolingual corpora (Munteanu and Marcu 2005; Wu and Fung 2005; Adafre and de Rijke 2006; Zhao and Vogel 2002), the required quantities of such data are not available for most language pairs (Rapp and Martin Vide 2007).[2]

This is why we propose a methodology for dictionary extraction that directly operates on monolingual corpora. As monolingual corpora are far easier to acquire than their bilingual counterparts, this should considerably diminish the data acquisition bottleneck. This is all the more true as one corpus per language is usually sufficient for monolingual corpora, whereas one corpus per language pair is required for parallel corpora. Consequently, instead of a linear increase, there is a quadratic increase with the number of languages.

The basic assumption underlying our approach is that, across languages, there is a correlation between the co-occurrence patterns of words that are mutual translations. For example if two words in language A co-occur more often than expected by chance, then their translated equivalents in language B should also co-occur more frequently than expected. In a feasibility study (Rapp 1995), we showed that this assumption holds for English and German even in the case of unrelated texts. When comparing an English and a German co-occurrence matrix of corresponding words, we found a high correlation between the co-occurrence patterns of the two matrices when the rows and columns of both matrices were in corresponding word order, whereas the correlation was low when the rows and columns were in random order.

The validity of this co-occurrence constraint is obvious for parallel corpora, but, as described above, it also holds for non-parallel corpora. It can be expected that this constraint will work best with parallel corpora, second best with comparable corpora and somewhat worse with unrelated corpora. Robustness is not a big issue in any of these cases. In contrast, when applying sentence alignment algorithms to parallel corpora, omissions, insertions and transpositions of text segments have critical negative effects. However, the co-occurrence constraint when applied to comparable corpora is much weaker than the word-order constraint as used with parallel corpora. This is why larger corpora and well-chosen statistical methods are needed.

The current work can be seen as a continuation of our previous work (Rapp 1995, 1999). We present a novel algorithm and provide quantitative results for six language pairs rather than for just one. Related work has been conducted by Fung and Yee (1998), Fung and McKeown (1997) and Chiao et al. (2004). By pre-supposing a lexicon of seed words, Fung and McKeown avoid the prohibitively expensive computational effort encountered by Rapp (1995). The method described here goes in the same direction. By assuming the existence of an initial lexicon, we significantly reduce the search space. We only conduct a relatively small number of vector comparisons instead of considering a very large number of permutations concerning potential correspondences of word order.

[2]For an overview on the availability of parallel texts for various languages, see Mike Maxwell's posting on the corpora mailing list of February 27, 2008, with subject line 'quantities of publicly available parallel text', archived at http://listserv.linguistlist.org/archives/corpora.html

Another new feature of this work is that it explores the possibility of utilising the transitivity property of dictionaries. What we mean by this is the following: If we have two dictionaries, one translating from language A to language B and the other from language B to language C, then we can also translate from A to C by using B as the pivot, interlingua or intermediate language. The property of transitivity, although having some limitations due to ambiguity problems, can be exploited for the automatic generation of a raw dictionary with mappings from A to C. One might consider this unnecessary as our corpus-based approach already allows us to generate such a dictionary with even higher accuracy directly from the respective comparable corpora.

However, this implies that we now have two ways to generate a dictionary for a particular language pair, which means that, in principle, we can validate one with the other. Furthermore, given several languages, there is not only one method to generate a transitivity-based dictionary for A to C, but there are several. This means that by increasing the number of languages, we also increase the possibilities of mutual cross-validation. This section presents an evaluation of the results that can be expected when constructing a dictionary using a single interlingua and compares them to the results obtained without the use of an interlingua. Our evaluation gives exact quality measures for six language directions.

7.1.2 Approach

As mentioned above, we assume that there is a strong correlation between the co-occurrences of words that are mutual translations across languages. It is further assumed that there is a small dictionary available at the beginning, and that our aim is to expand this base lexicon. Using a corpus of the target language, we first compute a co-occurrence matrix whose rows are all word types occurring in the corpus and whose columns are all target words appearing in the base lexicon. We then apply an association measure on this co-occurrence matrix, namely, the log-likelihood ratio (Dunning 1993). Next, we select a word of the source language whose translation is to be determined. Using our source-language corpus, we compute a co-occurrence vector for this word, and we also apply the association measure to it. After this, we translate all known words in this vector to their corresponding form in the target language. This is done via the base lexicon. In the case of ambiguous words, we use the primary translation, that is the one that is listed first in the lexicon. Since our base lexicon is small, only some of the translations are known. All unknown words are discarded from the vector. The entries of the target language vector are then sorted according to their association strengths. We keep only the 30 strongest associations and eliminate all others.

As a result, we have a vector of the source language word which comprises its top 30 associations that can be translated using the base lexicon. During the next step, the ranks of these 30 translations are determined for each word in the target language vocabulary (comprising all words of the target language corpus with a frequency of

100 or higher), and the product of their ranks is computed. The word with the smallest value of the product is considered to be the translation of the source language word.

This algorithm turned out to be a significant improvement over the previous one described by Rapp (1999). It provides better accuracy and considerably higher robustness with regard to sampling errors. The reason for the improvement appears to be that outliers and function words, which may have a negative effect on results, are usually not among the top 30 associations, and hence, they do not have any impact, at least not on the side of the source language.

The exploration of transitivity (see Sect. 7.1.1) was conducted as follows: Using the improved algorithm, we start by generating a dictionary that translates the test words from the source language to the 'interlingua'—one specific language chosen as a pivot, for which most dictionary links are available. Next, we translate the resulting word list from the interlingua to the target language. Finally, the outcome is compared to our gold standard. As the interlingua(pivot) approach is based on a two-stage process with errors cumulating, the results can be expected to be worse than for direct translation. Nevertheless, we believe in the virtues of this approach as there are different ways of choosing the interlingua which can be optimised for specific tasks.

7.1.3 Language Resources

Three languages were considered, namely, English, French and German, and so were all six language pairs that can be derived from these. To conduct the simulation, a number of resources were required:

1. Monolingual corpora for each of the three languages
2. A number of word equations (English–French–German) to be used as a gold standard for evaluating the results
3. Small base dictionaries for each of the six language pairs

For German, we used a corpus of 135 million words from the Frankfurter Allgemeine Zeitung newspaper (1993–1996). For English, we relied on a corpus of 163 million words from The Guardian (1990–1994). Only a small set of newspaper corpora was available to us for French, and therefore, we acquired a corpus comprising the French version of Wikipedia and ABU—LaBibliotheque Universelle (together about 70 million words). For each corpus, a specific clean-up program was written and applied.

Since these corpora are relatively large, we decided to remove all function words from the texts to save disk space and processing time. This was done on the basis of a stop-word list of approximately 600 German words, with second list of about 200 English words and a third list of about 500 French words. By eliminating function words, we assumed that we would lose little information. Function words are often highly ambiguous, and their co-occurrences are mostly caused by syntactic,

rather than semantic, patterns. Since semantic patterns are more reliable than syntactic patterns across language families, we hoped that eliminating the function words would increase the generality of our method.

Rapp (1999) used a list of 100 German test words together with their English translations as the gold standard for testing results. As this list is rather small, and as we also needed French translations, we decided to compile a larger trilingual list of test words. For this purpose, we used three editions of Collins Gem Dictionaries which are small pocket dictionaries intended for everyday use. We started with the German-to-English part of the Collins Gem German Dictionary which contains about 20,000 entries. For each German word, we considered only the primary English translation, that is the one that was listed first. We looked up each of these in the Collins English-to-French dictionary, again only taking primary translations into account. Finally, we looked up the French words in the same way in the Collins French-to-German dictionary. In this way, we obtained a large table of word translations comprising the following columns: German–English–French–German. From this table, we eliminated all lines where the German words in the first and fourth columns differed. From the remaining table of 1079 words, we eliminated the fourth column, as it had become redundant. The resulting list of trilingual word equations was used as the test set for our evaluations.

Note that in order to arrive at this test set, we used only three of the six language pairs, and the order in which we applied the dictionaries was more or less arbitrary. We had tried other language pairs and other dictionary orders, with somewhat different outcomes. We finally decided to choose the current one, as our intention had been to arrive at a test set of about 1000 items.

The six base lexicons required by our algorithm were also derived from the Collins Gem Dictionaries. All multi-word entries were eliminated. Since it would not make sense to apply our method to words that are already in the base lexicon, we removed all dictionary entries belonging to the 1079 test words in the source language of the respective language pair.

7.1.4 Results

Based on the algorithm and the corpora described above, we computed the translation of each word of the test word list into the two other languages.

Hereby, a window size of plus or minus two words from the given word was assumed for co-occurrence counting. As function words had been removed from the corpora beforehand, and assuming that roughly every second word is a function word, this corresponds to a window size of about plus or minus four words in the original text. Since our algorithm requires relatively time-consuming computations for each word in the target vocabulary, we decided to take into account only words with a corpus frequency of at least 100 in order to save processing time. As our corpora are rather large, this threshold leaves almost all common words in the vocabulary, while eliminating most misspelled words.

Table 7.1 Top ten computed translations for six German words

Historie (history)		*Leibwächter* (body guard)	
1. 29453 13.73	History	1. 949 40.02	Bodyguard
2. 4997 12.87	Literature	2. 5619 23.34	Policeman
3. 4758 8.74	Historical	3. 2535 8.18	Gunman
4. 2670 0.67	Essay	4. 26347 3.69	Kill
5. 6969 0.11	Contemporary	5. 9180 2.92	Guard
6. 18909 −1.72	Art	6. 401 −0.56	Bystander
7. 18382 −2.81	Modern	7. 815 −1.24	Police
8. 15728 −4.31	Writing	8. 8503 −2.33	Injured
9. 1447 −5.52	Photography	9. 2973−3.23	Stab
10. 2442 −5.53	Narrative	10. 1876 −3.58	Murderer
Raumfähre (space shuttle)		*Spirituell* (spiritual)	
1. 259 46.20	Shuttle	1. 2964 56.10	Spiritual
2. 666 26.25	Nasa	2. 1380 8.34	Christianity
3. 473 25.95	Astronaut	3. 7721 8.08	Religious
4. 287 25.76	Spacecraft	4. 9525 4.10	Moral
5. 1062 16.92	Orbit	5. 1414 0.63	Secular
6. 16086 11.72	Space	6. 5685 0.06	Emotional
7. 525 9.50	Manned	7. 4678 −1.04	Religion
8. 125 7.69	Cosmonaut	8. 6447 −1.49	Intellectual
9. 254 5.24	Mir	9. 8749 −2.25	Belief
10. 7080 3.70	Plane	10. 8863 −4.07	Cultural
Ukrainisch (Ukrainian)		*Umdenken* (rethink)	
1. 1753 50.69	Ukrainian	1. 1119 20.76	Rethink
2. 22626 39.88	Russian	2. 248 15.46	Reassessment
3. 3205 29.25	Ukraine	3. 84109 13.39	Change
4. 34572 23.63	Soviet	4. 12497 12.13	Reform
5. 978 21.13	Lithuanian	5. 236 10.00	Reappraisal
6. 1005 18.88	Kiev	6. 9220 9.97	Improvement
7. 10968 15.07	Gorbachev	7. 5212 9.48	Implement
8. 10209 14.51	Yeltsin	8. 1139 8.25	Overhaul
9. 16616 13.38	Republic	9. 13550 7.89	Unless
10. 502 11.71	Latvian	10. 9807 7.88	Immediate

Table 7.1 gives an idea of the system's performance. It shows the top ten computed translations for the following six German words: *Historie* (history), *Leibwächter* (bodyguard), *Raumfähre* (space shuttle), *spirituell* (spiritual), *ukrainisch* (Ukranian) and *umdenken* (rethink). The columns have the following meanings:

1. Rank of a potential translation
2. Corpus frequency of the translation
3. Score assigned to the translation (the larger the better)
4. Computed translation

Table 7.2 Evaluation results for translation directions

German → English	47.5%
English → German	35.7%
German → French	21.2%
French → German	21.7%
French → English	30.1%
English → French	34.9%

Table 7.3 Interlingua evaluation results for translation directions

German → French → English	11.4%
English → French → German	13.5%
German → English → French	24.7%
French → English → German	16.2%
French → German → English	16.3%
English → German → French	13.4%

The results for the five other language pairs are of roughly comparable quality. If we look at the table, we see that a correct translation is usually ranked first and that typical associations follow. This behaviour can be expected from our association-based approach.

To get a better picture of the quality of the results, we also conducted a quantitative evaluation. For all 1079 test words, we checked whether the predicted translation (first word in the ranked list) was identical to our expected translation (as taken from the word equations used as our gold standard). This was true for 512 of the 1079 test words in the case of the German-to-English language pair which corresponds to an accuracy of 47.5%. Note that this is a rather conservative assessment of the quality, as our measure requires string identity and therefore has no tolerance. For example correct alternative translations (e.g. *road* instead of *street* for *Straße*) or inflected forms of the expected translation are counted as mistakes. Table 7.2 gives analogous results for all the six language pairs.

As the results vary quite a bit, the question arises of how to explain the differences. Here are some attempts: On one hand, our corpus of French is considerably smaller than our corpora of English and German (70 versus about 150 million words), and it is a different genre (encyclopaedia rather than newspaper). On the other hand, French and German are highly inflectional languages, whereas English is not. So the risk of selecting an inflectional variant of the expected translation (which would be counted as incorrect) is lower in English. Another consideration concerns the degree to which two languages are related. Whereas French is a typical Romance language and German a typical Germanic language, English lies somewhere in between. From this point of view, it can be expected that the language pairs involving English achieve the best results which is confirmed by the table.

With regard to the interlingua approach, Table 7.3 shows quantitative results for the six possible language triplets as obtained using the algorithm described in Sect. 7.1.2. Whereas the performance figures without interlingua had been between 21% and 48%, the figures here vary between 11% and 25%; that is they are at about half

of this level which is clearly better than what could be expected in the case of statistical independence. This gives rise to the hope that at some point, it may be possible to obtain significantly improved results by combining several dictionaries generated via different interlinguae.

7.1.5 Discussion

In this section, we made an attempt to solve the difficult problem of identifying word translations on the basis of more or less unrelated monolingual corpora of different languages. We applied the same algorithm to six language pairs and, using a rather conservative automatic evaluation measure that is based on 1079 test words, we achieved accuracies in a range between 21% and 48%. We showed that the algorithm can be extended towards an interlingua approach that makes it possible to construct a dictionary for a particular language pair via several interlinguae, thereby opening up the possibility of improving the results through mutual cross-validation. What we suggest for future work is to perform a complete cross-validation that ranks each dictionary entry according to the number of successful cross-validations. If applicable, the work of a human end-validator can be facilitated by providing him a ranked list of the translations of a word, ordered according to these ratings. In addition, the amount of data to be considered by the validator can be significantly reduced by introducing a threshold, that is by eliminating translations that do not reach a certain level.

To make this feasible, we need large monolingual corpora (if possible from the same genre) for as many languages as possible. Well suited for this purpose would be, for example, the Gigaword Corpora from the Linguistic Data Consortium, which are billion word newsticker texts that are available (though at substantial cost) for Arabic, Chinese, English, French and Spanish.

Other possibilities for improvement include pre-processing of the corpora and bootstrapping of the base lexicon. Pre-processing depends on the tools that are available for the respective languages. For example a lemmatiser can convert inflectional variants to their respective base forms which should significantly reduce the problem of data sparseness. Alternatively, with a program for word sense disambiguation, different senses of a word can be distinguished, and the appropriate translations can be determined for each sense of a word. Alternatively, if no disambiguator is available, it can be considered whether to look at co-occurrences between sequences of words instead of co-occurrences between single words. The rationale behind this is that neighbouring words often disambiguate each other, so the word sequences are likely to carry less ambiguity than the words.

By bootstrapping off the base lexicon we mean that the algorithm starts from a very small base lexicon which can then be expanded iteratively. To improve operation, those source language words whose associations are covered by the base lexicon should be identified systematically, so that their translations can be determined first. For such words, the likelihood of arriving at a correct translation

ought to be highest. Once their translations are known, they are added to the base lexicon, and the process is repeated. After several iterations, the existing entries of the base lexicon can be repeatedly re-computed and revised, in order to obtain improved accuracy (which gains from the increase in lexicon coverage). Assuming large corpora of good quality, it is well possible that this process converges at accuracy levels that are significantly better than what we were able to present here.

7.2 Identifying Word Translations from Comparable Documents Without a Seed Lexicon

The extraction of dictionaries from parallel text corpora is an established technique. However, as parallel corpora are a scarce resource, in recent years, the extraction of dictionaries using comparable corpora has obtained increasing attention. For finding a mapping between languages, almost all approaches suggested in the literature rely on a seed lexicon. The work described here achieves competitive results without requiring such a seed lexicon. Instead, it pre-supposes mappings between comparable documents in different languages. Such mappings are either readily available or can be established relatively easily for some common types of textual resources (e.g. encyclopaedias or newspaper texts). The current work is based on Wikipedias where the mappings between languages are determined by the authors of the articles. We describe a neural-network inspired algorithm which first characterises each Wikipedia article by a number of keywords and then considers the identification of word translations as a variant of word alignment in a noisy environment. We present results and evaluations for eight language pairs involving Germanic, Romanic and Slavic languages as well as Chinese.

7.2.1 Motivation

The current section extends our work presented in Rapp (1999). For an overview of other related works, let us refer to Laws et al. (2010). However, apart from comparatively limited methods which are based on cognates and therefore only work for closely related languages, almost all previous approaches have in common that they pre-suppose an initial dictionary (bilingual lexicon of seed words) in order to be able to relate between languages. In contrast, the approach that we present here does not require such a lexicon, but instead assumes the availability of aligned comparable documents. This is not an unreasonable requirement for some common text types: for example Wikipedia articles of different languages that are aligned via the so-called inter-language links and newspaper articles that can be aligned via their dates of publication in combination with some basic topic detection software.

7.2.2 Approach

Like most previous ones, our approach is also based on the assumption that there is a correlation between the patterns of word-co-occurrence across languages. However, instead of pre-supposing a bilingual dictionary, it only requires pre-aligned comparable documents, that is small- or medium-sized documents across languages that are known to deal with similar topics. This can be, for example newspaper articles, scientific papers, contributions to discussion groups or encyclopaedic articles. As Wikipedia is a large resource and readily available for many languages, and to be able to compare our results to recent related works that also use Wikipedia (Laws et al. 2010), we decided to base our study on this encyclopaedia. Our algorithm is (apart from word segmentation issues) largely language independent and should lead to similarly good results for any languages where Wikipedias of reasonable size are available. Some statistics of the Wikipedias are shown in Table 7.4 (Wikipedia dumps were downloaded from http://dumps.wikimedia.org/ between November 4 and 12, 2011).

The Wikipedias have the so-called inter-language links, which connect two articles in different languages. Therefore, if a headword is dealt with in several languages, a special inter-language tag (iwiki) is usually placed in the respective article. For example the English article on the headword *Depth-of-field adapter*[3] contains iwiki links such as

Czech: DOF adaptér
German: 35-Millimeter-Adapter
Korean: DOF 어댑터
Japanese: DOFアダプタ
Russian: DOF-адаптер

The right column in Table 7.4 refers to the number of iwiki links from the various languages into English. Given that typically the members of a Wikipedia community

Table 7.4 Wikipedia statistics

Language	id	Million Tokens	Articles	EN iwiki links
Chinese	zh	101	137179	87389
Dutch	nl	163	435716	290979
English	en	1440	2524134	n/a
French	fr	459	838771	541715
German	de	563	1114696	603437
Portuguese	pt	156	361204	245102
Russian	ru	268	609525	345195
Spanish	es	365	664097	438864
Ukrainian	uk	81	214403	139827

[3]This is an image converter allowing the exchange of camera lenses, thereby providing a shallow depth of field.

contribute in their own language, only occasionally will an article connected in this way be a simple translation of the English article, and, in most cases, the contents will be rather different. On the positive side, the link structure of the inter-language links tends to be quite dense, see Table 7.4. It should be mentioned that the set of headwords connected by these links can already be considered as a raw dictionary of mainly nouns and proper nouns which, in principle, could be used for evaluation purposes. However, in this work, we decided to use evaluation data from an independent source.

7.2.2.1 Pre-Processing Steps

After download, each Wikipedia was minimally processed to extract the plain text contents of the articles. In this process, all templates (e.g. 'infoboxes') and tables were removed, and we kept only the webpages with more than 500 characters of running text (including white space). We maintained the iwiki links to the English webpages as well as 'Categories', though the latter were not used in the process discussed below.

Linguistic processing steps included tokenisation, tagging and lemmatisation using the default UTF-8 versions of the respective TreeTagger resources (Schmid 1994) for all languages except Russian and Ukrainian, for which tagging and lemmatisation was done using our own tools (Sharoff et al. 2008) based on TnT (Brants 2000) and CST lemmatiser (Jongejan and Dalianis 2009). Given that the tokeniser for Chinese used in TreeTagger (which is in turn our own development) uses the simplified script, the contents of the Chinese Wikipedia were converted to the simplified characters for uniformity reasons.

7.2.2.2 Alignment Steps

As these documents are typically not translations of each other, we cannot apply the usual procedure and tools available for parallel texts (e.g. the Gale & Church sentence aligner and the Giza++ word alignment tool). Instead, we conduct a two-step procedure:

1. We first extract salient terms from each of the documents.
2. We then align these terms across languages using an approach inspired by a connectionist (Rumelhart and McClelland 1987) WINner-Takes-It-All Network (WINTIAN algorithm).

The procedure for term extraction is based on using the frequency list of the entire Wikipedia to measure the keyness of words in each individual article. The articles are usually short, with an average length of about 500 words, and so we use the log-likelihood score as a measure of keyness, since it has been shown to be robust for small numbers of instances (Rayson and Garside 2000). For example the keywords extracted for English and German for the above-mentioned article (Depth-of-field adapter) are shown in Table 7.5.

Table 7.5 English and German keywords for the Wikipedia article 'Depth-of-field adapter'

English			German		
LL-score	*N*	Term	LL-score	*N*	Term
288.73	21	Adapter	253.75	14	Mattscheibe
173.00	17	Lens	116.09	5	35-mm-Adapter
151.71	10	Camcorder	46.58	3	Körnung
137.11	17	Screen	43.61	2	35-Millimeter-Adapter
120.45	18	Focus	38.84	3	Adapter
94.43	9	Flip	37.72	2	HD-Auflösung
83.97	11	Camera	32.35	8	Bild
80.00	15	Image	31.65	3	Linse
58.59	5	Macro	29.86	3	Objektiv
50.47	4	35 mm	29.44	4	Kamera
38.83	2	Plano-convex	28.01	3	Statisch
34.87	3	Translucent	27.76	2	Schärfentiefe
33.44	2	Vignetting	25.72	2	Videokamera
31.12	5	Mount	24.30	2	Spiegelreflexkamera
25.31	3	Photographic	23.54	2	Sucher
25.18	3	Texture	17.84	2	Bewegt
22.68	2	Flange	17.26	3	Hersteller
21.84	2	Aberration	16.94	2	Hundert
21.65	2	Post-production	16.10	2	Scheibe
20.27	2	Chromatic	15.25	2	Einschränkung
20.24	3	Module			
20.01	2	Upside			
19.88	3	Mirror			
19.52	2	Zoom			
19.03	2	Prism			
18.89	3	Monitor			
18.41	2	Blur			
16.80	5	Must			
16.80	3	Correct			
15.35	2	Canon			
15.35	3	Frame			
15.33	3	Attach			

LL log-likelihood, *N* term frequency in document

According to Rayson and Garside (2000), the threshold of 15.13 for the log-likelihood score is a conservative recommendation for statistical significance. The WINTIAN algorithm is used for establishing term alignments across languages. As a detailed technical description is given in Rapp (1996: 108), we only briefly describe this algorithm here, thereby focussing on the neural network analogy. The algorithm can be considered as an artificial neural network where the nodes are all English and German words occurring in the keyword lists. Each English word has connections to all German words whose weights are all one at the beginning, but will

be a measure of the translation probabilities after the completion of the algorithm. One after the other, the network is fed with the pairs of corresponding keyword lists. Each German word activates the corresponding German node with an activity of 1. This activity is then propagated to all English words occurring in the corresponding list of keywords. The distribution of the activity is not equal, but in proportion to the connecting weights. This unequal distribution has no effect at the beginning when all weights are one, but, later on, it leads to increase in rapid activity for pairs of words that often occur in corresponding keyword lists. Of course, it is assumed that these are translations of each other. The activity changes are stored in the connections using Hebbian learning (Rumelhart and McClelland 1987). We use a heuristic to avoid the effect that frequent keywords dominate the network. When more than 50 of the connections to a particular English node have weights higher than 1, the weakest 20 of them are re-set to 1. In this way, only translations that are frequently confirmed can build up high weights.

Let us look at an example. Assume we have the (very short) English keyword list '*bank money*' corresponding to the German list '*Bank Geld*' and another English list '*bank river*' corresponding to the German '*Bank Fluss*'. When the network receives the first pair of keywords in the first cycle, it cannot decide whether '*bank*' corresponds to '*Bank*' or to '*Geld*', and so, it will assign each possibility an activity of 0.5. So both weights will be increased equally. But, when it comes to distributing the activity of '*bank*' in the second cycle, the weight to '*Bank*' will be stronger than the one to '*Fluss*'. Therefore, '*Bank*' will receive more activity, and the respective weight will become even stronger, in effect, correctly disambiguating the ambiguous English word.

It turned out that the algorithm shows a robust behaviour in practice which is important as the corresponding keyword lists are very noisy and may well contain less than 20% words that are actually translations of each other. Reasons are that corresponding articles are often written from different perspectives and can considerably vary in length. (To give an example, the descriptions of politicians tend to be very country specific). Nevertheless, the algorithm is capable of grasping the regularities and often comes up with reasonable results.

7.2.2.3 Vocabularies

The WINTIAN algorithm needs as input vocabularies of the source and the target language. For each language, we constructed these as follows: Based on the keyword lists for the respective Wikipedia, we counted the number of occurrences of each keyword, and then applied a threshold of five: that is all keywords with a lower frequency were eliminated. The reasoning behind this is that rare keywords are of not much use due to data sparseness.[4] To this vocabulary, we added all words of the

[4]In corpus based studies, sometimes thresholds of e.g. 50 are recommended. However, as we here consider keywords that have a higher information content than an average token in a corpus, it makes sense to use a lower threshold.

Table 7.6 Corpus and vocabulary sizes

Language	id	Million tokens	Vocabulary size
Chinese	zh	101	36623
Dutch	nl	163	58563
English	en	1440	133806
French	fr	459	101399
German	de	563	144251
Portuguese	pt	156	50003
Russian	ru	268	80940
Spanish	es	365	89732
Ukrainian	uk	81	30888

applicable gold standard(s) relating to the respective language (i.e. including the Google translations, and, if applicable for a language, their manual corrections, and the TS100 test set). Note that adding the words from the gold standard(s) means only a modest increase in vocabulary size as most of them easily meet the frequency threshold. Applying this procedure led to the vocabulary sizes as shown in Table 7.6.

The vocabularies for larger Wikipedias are more comprehensive, because more keywords meet the minimum frequency. As the gold standard words are included in any case, the selection task for the WINTIAN algorithm is somewhat easier for languages with a smaller Wikipedia, since the choice of words is more limited. Although this is hardly noticeable at the above vocabulary sizes, it would be an important factor for very small vocabularies. As a consequence, not only corpus size but also vocabulary size is important when comparing different algorithms, a fact that is sometimes overlooked.

7.2.3 Evaluation Set-Up

Our aim was to have a gold standard of word equations to test the predicted translation equivalents as computed by the WINTIAN algorithm. The source language words in the gold standard were supposed to be systematically derived from a large corpus, covering a wide range of frequencies, parts of speech and variances of their distribution. In addition, the corpus from which the gold standard was derived was supposed to be completely separate from the development set (Wikipedia). The limitation of this method is, however, that translations were generated by Google Translate, and then manually checked, and only one of several possible translations of the English words is included in the gold standard.

For a quantitative evaluation, we used two datasets consisting of word equations. The first gold standard is the TS100 test set as described in Laws et al. (2010) and previously used by Rapp (1999) for the German–English pair. It comprises 100 English words together with their German translations.

As the TS100 test set is rather small, we developed a larger test set comprising 1000 items. We began with a list of words which was extracted by Adam Kilgarriff

from the British National Corpus (BNC) for the purpose of examining distributional variability. This list is described at http://kilgarriff.co.uk/bnc-readme.html. It contains 8187 words; which occur at least 100 times in a 10.1-million word subset of the BNC, comprising those documents that are at least 5000 words in length. Kilgarriff's main idea was to look at variation in frequencies across 2018 segments of 5000 words each. Thus, the items give us data about frequency and variability for future experiments, though, at present, we have not used this information.

Since these items are words, not lemmas, the next step was to pick uninflected forms by using the CLAWS (Constituent Likelihood Automatic Word-tagging System) tags attached. Taking the tagtypes shown in Table 7.7 and keeping only the highest in the list (most frequent) for each multi-tagged word, we get a total of 3857 entries. (We excluded items that don't begin with a letter and multi-word units with an underscore or hyphen as delimiter.) We selected 1001 from these at random. (One item, 'q.v.', was dropped as unsuitable, leaving a round thousand.) The number of items in each POS tag category is shown in Table 7.7.

The resulting list of 1000 English words was translated into the eight other languages (see Table 7.6) using Google Translate. For three of the languages, namely, German, Russian and Ukrainian, these translations were corrected by native speakers. The number of items that needed correction turned out to be approximately 100 per language. The translations for all other languages remained uncorrected.

7.2.4 Results and Evaluation

Using the WINTIAN algorithm, the English translations for all 144,251 words occurring in the German vocabulary have been computed. Table 7.8 shows sample results for three German words.

Table 7.7 Occurrences of post-tag categories

Part of speech	Number
aj0 Adjective	237
av0 Adverb	93
crd Cardinal number	12
nn0 Collective (or mass) noun	15
nn1 Singular noun	546
ord Ordinal number	3
prp Preposition	10
vbi Verb 'be' infinitive	1
vdi Verb 'do' infinitive	0
vhi Verb 'have' infinitive	1
vvb Verb base-form	7
vvi Verb infinitive	75

Table 7.8 Sample results

	LL-score	Translation
Given German Word	**Strasse**	
Expected Translation	**street**	
1	215.3	road
2	148.2	street
3	66.0	traffic
4	46.0	Road
5	42.6	route
6	34.6	building
Given German Word	**Krankheit**	
Expected Translation	**sickness**	
1	236.4	disease
2	105.3	symptom
3	61.6	illness
4	50.8	epidemic
5	44.0	treatment
6	39.1	genetic
Given German Word	**gelb**	
Expected Translation	**yellow**	
1	200.7	yellow
2	89.5	Yellow
3	17.9	green
4	13.8	tree
5	13.4	bright
6	13.1	pigment

Repeated occurrences of the same translation equivalent represent alternative capitalization variants found in corpus

7.2.4.1 Comparison with Other Works

For a quantitative evaluation, we verified in how many cases our algorithm had assigned the top rank to the expected translation (as provided by the gold standard) among all 133,806 translation candidates. (Candidates are all words occurring in the English vocabulary, see Sect. 7.2.2.3.) Table 7.9 compares our results to those of Laws et al. (2010) which represented the state of the art, and to the Rapp (1999) baseline.[5] (All results are based on the English and German Wikipedia corpora.)

As can be seen, the new approach outperforms the previous ones. However, it should be noted that the Wikipedia contents have changed over time and that a comparison based on only 100 test words can only give a rough indication.[6]

[5]Note that the scores reported in Rapp (1999) were based on different corpora and a proprietary seed lexicon, which is why this work had been replicated by Laws et al. (2010) using Wikipedia and a freely available lexicon.

[6]We could not easily compare with the TS1000 test set provided by Laws et al. (2010) as this adds some more sophistication (parts of speech and multiple translations) to the evaluation process, whereas we wanted to keep the evaluation process simple as we are dealing with many languages.

Table 7.9 Comparison of systems

System	Accuracy (%)
Baseline (Rapp 1999)	50
State of the art (Laws et al. 2010)	52
Current approach	61

A problem with our approach is that some words of the source language (typically ones with unspecific meanings) never make it as keywords, so no translations can be computed for them. In the case of the TS100 test set, this was the case for 7 out of 100 source language (i.e. German) words. This means that the WINTIAN algorithm only had a chance to come up with the correct result in 93 cases. (But the above accuracy of 61% of course relates to all 100 test items.)

To reduce this problem, we experimented with setting the log-likelihood threshold for keywords lower which, however, reduced the specificity of the keywords and consequently led to a lower overall accuracy (e.g. approximately 40% for a threshold of zero).[7]

Let us mention that the results in Table 7.9 refer to exact matches with the word equations in the gold standard. In reality, due to word ambiguity, other translations might also be acceptable (e.g. for '*Straße*' not only '*street*' but also '*road*' would be acceptable, see Table 7.8), so these figures are conservative and can be seen as a lower bound of the actual performance.

Another reason why the figures are conservative is translation asymmetry: To be comparable between languages our gold standard started with a list of English words which were translated into the other language. However, here we are considering the translation directions from the foreign languages into English (reverse direction to be covered in future work). In turn, if the most common translation of source language word A is target language word B, then, due to asymmetry, the most common back-translation of B is not necessarily A. This means our gold standard is sub-optimal when used in the direction from the foreign language to the source language.

Concerning our results, it may also be of interest in how many cases the expected translation was not ranked first, but ended up on other positions of the computed lists (as exemplified in Table 7.8). For the TS100 test set, rank 2 was obtained in nine cases and rank 3 was obtained in one case. Ranks 4–10 were not obtained in any case.

7.2.4.2 Application to Other Languages

In comparison to Laws et al. (2010), our approach is knowledge-poor, which means that, apart from word segmentation and lemmatisation (which improves results but is not essential), it does not require any linguistic processing. It also does not require a lexicon of seed words (typically comprising at least 10,000 words). For these reasons

[7] Variable thresholds depending on word frequency might reduce the problem, but this has not been implemented.

Table 7.10 Results for three language pairs where the gold standard had been verified by native speakers

	DE→ EN	RU→ EN	UK→ EN
KW	925	873	817
1	381	331	229
2	43	42	25
3	12	11	11
4	5	8	9
5	8	5	2
6	2	3	5
7	4	1	5
8	1	2	1
9	0	1	1
10	0	0	1

and because Wikipedia provides document alignments for many languages, it was straightforward to apply our algorithm to a number of other languages. However, for accurate measurements, a gold standard larger than the TS100 test set was desirable, and this had to be extended to the new languages, as described in Sect. 7.2.3. Applying our algorithm to the language pairs German → English, Russian → English and Ukrainian → English and comparing the outcome with the manually corrected versions of the gold standard led to the results as shown in Table 7.10.

In the second row here, 'KW' means the number of source language words in the gold standard (i.e. out of 1000) that actually occurred in the keyword list of the corresponding source language Wikipedia (see Sect. 7.2.4.1): that is where the WINTIAN algorithm had a chance to compute English translations. The numbers in column 1 are ranks, and the figures in the other columns indicate the number of expected translations that ended up on the respective rank. For example for the language pair German to English, 381 of the altogether 1000 expected translations (as taken from the gold standard) ended up on rank 1, 43 on rank 2 and so on. The accuracy for German is 38.1% as 381 of 1000 items were predicted correctly. This is considerably lower than our result for the TS100 test set, where we had an accuracy of 61%.

Note, however, that this drop in accuracy for the larger test set is in line with expectations. The TS100 test set contains almost only common words that have a high corpus frequency and are thus easy to predict. In contrast, by its construction, the 1000 item test set (random selection from Adam Kilgarriff's large word list) represents a much wider frequency spectrum. Laws et al. (2010) made a similar observation (i.e. drop in accuracy) with their larger test set, although theirs consisted of only the top 1000 most frequent Wikipedia words and should have therefore been easier to deal with than ours.

If we now compare the results for the three language pairs, as expected, we can observe an improvement in accuracy with an increase in the size of the respective version of Wikipedia (see Table 7.4). On the other hand, there are numerous other influences, including the relatedness of the source, the target language and the attitude of the respective Wikipedia community, where the spectrum ranges from

Table 7.11 Results for three language pairs where uncorrected Google translations are used as gold standard

	DE → EN	RU → EN	UK → EN
KW	948	861	777
1	316	319	220
2	38	44	24
3	13	15	12
4	5	10	7
5	9	5	2
6	2	3	2
7	3	1	4
8	1	1	1
9	1	1	1
10	0	1	1

simply translating English articles to the completely independent authoring of articles.

In the test sets for German, Russian and Ukrainian, the Google translations of the 1000 English words had been manually corrected by native speakers of the respective language. As this manual work is a hindrance when exploring new languages, the question arises whether an evaluation using the uncorrected Google translations might also be of some use. In general, according to the native speakers of these languages, roughly 10% of the Google translations had been erroneous, so we might also expect a drop of accuracy in this order of magnitude. Table 7.11 shows the respective results. The expected drop is noticeable in all three cases, though its degree varies. Nevertheless, the uncorrected Google translations seem suitable to give at least a rough idea of performance.

Based on this observation, we conducted an evaluation using a gold standard of uncorrected Google translations for the remaining languages.

Table 7.12 shows the results. As can be seen in conjunction with Table 7.11, the Romanic languages obtain considerably better results than the Germanic or Slavic ones, and—not too surprisingly, due to its high degree of word ambiguity—Chinese is the most difficult language to deal with.[8]

[8]For better results, an evaluation method taking into account multiple translation possibilities might be desirable for Chinese. On the other hand (similar to BLEU scores in machine translation), it is better not to take these accuracy figures as absolute but instead as a means for comparing the performances of different algorithms. We think that, for this application, it is preferable to consider only the most salient translations, because the degree of arbitrariness (as inherent in the production of any gold standard) is minimised in this way.

Table 7.12 Results for further language pairs where uncorrected Google translations are used as gold standard

	ES → EN	FR → EN	NL → EN	PT → EN	ZH → EN
KW	805	962	829	880	942
1	473	428	348	428	130
2	45	43	39	36	13
3	14	17	17	10	4
4	5	10	4	5	6
5	2	6	5	4	4
6	0	4	6	1	0
7	1	4	2	1	2
8	1	2	1	1	1
9	0	0	5	2	1
10	1	0	1	1	0

7.2.5 Discussion

We have presented a method for identifying word translations using comparable documents. Although it does not require a seed lexicon, it delivers competitive results. As has been shown, its knowledge-poor approach can be easily applied to other language pairs with reasonable results. Other than word segmentation and lemmatisation, no adaptation was required for the new language pairs, and no optimisation was conducted. The quantitative evaluations are based on a gold standard which had been developed independently before the simulations were conducted.

A disadvantage of our method is that it pre-supposes that the alignments of the comparable documents are known. On the other hand, there are methods for finding such alignments automatically, not only in special cases such as Wikipedia and newspaper texts, but also in the case of unstructured texts (although these methods may require a seed lexicon).

Our future work will concentrate on this and on refining the method and extending it to multi-word units and further languages.

7.3 Chinese–Japanese Parallel Sentence Extraction from Quasi-Comparable and Comparable Corpora

7.3.1 Motivation

While most studies are interested in language pairs between English and other languages, this section focuses on Chinese–Japanese, where parallel corpora are very scarce. We present a system that can extract Chinese–Japanese parallel sentences from both quasi-comparable and comparable corpora. It is an extension of our previous study (Chu et al. 2013) which proposes a system for extracting

Chinese–Japanese parallel sentences from quasi-comparable corpora. However, the effectiveness of the system on comparable corpora is not clear in our previous study and is further studied in this section.

We adopt a system proposed by Munteanu and Marcu (2005) which is for parallel sentence extraction from comparable corpora. We extend the system in several aspects to make it suitable for even quasi-comparable corpora. The core component of the system is a classifier that can separate parallel sentences from non-parallel sentences. The previous method of classifier training by the Cartesian product is not practical, because it differs from the real process of parallel sentence extraction. We propose a novel method of classifier training and testing that simulates the real sentence extraction process which guarantees the quality of the extracted sentences. Because Chinese characters are used both in Chinese and Japanese, they can be powerful linguistic clues to identify parallel sentences. Therefore, we use Chinese character features which significantly improve the accuracy of the classifier. We conduct parallel sentence extraction experiments on both quasi-comparable and comparable corpora and evaluate the quality of the extracted sentences either from the perspective of MT performance or manually. Experimental results show that our proposed system performs significantly better than the previous study.

7.3.2 Parallel Sentence Extraction System

The overview of our parallel sentence extraction system is presented in Fig. 7.1. Source sentences are translated into the target language using an SMT system ((1) in Fig. 7.1). We retrieve the top N documents from target language corpora with an information retrieval (IR) framework, using the translated sentences as queries ((2) in Fig. 7.1). For each source sentence, we treat all target sentences in the retrieved documents as candidates. Then, we pass the candidate sentence pairs through a sentence ratio filter and a word-overlap-based filter based on a probabilistic dictionary to reduce the candidates and keep the more reliable sentences ((3) in Fig. 7.1). Finally, a classifier trained on a small number of parallel sentences is used to identify the parallel sentences from the candidates ((4) in Fig. 7.1). A seed parallel corpus is needed to train the SMT system, generate the probabilistic dictionary and train the classifier.

Our system is inspired by Munteanu and Marcu (2005); however, there are several differences. The first difference is query generation. Munteanu and Marcu (2005) generate queries by taking the top N translations of each source word according to the probabilistic dictionary. This method is imprecise due to the noise in the dictionary. Instead, we adopt a method proposed by Abdul-Rauf and Schwenk (2011). We translate the source sentences into the target language with an SMT system trained on the seed parallel corpus. We then use the translated sentences as queries. This method can generate more precise queries, because phrase-based MT is better than word-based translation.

Another difference is that we do not conduct document matching. The reason is that documents on the same topic may not exist in quasi-comparable corpora.

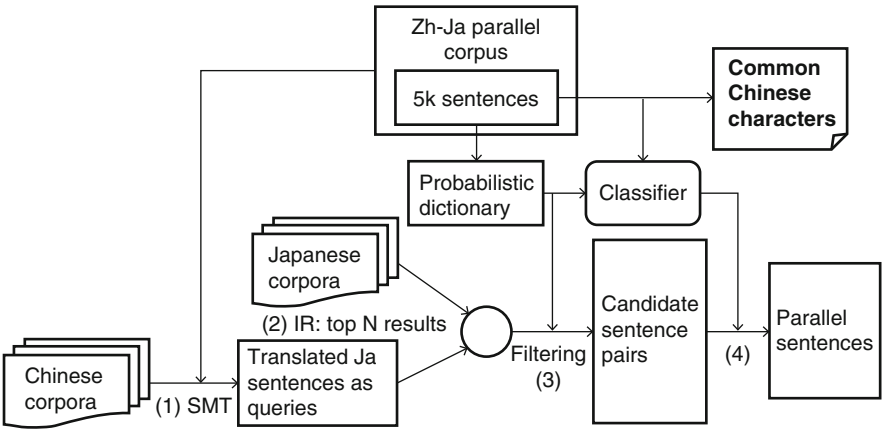


Fig. 7.1 Parallel sentence extraction system

Instead, we retrieve the top N documents for each source sentence. In comparable corpora, it is reasonable to only use the best target sentence in the retrieved documents as candidates (Abdul-Rauf and Schwenk 2011). In quasi-comparable corpora, it is important to further guarantee the recall. Therefore, we keep all target sentences in the retrieved documents as candidates.

Our system also differs in the way of classifier training and testing, which is described in Sect. 7.3.3 in detail.

7.3.3 Binary Classification of Parallel Sentence Identification

Parallel sentence identification from non-parallel sentences can be seen as a binary classification problem (Munteanu and Marcu 2005; Tillmann 2009; Smith et al. 2010; Stefanescu et al. 2012). Because the quality of the extracted sentences is determined by the accuracy of the classifier, it is the core component of the extraction system. In this section, we first describe the training and testing process, and then we introduce the features that we use for the classifier.

7.3.3.1 Training and Testing

Munteanu and Marcu (2005) propose a method of creating training and test instances for the classifier. They use a small number of parallel sentences as positive instances and generate non-parallel sentences from the parallel sentences as negative instances. They generate all the sentence pairs except for the original parallel sentence pairs in the Cartesian product and discard the pairs that do not fulfil the condition of a sentence ratio filter and a word-overlap-based filter. Furthermore, they randomly discard some of the

non-parallel sentences when necessary to guarantee the ratio of negative to positive instances smaller than 5 for the performance of the classifier.

Creating instances by using the Cartesian product is not practical, because it differs from the real process of parallel sentence extraction. Here, we propose a novel method of classifier training and testing that simulates the real parallel sentence extraction process. For training, we first select 5k parallel sentences from a seed parallel corpus. Then, we translate the source side of the selected sentences into the target language with an SMT system trained on the seed parallel corpus excluding the selected parallel sentences. We retrieve the top *N* documents from the target language side of the seed parallel corpus, using the translated sentences as queries. For each source sentence, we consider all target sentences in the retrieved documents as candidates. Finally, we pass the candidate sentence pairs through a sentence ratio filter and a word-overlap-based filter and get the training instances. We treat the sentence pairs that exist in the original 5k parallel sentences as positive instances and the remainder as negative instances. Note that positive instances may be less than 5k, because some of the parallel sentences do not pass the IR framework and the filters. For the negative instances, we also randomly discard some of them when necessary to guarantee the ratio of negative to positive instances smaller than 5. Test instances are generated by another 5k parallel sentences from the seed parallel corpus using the same method.

There are several merits of the proposed method. It can guarantee the quality of the extracted sentences because of the similarity with the real sentence extraction process. Also, features from the IR results can be used to further improve the accuracy of the classifier. The proposed method can be evaluated not only on the test sentences that passed the IR framework and the filters but also on all of the test sentences which is similar to the evaluation for the real extraction process. However, there is a limitation of our method in that both a sentence-level and document-level aligned seed parallel corpus is needed.

7.3.3.2 Features

Basic Features

The following features are the basic features that we use for the classifier which were proposed by Munteanu and Marcu (2005):

- Sentence length, length difference and length ratio
- Word overlap: the percentage of words on each side that have a translation on the other side (according to the dictionary)
- Alignment features:
  - Percentage and number of words that have no connection
  - Top three largest fertilities
  - Length of the longest contiguous connected span
  - Length of the longest unconnected sub-string.

Alignment features are extracted from the alignment results of the parallel and non-parallel sentences that are used as instances for the classifier. Note that alignment features may be unreliable when the quantity of non-parallel sentences is significantly larger than the parallel sentences.

Chinese Character Features

In contrast to some other language pairs, Chinese and Japanese share Chinese characters. In Chinese, the Chinese characters are called hanzi, while they are called kanji in Japanese. Hanzi can be divided into two groups—Simplified Chinese (used in mainland China and Singapore) and Traditional Chinese (used in Taiwan, Hong Kong and Macau). The number of strokes needed to write characters has been largely reduced in Simplified Chinese, and the shapes may be different from those in Traditional Chinese. Because kanji characters originated from ancient China, many common Chinese characters exist between hanzi and kanji. Table 7.13 gives some examples of common Chinese characters in Traditional Chinese, Simplified Chinese and Japanese, along with their Unicode.

Because Chinese characters contain significant semantic information and common Chinese characters share the same meaning, they can be valuable linguistic clues for many Chinese–Japanese natural language processing (NLP) tasks. Many studies have exploited common Chinese characters. Tan and Nagao (1995) used the occurrence of identical common Chinese characters in Chinese and Japanese (e.g. 'snow' in Table 7.13) in the automatic sentence alignment task for document-level aligned text. Goh et al. (2005) detected common Chinese characters where kanji are identical to Traditional Chinese but different from Simplified Chinese (e.g. 'love' in Table 7.13). Using a Chinese encoding converter[9] that can convert Traditional Chinese into Simplified Chinese, they built a Japanese-Simplified Chinese dictionary, partly using the direct conversion of Japanese into Chinese for Japanese kanji words. Chu et al. (2011) made use of the Unihan database[10] to detect common Chinese characters that are visual variants of each other (e.g. 'begin' in Table 7.13) and proved the effectiveness of common Chinese characters in Chinese–Japanese phrase alignment. Chu et al. (2012a) exploited common Chinese characters

Table 7.13 Examples of common Chinese characters (TC denotes Traditional Chinese and SC denotes Simplified Chinese)

Meaning	Snow	Love	Begin
TC	雪(U+96EA)	愛(U+611B)	發(U+767C)
SC	雪(U+96EA)	爱(U+7231)	发(U+53D1)
Kanji	雪(U+96EA)	愛(U+611B)	発(U+767A)

[9] http://www.mandarintools.com/zhcode.html
[10] http://unicode.org/charts/unihan.html

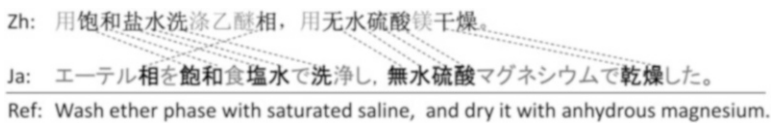


Fig. 7.2 Example of common Chinese characters in a Chinese–Japanese parallel sentence pair

in Chinese word segmentation optimisation which improved translation performance.

In this study, we exploit common Chinese characters in parallel sentence extraction. Chu et al. (2011) investigated the coverage of common Chinese characters on a parallel corpus of scientific paper abstracts and showed that over 45% of Chinese hanzi and 75% of Japanese kanji are common Chinese characters. Therefore, common Chinese characters can be powerful linguistic clues to identify parallel sentences.

We make use of the Chinese character mapping table created by Chu et al. (2012b) to detect common Chinese characters. The features used are listed below. We use an example of a Chinese–Japanese parallel sentence, which is presented in Fig. 7.2, to explain the features in detail. In the example, common Chinese characters are in bold and linked with dotted lines.

- Number of Chinese characters on each side (Zh: 18, Ja: 14).
- Percentage of characters that are Chinese on each side (Zh: 18/20 = 90%, Ja: 14/32 = 43%).
- Ratio of Chinese characters on both sides (18/14 = 128%).
- Number of *n*-gram common Chinese characters (1-gram: 12, 2-gram: 6, 3-gram: 2, 4-gram: 1).
- Percentage of *n*-gram Chinese characters out of all *n*-gram common Chinese characters on each side (Zh: 1-gram: 12/18 = 66%, 2-gram: 6/16 = 37%, 3-gram: 2/14 = 14%, 4-gram: 1/12 = 8%; Ja: 1-gram: 12/14 = 85%, 2-gram: 6/9 = 66%, 3-gram: 2/5 = 40%, 4-gram: 1/3 = 33%).

Note that Chinese character features are only applicable to Chinese–Japanese. However, because Chinese and Japanese character information is a kind of cognate (words or languages that have the same origin), the similar idea can be applied to other language pairs by using cognates. Cognates among European languages have been shown to be effective in word alignments (Kondrak et al. 2003). We also can use cognates for parallel sentence extraction.

Rank Feature

One merit of our classifier training and testing method is that features from the IR results can be used. Here, we use the ranks of the retrieved documents returned by the IR framework as a feature.

7.3.4 Experiments

We conducted classification, extraction and translation experiments on quasi-comparable corpora and extraction experiments on comparable corpora to evaluate the effectiveness of our proposed parallel sentence extraction system.

7.3.4.1 Data

Seed Parallel Corpus

The seed parallel corpus that we used is the Chinese–Japanese section of the Asian Scientific Paper Excerpt Corpus (ASPEC).[11] This corpus is a scientific domain corpus provided by the Japan Science and Technology Agency (JST)[12] and the National Institute of Information and Communications Technology (NICT).[13] It was created by the Japanese project 'Development and Research of Chinese-Japanese Natural Language Processing Technology' and contains various domains such as chemistry, physics, biology, agriculture, etc. This corpus is aligned in both sentence-level and document-level and contains 680k sentences and 100k articles (18.2M Chinese and 21.8M Japanese tokens, respectively).

Quasi-Comparable Corpus

The quasi-comparable corpus that we used is comprised of scientific paper abstracts collected from academic websites. The Chinese side of the corpus was collected from CNKI[14] and contains 420k sentences and 90k articles. The Japanese side of the corpus was collected from the CiNii[15] web portal and contains 5M sentences and 880k articles. Most articles in the Chinese side of the corpus belong to the domain of chemistry, while the Japanese side of the corpus contains various domains such as chemistry, physics and biology. However, the domain information is un-annotated in both corpora. Note that because the articles in these two websites were written by Chinese and Japanese researchers respectively, the collected corpus is very non-parallel. In addition, article alignment has not been established for this corpus.

[11] http://lotus.kuee.kyoto-u.ac.jp/ASPEC
[12] http://www.jst.go.jp
[13] http://www.nict.go.jp
[14] http://www.cnki.net
[15] http://ci.nii.ac.jp

Comparable Corpus

The comparable corpus that we used is comprised of bilingual news articles collected from the Chinese[16] and Japanese[17] versions of the People website which is the official website of the People's Daily[18] newspaper. We collected the articles with dates ranging from 2012/11/16 to 2013/05/31. The Chinese side of the corpus contains 372k sentences and 20k articles, and the Japanese side contains 316k sentences and 26k articles. Because this corpus is collected from the same news agency and the articles tend to describe similar news topics, we can treat it as a comparable corpus. Note that article alignment has not been established for this corpus either.

7.3.4.2 Classification Experiments

We conducted experiments to evaluate the accuracy of the proposed method of classification and used different 5k parallel sentences from the seed parallel corpus as training and test data.

Settings

Probabilistic dictionary: We took the top 5 translations with translation probability larger than 0.1 created from the seed parallel corpus.

- IR tool: Indri[19] with the top 10 results.
- Segmenter: For Chinese, we used a segmenter optimised for Chinese–Japanese SMT (Chu et al. 2012a). For Japanese, we used JUMAN (Kurohashi et al. 1994).
- Alignment: GIZA++.[20]
- SMT: We used the state-of-the-art phrase-based SMT toolkit Moses (Koehn et al. 2007) with default options, except for the distortion limit (6→20).
- Classifier: LIBSVM[21] with fivefold cross-validation and radial basis function (RBF) kernel.
- Sentence ratio filter threshold: 2.
- Word-overlap-based filter threshold: 0.25.
- Classifier probability threshold: 0.5.

[16] http://people.com.cn
[17] http://j.people.com.cn
[18] http://en.wikipedia.org/wiki/People's_Daily
[19] http://www.lemurproject.org/indri
[20] http://code.google.com/p/giza-pp
[21] http://www.csie.ntu.edu.tw/~cjlin/libsvm

Table 7.14 Classification results for the filtered test sentences (before '/') and all the test sentences (after '/')

Features	Precision	Recall	F-measure
Munteanu+ 2005	88.43	85.20/79.76	86.78/83.87
+Chinese character	91.62	93.63/87.66	92.61/89.60
+Rank	**92.15**	**94.53/88.50**	**93.32/90.29**

Values for the best performing method are in bold

Evaluation

We conducted classification experiments and compared the following three experimental settings:

- Munteanu+ 2005: Only using the features proposed by Munteanu and Marcu (2005).
- +Chinese character: Add the Chinese character features.
- +Rank: Further add the rank feature.

We evaluated the performance of classification by computing precision, recall and F-measure. Results evaluated for the test sentences that passed the IR framework and the filters, as well as all of the test sentences, are shown in Table 7.14. We can see that the Chinese character features can significantly improve the accuracy. The accuracy can be further improved by the rank feature.

7.3.4.3 Extraction and Translation Experiments on Quasi-Comparable Corpora

We extracted parallel sentences from the quasi-comparable corpus and evaluated Chinese-to-Japanese MT performance by appending the extracted sentences to a baseline setting.

Settings

Baseline: Using all of the 680k parallel sentences in the seed parallel corpus as training data (containing 11k sentences of chemistry domain).

- Tuning: Using another 368 sentences of chemistry domain.
- Testing: Using another 367 sentences of chemistry domain.
- Language model: 5-gram LM trained on the Japanese side of the seed parallel corpus (680k sentences) using the SRILM toolkit.[22]

The reason we evaluate on the chemistry domain is the one we described in Sect. 7.3.4.1, which is that most documents in the Chinese corpora belong to the domain of chemistry. We keep all of the sentence pairs rather than just the top 1 result (used in the classification evaluation) identified as parallel by the classifier. The other settings are the same as the ones used in the classification experiments.

[22] http://www.speech.sri.com/projects/srilm

Results

The number of extracted sentences using different classifiers is shown in Table 7.15, where:

- Munteanu+ 2005 (Cartesian): Classifier trained using the Cartesian product and only using the features proposed by Munteanu and Marcu (2005).
- Munteanu+ 2005 (Proposed): Classifier trained using the proposed method and only using the features proposed by Munteanu and Marcu (2005).
- +Chinese character (Proposed): Add the Chinese character features.
- +Rank (Proposed): Further add the rank feature.

We can see that the extracted number is significantly decreased by the proposed method in comparison to the Cartesian product which may indicate a quality improvement of the extracted sentences. Adding more features further decreases the number.

We conducted Chinese-to-Japanese translation experiments by appending the extracted sentences to the baseline. Tuning was performed by minimum error rate training (Och 2003), and it was re-run for every experiment. We report the translation results on the test set using BLEU-4 (BiLingual Evaluation Understudy MT evaluation metric, with N-gram length up to 4) score (Papineni et al. 2002) in Table 7.16. The significance test was performed using the bootstrap re-sampling method proposed by Koehn (2004). We can see that our proposed method of classifier training performs better than the Cartesian product. Adding the Chinese character features and rank feature significantly further improves the translation performance.

Discussion

The translation results indicate that, in comparison to the previous study, our proposed method can extract sentences with better quality. However, when we

Table 7.15 Number of extracted sentences

Method	# Sentences
Munteanu+ 2005 (Cartesian)	27,077
Munteanu+ 2005 (Proposed)	5994
+Chinese character (Proposed)	3936
+Rank (Proposed)	3516

Table 7.16 BLEU scores for Chinese-to-Japanese translation experiments

System	BLEU
Baseline	38.64
Munteanu+ 2005 (Cartesian)	38.10
Munteanu+ 2005 (Proposed)	38.54
+Chinese character (Proposed)	38.87†
+Rank (Proposed)	**39.47**‡*

'†' and '‡' denotes the result is significantly better than 'Munteanu+ 2005 (Cartesian)' at $p < 0.05$ and $p < 0.01$, respectively, '*' denotes the result is significantly better than the 'Baseline' at $p < 0.01$

Value for the best performing method is in bold

Fig. 7.3 Examples of sentences extracted from the quasi-comparable corpus (parallel fragments are in bold)

investigated the extracted sentences, we found that most of the extracted sentences are not sentence-level parallel. Instead, they contain many parallel fragments. Figure 7.3 presents two examples of sentence pairs extracted by '+Rank (Proposed)', and parallel fragments are indicated in bold. We investigated the alignment results of the extracted sentences. We found that most of the parallel fragments were correctly aligned with the help of the parallel sentences in the baseline system. Therefore, translation performance was improved by appending the extracted sentences. However, it also led to many wrong alignments among the non-parallel fragments which are harmful to translation. In the future, we plan to further extract these parallel fragments which can be more effective for SMT (Munteanu and Marcu 2006).

7.3.4.4 Extraction Experiments on Comparable Corpora

In addition, we extracted the parallel sentences from the comparable corpus. Different from Sect. 7.3.4.3, we manually evaluated the accuracy of the extracted sentences in this section. The main reason for this is that there are many truly parallel sentences in comparable corpora, and, by evaluating accuracy, we can show the effectiveness of our proposed system for extracting truly parallel sentences. Another reason is that we do not have a news domain test set for conducting MT experiments.

In our experiments, we again compared our proposed sentence extraction method with Munteanu and Marcu (2005). The experimental settings are the same as the ones used in Sect. 7.3.4.3, except that we set a window of 5 days around the publication date of the Chinese document containing the query sentence as a condition when we retrieve the Japanese documents ((2) in Fig. 7.1), following Munteanu and Marcu (2005). The sentence extraction results from the comparable corpus using different methods are shown in Table 7.17. Regarding the number of the extracted sentences, the results are similar to the ones reported in Sect. 7.3.3.2 in that our proposed method extracted fewer sentences than Munteanu and Marcu (2005).

Table 7.17 Number and accuracy of the extracted sentences (The accuracy was manually evaluated for 50 sentences randomly selected from the sentences extracted using different methods and is based on the number of exact matches)

Method	# Sentences	Accuracy
Munteanu+ 2005 (Cartesian)	7357	0.42
+Chinese character (Proposed)	5426	0.64
+Rank (Proposed)	4322	**0.68**

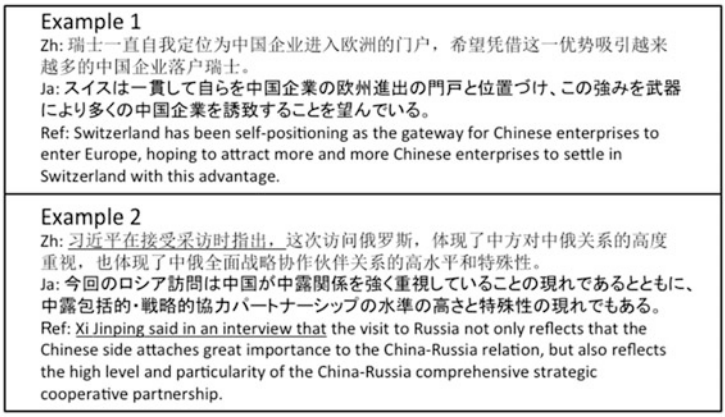


Fig. 7.4 Examples of sentences extracted from the comparable corpus (noisy parts are underlined)

To evaluate accuracy, we randomly selected 50 sentences extracted using different methods. We manually evaluated the accuracy based on the number of exact matches, namely we only treated the sentence pairs that are exact translation equivalents of each other as truly parallel sentences. We can see that the accuracy of our proposed method is significantly higher than that of Munteanu and Marcu (2005) and that adding the rank feature further improves the accuracy. This result indicates that our proposed method is effective not only for quasi-comparable corpora but also for comparable corpora.

Figure 7.4 shows two examples of the sentences extracted from the comparable corpus, where example 1 is a truly parallel sentence pair, while example 2 contains a little noise. Compared to the examples shown in Fig. 7.3 that are extracted from the quasi-comparable corpus, the sentences extracted from the comparable corpus are more parallel, which is a natural result.

7.3.5 Related Work

As parallel sentences tend to appear in similar document pairs, many studies first conduct document matching and then identify the parallel sentences from the

matched document pairs (Utiyama and Isahara 2003; Fung and Cheung 2004; Munteanu and Marcu 2005). Approaches without document matching have also been proposed (Tillmann 2009; Abdul-Rauf and Schwenk 2011; Stefanescu et al. 2012). These studies directly retrieve candidate sentence pairs and select the parallel sentences using various filtering methods. We adopt a moderate strategy, which retrieves candidate documents for sentences.

Parallel sentence identification methods can be classified into two different approaches: binary classification (Munteanu and Marcu 2005; Tillmann 2009; Smith et al. 2010; Stefanescu et al. 2012) and translation similarity measures (Utiyama and Isahara 2003; Fung and Cheung 2004; Abdul-Rauf and Schwenk 2011). Similar features such as word-overlap- and sentence-length-based features are used in both of these approaches. We believe that a machine learning approach can be more discriminative with respect to the features, and thus we adopt a binary classification approach with a novel classifier training and testing method and Chinese character features.

Few studies have been conducted for extracting parallel sentences from quasi-comparable corpora. We are aware of only two previous efforts. Fung and Cheung (2004) proposed a multi-level bootstrapping approach. Wu and Fung (2005) exploited generic bracketing inversion transduction grammars (ITGs) for this task. Our approach differs from the previous studies in that we extend the approach for comparable corpora in several aspects to make it work well for quasi-comparable corpora.

7.3.6 Conclusion and Future Work

In this section, we proposed a novel method of classifier training and testing that simulates the real parallel sentence extraction process. Furthermore, we used linguistic knowledge of Chinese character features. Experimental results of parallel sentence extraction from both quasi-comparable and comparable corpora indicated that our proposed system performs significantly better than the previous study.

As future work, firstly, because the scales of both the quasi-comparable and comparable corpora used in our experiments are small, we plan to collect more data and conduct large-scale experiments. Secondly, as parallel sentences rarely exist in quasi-comparable corpora, we plan to extend our system to parallel fragment extraction. Finally, since our study showed that Chinese character features are helpful for Chinese–Japanese parallel sentence extraction, we plan to apply a similar idea to other language pairs by using cognates.

References

Abdul-Rauf, S., & Schwenk, H. (2011). Parallel sentence generation from comparable corpora for improved SMT. *Machine Translation, 25*(4), 341–375.

Adafre, S. F., & de Rijke, M. (2006). Finding similar sentences across multiple languages in Wikipedia. In *Proceedings of EACL* (pp. 62–69).

Armstrong, S., Kempen, M., McKelvie, D., Petitpierre, D., Rapp, R., & Thompson, H. (1998). Multilingual corpora for cooperation. In *Proceedings of the 1st International Conference on Linguistic Resources and Evaluation (LREC)* (Vol. 2, pp. 975–980), Granada.

Brants, T. (2000). TnT – A statistical part-of-speech tagger. In *Proceedings of the 6th Applied Natural Language Processing Conference* (pp. 224–231).

Chiao, Y.-C., Sta, J.-D., & Zweigenbaum, P. (2004). A novel approach to improve word translations extraction from non-parallel, comparable corpora. In *Proceedings of the International Joint Conference on Natural Language Processing*, Hainan, China, AFNLP, 2004.

Chu, C., Nakazawa, T., & Kurohashi, S. (2011). Japanese-Chinese phrase alignment using common Chinese characters information. In *Proceedings of MT Summit XIII* (pp. 475–482), Xiamen, China, September.

Chu, C., Nakazawa, T., Kawahara, D., & Kurohashi, S. (2012a, May). Exploiting shared Chinese characters in Chinese word segmentation optimization for Chinese-Japanese machine translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT2012)* (pp. 35–42), Trento, Italy.

Chu, C., Nakazawa, T., Kawahara, D., & Kurohashi, S. (2012b, May). Chinese characters mapping table of Japanese, Traditional Chinese and Simplified Chinese. In *Proceedings of the Eighth Conference on International Language Resources and Evaluation (LREC2012)* (pp. 2149–2152), Istanbul, Turkey.

Chu, C., Nakazawa, T., Kawahara, D., & Kurohashi, S. (2013, August). Chinese–Japanese parallel sentence extraction from quasi–comparable corpora. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora* (pp. 34–42). Association for Computational Linguistics, Sofia, Bulgaria.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics, 19*(1), 61–74.

Fung, P., & Cheung, P. (2004). Multi-level bootstrapping for extracting parallel sentences from a quasi-comparable corpus. In *Proceedings of Coling 2004* (pp. 1051–1057), Geneva, Switzerland, Aug 23–Aug 27. COLING.

Fung, P., & McKeown, K. (1997). Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora* (pp. 192–202), Hong Kong.

Fung, P., & Yee, L. Y. (1998). An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of COLING-ACL 1998* (Vol. 1, pp. 414–420), Montreal.

Goh, C. L., Asahara, M., & Matsumoto, Y. (2005). Building a Japanese-Chinese dictionary using kanji/hanzi conversion. In *Proceedings of the International Joint Conference on Natural Language Processing* (pp. 670–681).

Jongejan, B., & Dalianis, H. (2009). Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike. 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* (pp. 145–153).

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In D. Lin, & D. Wu (Eds.), *Proceedings of EMNLP 2004* (pp. 388–395). Association for Computational Linguistics, Barcelona, Spain.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit* (pp. 79–86), Phuket, Thailand.

Koehn, P., Hoang, H., Birch, A., et al. (2007, June). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions* (pp. 177–180), Association for Computational Linguistics, Prague, Czech Republic.

Kondrak, G., Marcu, D., & Knight, K. (2003). Cognates can improve statistical translation models. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 46–48).

Kurohashi, S., Nakamura, T., Matsumoto, Y., & Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*(pp. 22–28).

Laws, F., Michelbacher, L., Dorow, B., Scheible, C., Heid, U., & Schütze, H. (2010). A linguistically grounded graph model for bilingual lexicon extraction. In *Proceedings of Coling, Poster Volume* (pp. 614–622).

Munteanu, D. S., & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics, 31*(4), 477–504.

Munteanu, D. S., & Marcu, D. (2006, July). Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (pp. 81–88). Association for Computational Linguistics, Sydney, Australia.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 160–167). Association for Computational Linguistics, Sapporo, Japan.

Papineni, K., Roukos, S.,Ward, T., & Zhu, W-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics* (pp. 311–318), Philadelphia, PA.

Rapp, R. (1995). Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics* (pp. 320–322), Cambridge, MA.

Rapp, R. (1996). *Die Berechnung von Assoziatonen*. Hildesheim: Olms.

Rapp, R. (1999). Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (pp. 519–526), College Park, MD.

Rapp, R., & Martin Vide, C. (2007). Statistical machine translation without parallel corpora. In G. Rehm, A. Witt, & L. Lemnitzer (Eds.), *Datenstrukturen für linguistische Ressourcen und ihre Anwendungen/Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007* (pp. 231–240). Gunter Narr Verlag, Tübingen.

Rapp, R., & Zock, M. (2010). Automatic dictionary expansion using non-parallel corpora. In A. Fink, B. Lausen, W. Seidel, & A. Ultsch (Eds.) *Advances in Data Analysis, Data Handling and Business Intelligence. Proceedings of the 32nd Annual Meeting of the GfKl*, 2008. Springer, Heidelberg.

Rapp, R., Sharoff, S., & Babych, B. (2012). Identifying word translations from comparable documents without a seed lexicon. In *Proceedings of LREC 2012*, Istanbul.

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the Workshop on Comparing Corpora (WCC '00)* (Vol. 9, pp. 1–6).

Rumelhart, D. E., & McClelland, J. L. (1987). *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Vol. 1: Foundations*. MIT Press.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. *International Conference on New Methods in Language Processing* (pp. 44–49).

Sharoff, S., Kopotev, M., Erjavec, T., Feldman, A., & Divjak, D. (2008). Designing and evaluating a Russian tagset. In *Proceedings of the Sixth Language Resources and Evaluation Conference, LREC 2008* (pp. 279–285), Marrakech.

Smith, J. R., Quirk, Ch., & Toutanova, K. (2010, June). Extracting parallel sentences from comparable corpora using document level alignment. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 403–411), Association for Computational Linguistics, Los Angeles, CA.

Stefanescu, D., Ion, R., & Hunsicker, S. (2012, May). Hybrid parallel sentence mining from comparable corpora. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT2012)* (pp. 117–128), Trento, Italy.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., et al. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Tan, Ch. L., & Nagao, M. (1995). Automatic alignment of Japanese-Chinese bilingual texts. *IEICE Transactions on Information and Systems, E78-D*(1), 68–76.

Tillmann, Ch. (2009, August). A beam-search extraction algorithm for comparable data. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers* (pp. 225–228), Association for Computational Linguistics, Suntec, Singapore.

Utiyama, M., & Isahara, H. (2003, July). Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (pp. 72–79), Association for Computational Linguistics, Sapporo, Japan.

Wu, D., & Fung, P. (2005). Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-2005)*, Jeju, Korea.

Zhao, B., & Vogel, S. (2002). Adaptive parallel sentences mining from web a bilingual news collections. In *Proceedings of the 2002 I.E. International Conference on Data Mining* (pp. 745–748), IEEE Computer Society, Maebashi City, Japan.

Chapter 8
Appendices

Ahmet Aker, Radu Ion, Nikos Mastropavlos, Monica Paramita, Mārcis Pinnis, Dan Ştefănescu, Fangzhong Su, Gregor Thurmair, Elena Irimia, Nikola Ljubešić, Evangelos Kanoulas, Judita Preiss, Rob Gaizauskas, Paul Clough, Emma Barker, Nikos Glaros, Tiberiu Boroș, Inguna Skadiņa, and Andrejs Vasiļjevs

8.1 Introduction

The tools that were developed through the ACCURAT project and are presented in this book are packed into the ACCURAT toolkit[1] (Pinnis et al. 2012a)—a collection of tools that are capable of collecting comparable corpora, analysing and extracting parallel data. The ACCURAT toolkit produces

- **comparable text** corpora containing a significant amount of mappable textual data

Chapter editor: Inguna Skadiņa

[1] http://www.accurat-project.eu/

A. Aker · M. Paramita · E. Kanoulas · J. Preiss · R. Gaizauskas · P. Clough · E. Barker
University of Sheffield, Sheffield, UK

R. Ion · D. Ştefănescu · E. Irimia · T. Boroș
Romanian Academy, Research Institute for Artificial Intelligence, Bucharest, Romania

N. Mastropavlos · N. Glaros
Institute for Language and Speech Processing, Athens, Greece

M. Pinnis · I. Skadiņa (✉) · A. Vasiļjevs
Tilde, Riga, Latvia
e-mail: Inguna.Skadina@tilde.lv

F. Su
University of Leeds, Leeds, UK

G. Thurmair
Linguatec, Munich, Germany

N. Ljubešić
Faculty of Humanities and Social Sciences, University of Zagreb, Zagreb, Croatia

I. Skadiņa et al. (eds.), *Using Comparable Corpora for Under-Resourced Areas of Machine Translation*, Theory and Applications of Natural Language Processing,
https://doi.org/10.1007/978-3-319-99004-0_8

- **comparable document pairs** with comparability scores, allowing to estimate the overall comparability of corpora
- **parallel sentences** which can be used as additional parallel data sources for statistical translation model learning
- **terminology dictionaries**—this type of data is expected to improve domain-dependent translation
- **named entity (NE) dictionaries**.

Although the toolkit may be used for parallel data acquisition for open (broad) domain systems, it will be most beneficial for under-resourced languages or specific domains that are not covered by available parallel resources.

The ACCURAT Toolkit provides three workflows (Fig. 8.1) that use different chains of tools included in the toolkit:

- The **workflow for building comparable corpora** supports gathering large amounts of comparable textual data from the web for three different types of comparable corpora: (1) corpora consisting of news articles published concurrently, (2) inter-language linked Wikipedia articles and (3) corpora that cover domain-specific language.
- The **workflow for parallel data mining from comparable corpora** (PDMWF) aligns comparable corpora in the document level and then outputs pairs of parallel, quasi-parallel and strongly comparable sentence pairs that are ready for SMT training.
- The **workflow for named entity** (NE) **and terminology extraction** (TE) **and mapping** (NERTEWF) from comparable corpora extracts data in a dictionary-like format. Providing a list of document pairs, the workflow tags NEs or terms in all documents using language-specific taggers and performs multi-lingual NE or term mapping, thereby producing bilingual NE or term dictionaries. The workflow also accepts pre-processed documents, thus skipping the tagging process.

Since all tools use command line interfaces, task automation and workflow specification can be done with simple console/terminal scripts.

8.2 Tools for Building a Comparable Corpus from the Web

ACCURAT has investigated efficient methods and developed tools for identifying and gathering large amounts of comparable textual data from the web.[2] The developed tools can be used for gathering large amounts of comparable textual data from the web for three different types of comparable corpora: (1) corpora consisting of

[2]Whilst they may not be directly applicable, it is straightforward to adopt and apply our methods for building comparable corpora from the web to digital archives or other off-line textual data collections that are very large.

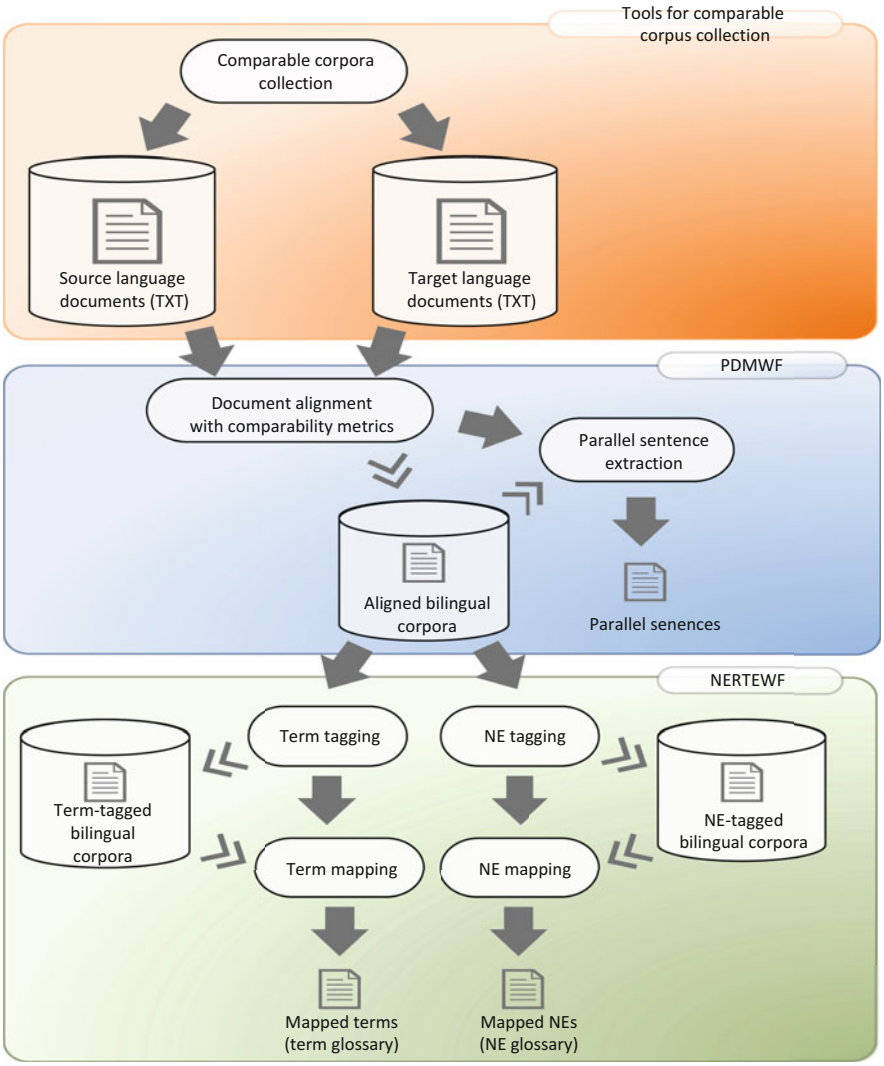


Fig. 8.1 Workflows of the ACCURAT toolkit

news articles published concurrently, (2) inter-language linked Wikipedia articles and (3) corpora that cover domain-specific language. This section presents the most successful of the corpus collection approaches that have been refined and documented to a level suitable for public release:

- *A Workflow-based Corpora Crawler*
- *Focussed Monolingual Crawler (FMC)*
- *Wikipedia Retrieval Tool*
- *News Information Downloader using RSS feeds*

- *News Information Downloader using Google News Search*
- *News Text Crawler and RSS Feed gatherer*
- *News Article Alignment and Downloading Tool.*

8.2.1 A Workflow-Based Corpora Crawler

We provide a tool for automatic extraction of a parallel or strongly comparable corpus. The logic of the application is controlled from within the flow editor which enables the user to easily create and manage workflows, thus having a global view of the extraction process. Note that this is not intended as a standalone crawler but more as a development system for data extraction applications.

By using the workflow approach, we are trying to provide the means for interaction between such text-based tools and other applications. There are two ways to accomplish this:

1. When using the console application, we give a simple regular expression-driven mechanism for input/output control. When the flow is executed, the output of each unit is processed and an input for the next unit is generated. If input/output (I/O) regular expressions are defined, these are applied to I/O data.
2. When the first method is unusable, the user can create plug-ins in order to implement the needed functionality. Plug-ins are .NET assemblies that implement the *ProcessingBlock* or *DecisionBlock* interfaces.

The workflow-based crawler is shipped with plug-ins and workflows (saved in XML files) for crawling Wikipedia and Europarl sites.

The use of a workflow gives the means for high scalability and integration of modules written in different programming languages or interpreters. This system gives the advantage of organising the logic of the application around processing units and decision blocks. The user can alter the behaviour of the global application by adding new blocks or modifying the way the I/O data is being handled. Another advantage is that the independent processing modules are unloaded when no data is available, in order to preserve computational resources.

We start by creating a basic crawling workflow. We refer to the units that do the actual work as *active blocks*, as opposed to the start and end nodes that are just *visual markers*.

There are two types of active blocks: *decision blocks* and *processing units*. Every block has an external application (script, compiled program, etc.)/plug-in associated with it, that takes the data from the preceding block, processes it and passes the result to the next block in the chain. By clicking on the active blocks, the user can edit the following parameters:

1. *Name*—represents the label on the block that will be displayed on the screen.
2. *Execution parameters*
   2.1. *Executable path*—path to the application that will be executed when the block is invoked. It can be a stand-alone application or an interpreter for the script.
   2.2. *Command line parameters*—will be passed to the application. We use special keywords like '$script' for the script filename or '$input' for the input produced by the parent node.
   2.3. *PlugInDLL*—the full path to a C# plugin DLL which implements the *ProcessingBlock* or *DecisionBlock* interfaces (included in the distribution of this crawler).
   2.4. *Script path*—should be used only with interpreted languages and will be passed as a command line argument.
3. Regular expressions (applied only in case of external applications invoked by the respective block)
   3.1. *InputRegex*—this will be applied to the text input before it is passed to the external application (script, compiled application, etc.). It must have capturing parentheses, because only the captured text is actually passed on.

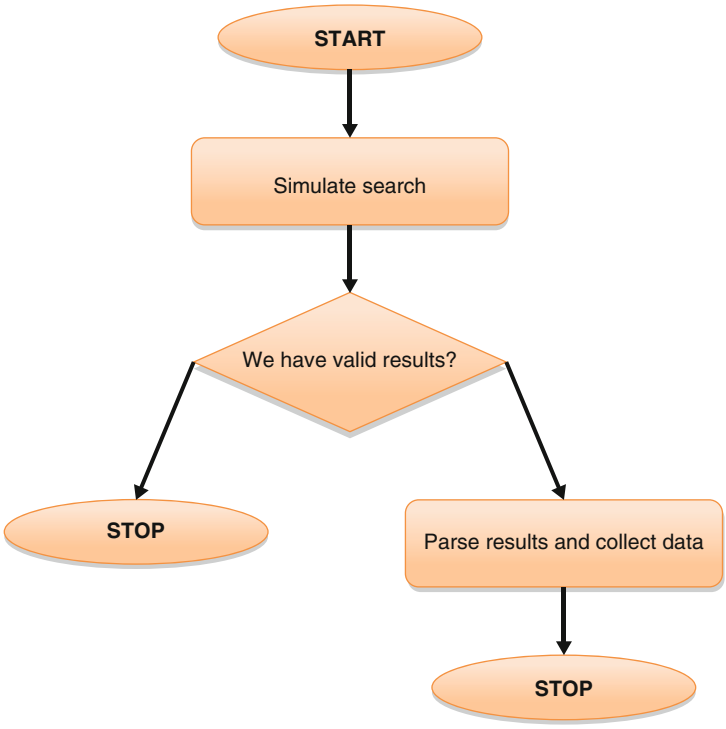


Fig. 8.2 The typical crawling workflow (if we are sure that all results are valid, we may do without the decision block)

3.2. *OuputRegex*—used to pre-process the output of the external application before it is passed to the next block.
3.3. *ConditionRegex*—used only on decision blocks. Produces 'true' if the output matches the regular expression and 'false' otherwise.

After the workflow is defined, the user may save it and execute it from the toolbar above the workflow creation area. Depending on the selected plug-ins, a dialog will open that will ask for information such as the destination folder (Fig. 8.2).

8.2.2 Focussed Monolingual Crawler

The *Focussed Monolingual Crawler* (FMC) tool is used to collect narrow domain bi(multi)lingual comparable corpora from the web. It does so by making a separate crawl for each specified language. Each time, it retrieves only web pages that are relevant to a pre-defined narrow domain or topic. The comparability of the bi(multi) lingual documents retrieved is achieved by ensuring that, for each language specified, the FMC is made to return web documents that are all close to the same topic.

Given a language pair (or a set of languages) and a topic, the user has to first create a list of topic-specific single- or multi-word terms as well as a simple list of URLs that are considered to be highly relevant to the topic in question. These data (input to the FMC) have to be prepared for each language.

The list of (generally) multi-word expressions related to a specific topic can be created either manually (possibly with the aid of some available online resources, e.g., Eurovoc[3]) or can be automatically extracted from small topic-specific corpora using tf-idf (term frequency–inverse document frequency) and term extraction algorithms.

The list of topic-related URLs that the FMC treats as seed URLs can be constructed semi-automatically (using directories from known search engines, e.g., *Yahoo*, *Google*, dmoz) or automatically. One possible way to automate the construction of the URL list is to use BootCaT's (Baroni and Bernardini 2004) tuple generation algorithm, as follows: first generate a number of n-topic-term combinations, and then *Google* search them and keep the top 5 or 10 URL results from each search as candidates for the final seed URL list.

Once topic specific terms and the seed URL list have been generated, the user may then optionally choose to configure the crawler engine. That is, the user has the option to adjust various crawler settings prior to crawling start. For example the user can set file types to download (e.g. PDF, doc and xls), domain-filtering options (using regular expressions), self-terminating conditions (size or time limits), crawling politeness parameters, etc.

[3]The EU's multilingual thesaurus, http://eurovoc.europa.eu/

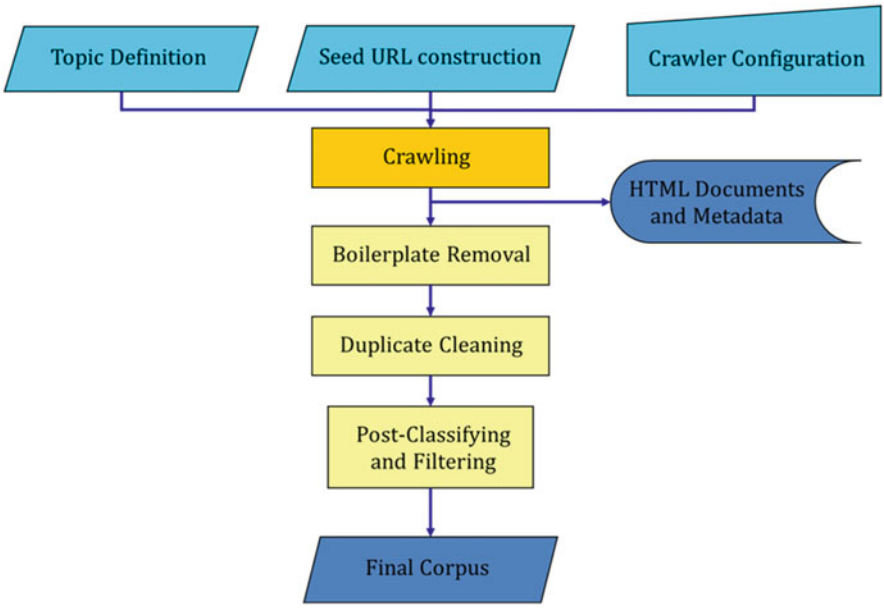


Fig. 8.3 FMC crawler workflow

Having completed the above steps, the user should next run FMC once per each language.

The main steps that the crawler executes are illustrated in Fig. 8.3.

8.2.3 Wikipedia Retrieval Tool

Wikipedia articles that describe the same topic are connected to each other by Wikipedia *inter-language links*, enabling us to extract a corpus that is already aligned in document level. Even though these documents contain the same topic, their ranges of comparability vary widely. Some articles might be translations of each other; however, some of them might be written independently and do not contain any shared information. Inclusion of these documents in the corpora might reduce the performance of MT. Therefore, a retrieval tool is needed to identify and gather only the comparable documents.

Different from a crawling tool, this retrieval tool makes use of available Wikipedia dump data (available for download in http://dumps.wikimedia.org) which contains extensive information of Wikipedia documents, including inter-language links between bilingual articles.

This tool aims to identify and retrieve comparable documents by specifically looking for pairs that contain similar sentences (sentences with information overlap, such as links—also referred to as anchor texts—and words). Our approach, which is

language independent, is based on the method proposed by Adafre and de Rijke (2006). This approach uses anchor text information from the Wikipedia articles to identify parallel sentences. First, a bilingual lexicon is constructed by extracting all document titles that are connected by the Wikipedia inter-language links. This lexicon is then used to translate all anchor texts found in the non-English articles into English. We then calculate the Jaccard coefficient to measure the similarity of sentences, pairing each sentence in the shorter document to the highest scoring sentence in the longer document. Finally, a measure of document-level similarity is computed based on averaging the scores of the paired sentences. Document pairs whose scores fall above a pre-defined minimum threshold are considered to be comparable; those below are filtered out.

This tool needs Wikipedia documents of the source and target languages along with the alignment file as an initial corpus. This retrieval method contains five main processes, as shown in Fig. 8.4:

1. *ExtractBilingualLexicon.pl*. This script builds bilingual lexicons by extracting document titles that are connected by *inter-language links* by Wikipedia. Therefore, this retrieval tool does not need any linguistic resources to perform translation.
2. *FilterWikipedia.pl*. This script filters out unnecessary information in Wikipedia documents, such as footnotes, table formatting, images, etc.
3. *SentenceSplitter.pl*. This script aims to split documents into sentences, enabling further processing to find information overlap in sentence level.
4. *ReplaceAnchorsUsingBilingualLexicon.pl*. This script replaces all links (anchor texts) in the source documents with its corresponding text in the target language if they are available in the bilingual lexicon.
5. *ComparabilityMeasurerOnSentenceLevel.pl*. Lastly this script measures the comparability of the documents on the sentence level on all the available document pairs in the specified language pair by measuring word overlap.

8.2.4 News Information Downloader Using RSS Feeds

The aim of this tool is to collect news articles from the web. To do this, it uses monolingual RSS feeds which are XML structured documents. It parses each of the XML documents and records the news published within each RSS feed document.

The output of this tool is collected in pools to produce the comparable corpora. More precisely, for each language, a pool is generated that contains all the output of all these pools for that particular language. The entries in each pool are compared to the others to perform alignment between the news articles. For instance we compare the articles from the English pool with the ones in the German pool to produce a English–German comparable corpora. The alignment is performed using the *News Article Alignment and Content Downloading Tool* described below.

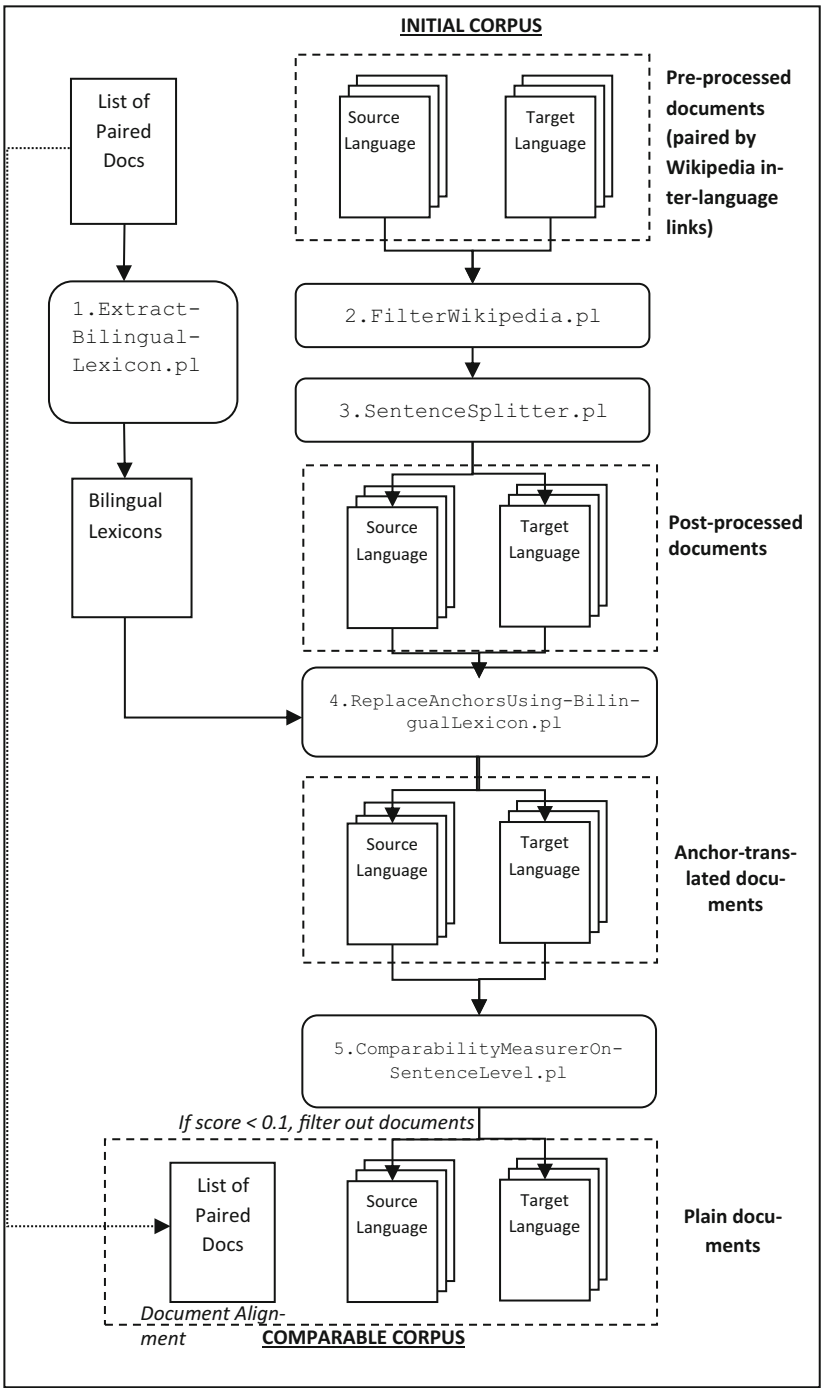


Fig. 8.4 Data flow diagram of Wikipedia retrieval

8.2.5 News Text Crawler and RSS Feed Gatherer

This crawler is suitable for extracting texts for parallel phrase extraction. Given a list of URLs, the tool observes any restrictions imposed on automatic programs visiting the webpages specified in the domain's *robots.txt* files. The focus of the tool is news texts, with the assumption that the same news stories are likely to be covered in similar ways in multiple languages. Some online newspapers do not allow a full automatic crawl of their webpages (by specifying this in their '*robots.txt*' file), instead they provide an RSS feed. Therefore our 'crawler' consists of two parts: *URL crawl* and *RSS feed gathering*.

Given files containing URLs to crawl, the crawler (*retrieve_crawled.pl*) enforces '*robots.txt'* compatibility and tags downloaded files with the time-stamp of the download. To prevent duplicates (should the crawl be restarted), *md5sums* of the new pages are compared against existing downloaded pages—in case of duplication, the more recent download is discarded. Therefore, the crawler can be restarted to allow re-crawls of news sites or could even be run in a continuous loop through the use of a wrapper script (such a script is not included and would need to be created by the user). Given files containing lists of RSS feeds, the RSS feed retrieval tool (r*etrieve_rss_feeds.pl*) downloads the most recent RSS stories from each feed. As with the URL crawler, the user can implement either of two options for repeated downloading:

- Set up the program to repeat with a time-based job scheduler (e.g., every 10 min).
- A wrapper script to repeat the program's execution—a loop returning to the start of the RSS links list, possibly with an enforced time delay, a maximum number of repeats or set to repeat infinitely.

8.2.6 News Article Alignment and Downloading Tool

The purpose of this tool is to (1) align or pair news articles written in different languages and (2) to download the content of the paired news URLs. For a pair of given news URLs, it streams the HTML codes, extracts the text from them and saves the extracted text in separate files. The files are encoded in UTF-8.

In step (1), it uses the titles of the news articles and the date information to produce alignment between the articles written in the source and target languages.

To perform step (2), the tool uses an HTML parser[4] to construct a parsing tree from the HTML document, following the Document Object Model (DOM).[5] Within this parsing tree, the tool checks only the BODY and the TITLE tags of the

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

[5] http://www.w3.org/DOM/

document. It ignores the SCRIPT, TABLE and FORM tags within the BODY tag, as these are likely not to contain relevant text. In addition to this, it ignores parts of the BODY that contain enumeration of information, such as menu items, copyright information, privacy notices and navigation hyperlinks. Furthermore, short texts are ignored as well, as they are likely to contain advertisements. The text identified by the tool as pure is then prepared for saving. The tool replaces any ASCII coding within the text and adds a dot on the end of each paragraph if it does not end with a punctuation mark.

8.3 Parallel Data-Mining Workflow

The parallel data-mining workflow aims at providing parallel textual unit mining (sentences and/or phrases) from comparable corpora (Fig. 8.5). The assumption that we have worked with is that, given two collections of source and target documents, these documents at first need to be aligned as to their probability of containing parallel textual units so that the parallel textual unit extractors (CPU intensive algorithms) do not have to search in each possible document pair. After the document alignment has been found, a generic parallel textual unit extractor can search for parallel pairs only in the offered document pairs.

The toolkit contains four applications that implement the 'generic document aligner/document pair classifier' operation (EMACC, ComMetric, DictMetric and Feature-based document pair classifier) and three applications that take over the role of a 'parallel textual unit extractor' operation (PEXACC (Ion 2012), LEXACC (Ştefănescu et al. 2012), and MaxEnt Extract).

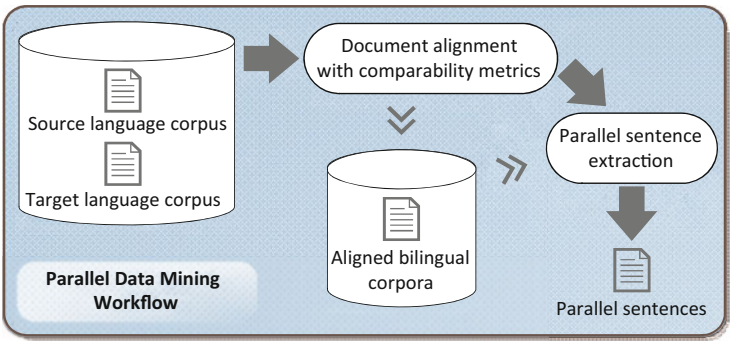


Fig. 8.5 Overview of parallel data mining workflow

8.3.1 Tools to Identify Comparable Documents and to Extract Parallel Sentences and/or Phrases from Them

This section covers the tools that classify or rank document pairs according to their comparability levels. The tools of the ACCURAT toolkit that deal with document pairing are

- *ComMetric*—a toolkit for measuring comparability of comparable documents
- *DictMetric*—a toolkit for measuring comparability of comparable documents
- *Features extractor and document pair classifier*
- *EMACC*—a textual unit aligner for comparable corpora using Expectation-Maximisation.

8.3.1.1 ComMetric: A Toolkit for Measuring Comparability of Comparable Documents

ComMetric (Su and Babych 2012a) is designed to measure the comparability levels of document pairs via a cosine measure. The toolkit can compute comparability scores for both monolingual document pairs and bi-lingual document pairs (via using our translation toolkit). Also, given the fact that for some under-resourced languages, it is usually difficult to obtain satisfactory language-processing resources or tools (e.g., POS taggers, machine-readable lexicons, stop word lists, word stemmers and lemmatisers), *ComMetric* at first translates monolingual documents into English (if the MT system, which can translate the non-English texts into English, is available) and then measures the comparability levels utilising the rich language resources for English.

ComMetric contains two modules: text translation and the cosine-based comparability computation.

The translation toolkit allows users to translate text collections from a source language to a target language by using the available *Google* translation java API, *Microsoft Bing* translation java API or DFKI's *MT-serverland*.

The translation toolkit supports two different manners of translation. For each translation call, you can send either a text string or a string array for translation. By default, the toolkit will call Manner 1 unless the user specifies using string array translation (Manner 2).

Also, the toolkit supports two different inputs of source documents that will be translated. (1) Users can put all the documents to be translated in a directory, and the toolkit will read all the documents from that directory for translation. (2) Sometimes the documents to be translated are from different directories. In this case, the user can provide a file that lists all the documents to be translated with full path, and the toolkit will read the documents using this file and proceed with the translation. Finally, apart from outputting the translated documents, a file that lists the full path of each translated document will be generated as well.

For comparability computation the toolkit at first calls the *Standford CoreNLP tool* (available at http://nlp.stanford.edu/software/corenlp.shtml) for POS-tagging and word tokenisation. Then, *JWI* (*MIT Java WordNet Interface*) is called for *WordNet*-based stemming. After word stemming, the stemmed texts are converted into lexical vectors. The comparability metric takes four different types of features into account:

1. *Lexical features*—the stemmed lexical vectors with stop-word filtering.
2. *Structural feature*—number of sentences and number of content words (using the POS-tagged result) of each document.
3. *Keyword feature*—the top-20 keywords (based on TFIDF weight) of each document.
4. *Named entity feature*—named entities of each document by using Stanford NER module in the CoreNLP tool.

Finally, the toolkit applies the cosine similarity measure on lexical features, keyword features and named entity features individually and then uses a weighted average strategy to combine these cosine scores into the comparability metric. Document pairs with a comparability score $\geq$ threshold (a predefined value, between 0 and 1) are returned as output.

8.3.1.2 DictMetric: A Toolkit for Measuring Comparability of Comparable Documents

DictMetric (Su and Babych 2012b) is designed to measure the comparability levels of document pairs via cosine measure. The toolkit can compute comparability scores for both monolingual document pairs and bi-lingual document pairs. Overall, the toolkit contains two modules: text translation by lexical mapping and cosine-based comparability computation.

The toolkit supports two types of text translation. First, for non-English and English language pairs, we translate the non-English texts into English by using lexical mapping from the available *GIZA++*based bilingual dictionaries. Second, for non-English language pairs (i.e. both the source and target languages are not English), the toolkit can either translate source language (i.e. Greek or Romanian) texts into the target language (i.e. Romanian or German) using a dictionary or use English as the pivot language and translate both source and target language texts into English.

For comparability computation, at first, the toolkit calls the *Standford POS-tagger* (available at http://nlp.stanford.edu/software/tagger.shtml) for POS-tagging and word tokenisation. Then *JWI* (*MIT Java Wordnet Interface*) is called for *WordNet*-based English word stemming. After word stemming for the English language, the stemmed texts are converted into index vectors. If the translated texts are not in English, then the word stemming step will be skipped and directly go to feature vector conversion. Finally, the toolkit computes the comparability score of document pairs by applying the cosine similarity measure on the index vectors. Document pairs with

a cosine score $\geq$ threshold (a predefined value between 0 and 1) are returned as output.

8.3.1.3 Features Extractor and Document Pair Classifier

This tool aims to select and pair documents that are judged to be comparable from this set. Given a list of source documents and target documents, this tool will use all possible pairs of documents and extract numerous features from them. These features will then be used by the classifier to predict the comparability class of all the given pairs, enabling a subset of document pairs to be chosen as comparable documents.

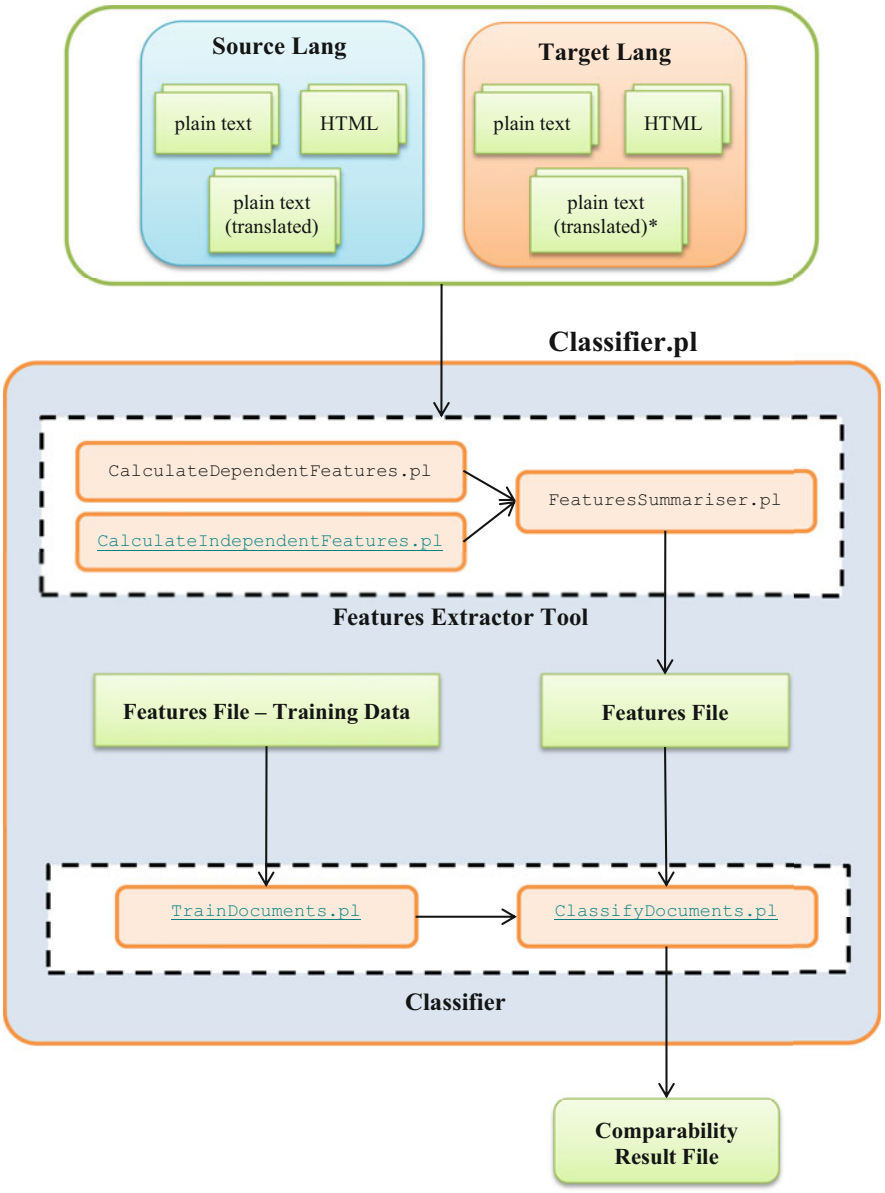
This tool contains two processes: features extractor and classifier which is wrapped using '*Classifier.pl*'. The main workflow is described in Fig. 8.6.

The feature extractor tool will extract language-dependent features and language-independent features from the pairs. To enable all features to be extracted correctly, the tool will require the English translation of documents (to calculate language-dependent features) and *HTML* documents (to calculate language-independent features). These features will be extracted using '*CalculateDependentFeatures.pl*' and '*CalculateIndependentFeatures.pl*' and later summarised using '*FeaturesSummariser.pl*'. The output of the Features Extractor tool will contain the score of all extracted features for all the document pairs. This output is then passed to the Classifier.

This classifier is made of two major components: (1) a binary classifier which used Thorsten Joachim's SVM^{light} to implement the method, and (2) an error correction schema. At the moment, the classifier has already been trained using the Initial Comparable Corpora (Skadiņa et al. 2010). However, users may use different training data for the classifier by running '*TrainDocuments.pl*'. All the previously extracted features will be passed to '*ClassifyDocuments.pl*' together with the classifier model resulting from the training process, and the final output of this tool consists of selected document pairs and their predicted comparability levels.

8.3.1.4 EMACC: A Textual Unit Aligner for Comparable Corpora Using Expectation-Maximisation

EMACC is designed to align (translation-wise) different types of textual units such as documents, paragraphs or sentences in order to reduce the search space for subsequent alignment tasks. For instance, suppose that we want to word-align a bilingual comparable corpus consisting of M documents per language, each with k words, using the IBM-1 word alignment algorithm (Brown et al. 1993). For each source word, this algorithm searches the target words that have a maximum translation probability with the source word. Aligning all the words in our corpus with no regard to document boundaries would yield a time complexity of k^2M^2 operations. The alternative would be in finding a 1:p (with p a small positive integer, usually 1, 2 or 3)



* translated target document is only needed for non-English document.

Fig. 8.6 Workflow of features extractor and classifier

document assignment (a set of aligned document pairs) that would enforce the 'no search outside the document boundary' condition when doing word alignment with the advantage of reducing the time complexity to k^2Mp operations. When M is large,

the reduction may actually be vital in getting a result in a reasonable amount of time. The downside of this simplification is the loss of information; two documents may not be correctly aligned, thus depriving the word-alignment algorithm of the part of the search space that would have contained the right alignments.

The principle behind *EMACC's* functionality is that translation equivalents (both correct and, surprisingly, incorrect) play a key role in document alignment. We have experimentally found that there is a certain balance between *the degree of correctness of translation equivalents* and *their ability to pin-point correct document alignments*. In other words, the paradox resides in the fact that if a certain pair of translation equivalents is not correct but the respective words appear only in documents that correctly align to one another, then that pair is very important to the alignment process. Conversely, if a pair of translation equivalents has a very high probability score (thus being correct), but appears in almost every possible pair of documents, then that pair is not informative to the alignment process and must be excluded. We see now that *EMACC* aims at finding the set of translation equivalents that is maximally informative with respect to the set of document alignments.

The basic workflow of *EMACC* is as follows:

1. Pre-compute the initial document alignment distribution according to the *D2* distribution (Ion et al. 2011).
2. Iteratively (greedily) find the best document alignment set (called an assignment) by computing a (translation equivalents based) similarity measure between each pair of source and target documents.
3. Re-estimate the probabilities of translation equivalents from the best assignment and resume from step 2 for a given number of steps.

8.3.2 Tools to Extract Parallel Sentences and/or Phrases from Comparable Documents

The tools that deal with parallel sentence/phrase extraction included in this section are:

- *PEXACC*—a parallel phrase extractor from comparable corpora
- *LEXACC*—fast parallel sentence mining from comparable corpora.

8.3.2.1 PEXACC: A Parallel Phrase Extractor from Comparable Corpora

Comparable corpora are inherently different from parallel corpora:

- The order of translation is not preserved. Thus, the significant search space optimisation from which all parallel alignment algorithms beneficiate (the 'translation window' out of which no translations are possible) is null in this case.

- The translations that one finds in comparable corpora are (most of them) accidental. Thus, the match between pieces of text is more difficult due to the fact that the meaning of the source phrase may only be approximately reproduced in one target candidate phrase.

Given these characteristics of comparable corpora, *PEXACC* will try to alleviate the effect of these problems by

- Trying (and scoring) all possible combinations of pairs of pieces of text (or textual units) so that each pair will receive a 'translation probability score' that the source textual unit is translated by the respective target textual unit
- Using relevance feedback loops which are a mechanism by which *PEXACC* learns new translations from the already mapped data so that new information may be found and added to the already found parallel data.

The general purpose of *PEXACC* is to extract parallel data from comparable corpora for use in SMT training of translation models. The granularity level of the textual units that can be mapped is customisable. Thus, *PEXACC* can align sentences and/or sub-sentential parts of text to which we will refer to as 'chunks'. We have imposed this restriction in order to deal with weakly comparable corpora which, generally, do not contain sentential translations.

The general processing flow of *PEXACC* is as follows:

1. For a list of document pairs found by *EMACC* and for each pair of documents from that list.
2. Split the source and the target documents at sentence/chunk level (depending on a configuration option).
3. Find all pairs of sentences/chunks that score above a certain threshold at 'translation probability'.
4. Apply *GIZA++* on all the pairs found at step 3 and add the resulting dictionary to the base dictionary that *PEXACC* uses.
5. Go to step 3 and rescore all the pairs of sentences/chunks. Repeat this loop for a number of steps (experimentally set to 5).

8.3.2.2 LEXACC: Fast Parallel Sentence Mining from Comparable Corpora

Finding parallel sentences in comparable corpora is confronted with the vast search space one has to consider, since no positional clues indicating parallel or partially parallel sentences are available.

The brute force approach is to analyse every element of the Cartesian product built between the two sets containing sentences in the source and target languages. This approach is clearly impractical, because the resulting algorithm would be very slow and/or would consume a lot of memory. In order to reduce the search space, we turned to a framework that belongs to Information Retrieval: Cross-Language Information Retrieval (CLIR). The idea is simple: use a search engine to find

sentences in the target corpus that are the most probable translations of a given sentence from the source corpus. The first step is to consider the target sentences as documents and index them. Then, for each sentence in the source corpus, one selects the content words and translates them into the target language according to a given dictionary. The translations are used to form a Boolean query that is then fed to the search engine. The top hits are considered to be translation candidates.

LEXACC is a parallel sentence extraction algorithm that uses a search engine (Lucene, http://lucene.apache.org/) to index the target document collection in order to retrieve the translation candidates for the input source sentence. Then, after a pre-filtering phase, it applies the translation similarity measure of PEXACC to select the desired parallel sentence pairs to which it assigns the computed PEXACC score.

8.4 The Workflow for Named Entity and Terminology Extraction and Mapping

The workflow for named entity and terminology extraction and mapping provides three different processing methods: named entity extraction and/or named entity mapping using bilingual comparable corpora, terminology extraction and/or term mapping using bilingual comparable corpora and term mapping using parallel data (Fig. 8.7).

Starting with named entity recognition, the toolkit features the first named entity recognition (NER) systems for Latvian and Lithuanian (Pinnis 2012). It also contains NER systems for English (through an *OpenNLP*[6] wrapper) and Romanian (*NERA*). The named entity mapping tool *NERA2* compares each NE from the source language to each NE from the target language using cognate-based methods. It also uses a GIZA++ format statistical dictionary to map NEs containing common nouns that are frequent in location names. This approach allows frequent NE mapping if the cognate-based method fails, therefore allowing increase of the recall of the mapper. Precision and recall can be tuned with a confidence score threshold.

ACCURAT terminology extraction workflow allows the extraction of term candidates from text documents, term tagging in the documents and bilingual term mapping in comparable corpora. The terminology mapper *TEA* is designed to map terms extracted from comparable or parallel documents. The method is language independent and can be applied if a translation equivalents table exists for a language pair. As input, the application requires term-tagged bilingual corpora aligned at the document level. The aligner maps terms based on two criteria (Pinnis et al. 2012b; Ştefănescu 2012): (1) a GIZA++-like translation equivalents table and (2) string similarity in terms of *Levenshtein* distance between term candidates.

[6]Open NLP—http://incubator.apache.org/opennlp/

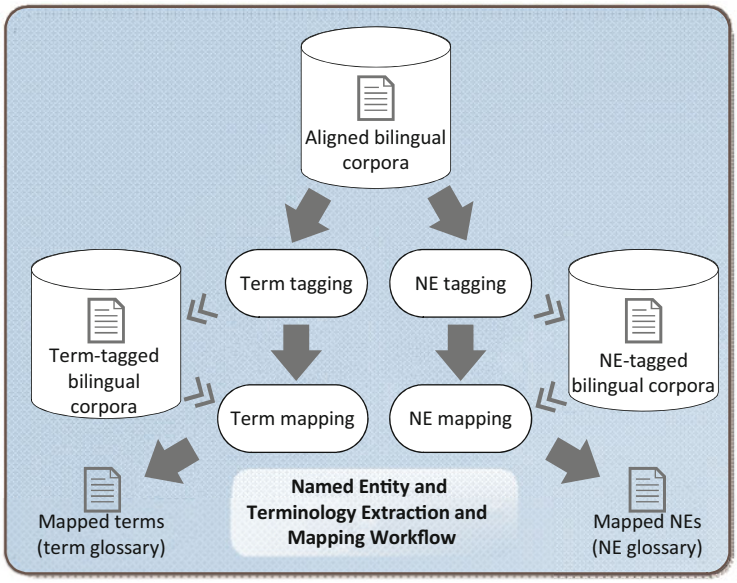


Fig. 8.7 Overview of the workflow for named entity (NE) and terminology extraction and mapping

8.4.1 Tools for Named Entity Recognition

This section covers the tools that perform named entity recognition and tools that are created to integrate out of ACCURAT project developed tools within the toolkit's general use case workflows. The tools included in this section of the ACCURAT toolkit are

- *TildeNER*—Latvian and Lithuanian named entity recognition tool
- *OpenNLPWrapper*—*OpenNLP* English named entity recognition system wrapper
- *NERA1*—Named Entity Recognition tool for English and Romanian.

8.4.1.1 TildeNER

TildeNER is a named entity recognition and classification system. The system contains workflows that allow not only named entity (NE) tagging of single files but also pre-processing and post-processing of plaintext documents and even whole directories. The system also contains a heavily configurable bootstrapping module, which allows training, improvement and evaluation of new NE models, if necessary. The system is originally designed and developed for Latvian and Lithuanian named entity recognition but is not limited to the design languages; therefore, new languages can be easily added with the included bootstrapping module.

The system's core functionality—classification is done by the *Stanford NER* conditional random field (*CRF*) classifier (some minor changes have been made to

the classifier in order to support additional feature functions and the *TildeNER* input and output data standards).

The tool allows tagging of plaintext or pre-processed tab-separated (tokenised, POS-tagged, lemmatised) documents, and it allows the results to be saved in an *MUC-7* compliant plaintext mark-up or as tab-separated (tokenised, POS-tagged, lemmatised and NE-tagged) documents. The architecture of the *TildeNER* bootstrapping system is shown in Fig. 8.8. The system requires an *MUC-7* compliant annotated corpus (seed list, development data and test data) and an unlabelled data

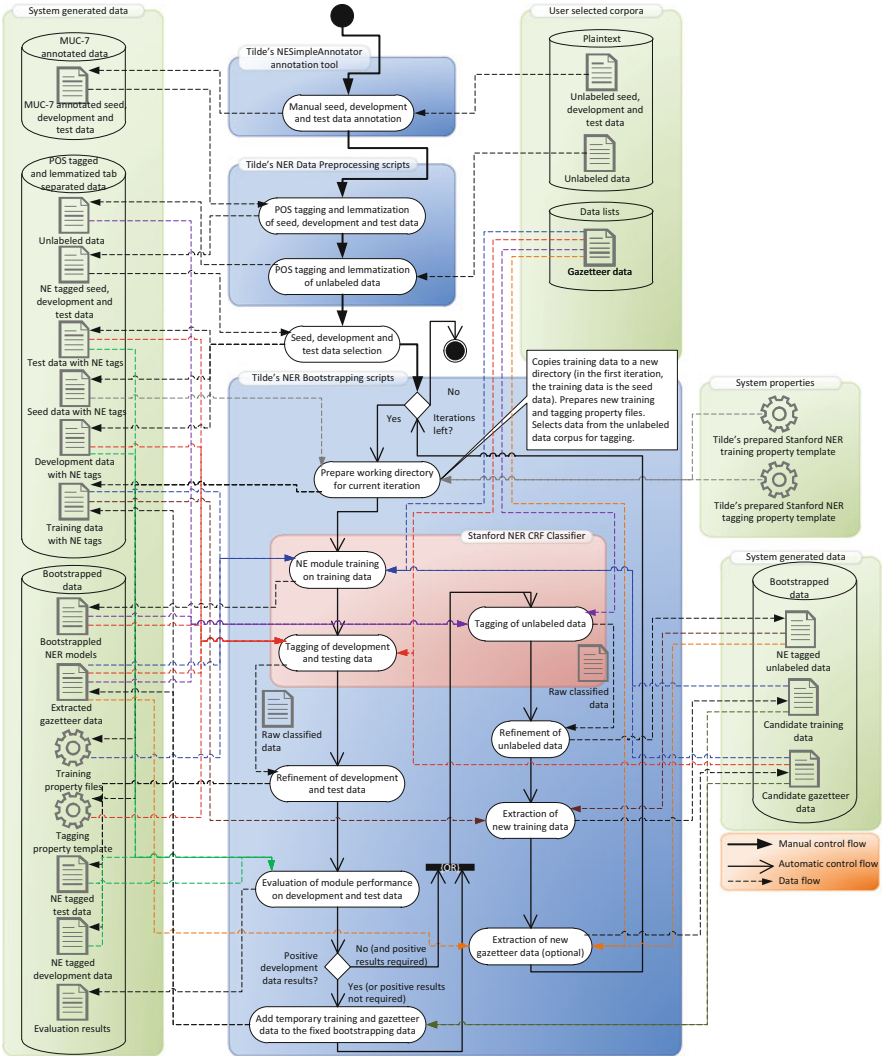


Fig. 8.8 TildeNER bootstrapping architecture

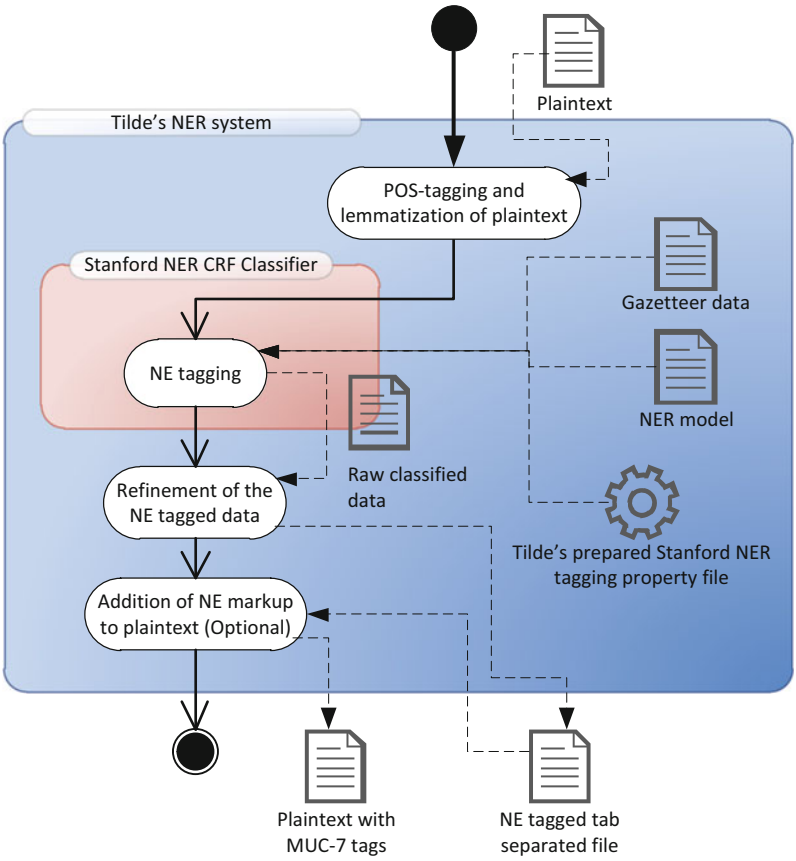


Fig. 8.9 Sample workflow of plaintext to MUC-7 annotated data tagging

corpus in order to train an NER model (an annotation tool is included in the toolkit). Gazetteer data can also be provided (but is not mandatory) in order to train the system.

The system iteratively trains new NER models on the training data of the particular iteration. In the first iteration, the training data is the seed data, but, in the further iterations, new data is acquired by tagging the unlabelled data corpus and selecting new candidate training sentences based on uniqueness constraints and sentence ranking. The sentences are ranked according to the classifier-assigned NE token average probabilities. A threshold is used to control that low likelihood NE-tagged tokens do not get selected as new candidate training data.

In each of the iterations, the system is evaluated on development and test data. We use the development data in order to fine-tune the system. An option allows usage of only positive iteration candidate training data for further iterations (it has proven to give the best results).

The system also allows execution of several refinements which allow fine-tuning of the system towards increasing recall (increases also the *F*-measure) or precision of

the system. Some of the refinements also try to correct corrupt NE tagging (missing quotation marks, web addresses incorrectly tagged as entities, etc.). The Latvian and Lithuanian models have been bootstrapped using fine-tuning for precision and using only positive iterations.

The system also allows automatic extraction of gazetteer data which is then used in order to train new NER models in further iterations.

As a sample, Fig. 8.9 shows the tagging workflow of a plaintext document. The results are saved in the *MUC-7* annotated data format.

8.4.1.2 OpenNLP Wrapper

In the multi-lingual NE and term mapper, we make use of *OpenNLP* to tag named entities for the English documents. *OpenNLP* is an existing tool and is not implemented within the ACCURAT project. The output of this system is, therefore, different from the input format of the NE mapper. The wrapper

- Enables that the output of *OpenNLP* is of the same format as the input files to the mapper
- Provides a scenario to users where the mapper can be run on existing annotated data
- Enables users to use other NER systems to prepare the input to the mapper.

8.4.1.3 NERA1: Named Entity Recognition for English and Romanian

NERA1 tool is designed to identify and label named entities in raw or already pre-processed texts. It is designed to work for English and Romanian and to identify 6 types of Named Entities: *PERSON*, *ORGANIZATION*, *LOCATION*, *PRODUCT*, *DATE*, *TIME* and *MONEY*. The current version focusses mainly on the first 3 types and works without any use of gazetteers. First, it identifies boundaries of named entities using regular expressions, and, then, it labels the entities according to a Maximum Entropy classifier trained on contextual features. *NERA1* needs the input files to be pre-processed, and, in order to do this, it calls the *TTL*web service.[7] However, as Romanian is a language with diacritics and many Romanian texts are missing these diacritics, when dealing with it, *NERA1* is able to call the *diacritics insertion* web service, if requested.

Important facts:

- *NERA1* can receive as input raw text files with no pre-processing. In this case, the *TTL* web service is called for pre-processing and an internet connection is needed.
- *NERA1* can work on existing annotated data if the already existing annotation is compliant with RACAI's *XML* resource format.

[7] http://www.racai.ro/en/tools/text/

8.4.2 Tools for Terminology Extraction

This section covers the tools that perform terminology extraction and tools that are created to integrate out of ACCURAT project developed tools within the toolkit's general use case workflows. The tools included in this section of the ACCURAT toolkit are

- *CollTerm*—a tool for term extraction
- *Tilde's wrapper system for CollTerm*
- *KEA wrapper*—a wrapper system for the external tool KEA
- *Terminology Extraction tool for English and Romanian.*

8.4.2.1 CollTerm—A Tool for Term Extraction

CollTerm is a tool for collocation and term extraction, that is extracting word sequences that co-occur more than by chance or that occur significantly more frequently in a domain corpus than in a reference corpus. This tool extracts collocation and term candidates by applying POS/MSD phrase filters and stop-word filters and computing different statistical association measures between sequences of words. If an *IDF* list file is present, the tool takes into account the significance of the term frequency regarding a reference corpus. The output of the tool is a list of collocation and term candidates ranked by their strength.

The scoring of the *n*-grams (starting from bigrams) that pass the POS/MSD filters and stop-word filters is performed by five different association measures. Association measures, loosely speaking, measure how much words in a sequence of words co-occur more than by chance.

The five association measures implemented in this tool are the following:

- *Dice coefficient:*

$$\mathrm{DICE}(w_1 \ldots w_n) = \frac{nf(w_1 \ldots w_n)}{\sum_{i=1}^{n} f(w_i)},$$

where $f(.)$ is the frequency of a specific *n*-gram.

- *Modified pointwise mutual information:*

$$I'(w_1 \ldots w_n) = \log_2 \frac{f(w_1 \ldots w_n) P(w_1 \ldots w_n)}{\prod_{i=1}^{n} P(w_i)},$$

where $f(.)$ is the frequency of a specific n-gram and $P(.)$ is the probability of an n-gram calculated as a maximum likelihood estimate.

- *Chi-square statistic:*

$$\chi^2 = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}},$$

where O_{ij} and E_{ij} are observed and expected frequencies in a contingency table of two dimensions for bigrams (contingency tables for n-grams have n dimensions).

- *Log-likelihood ratio:*

$$G^2 = 2 \sum_{i,j} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

where observed and expected frequencies are calculated as in the chi-square statistic.

- *T-score statistic:*

$$t\text{-score} = \frac{O_{11} - E_{11}}{\sqrt{E_{11}}}$$

where observed and expected frequencies are calculated as in the chi-square statistic and the log-likelihood ratio.

These association measures have been selected from an exhaustive list of existing association measures, since previous research for bigrams (Evert 2005; Pecina 2009) and n-grams (Petrović et al. 2010) has shown that these measures show the most consistent results on different datasets and languages.

8.4.2.2 Tilde's Wrapper System for CollTerm

Tilde's wrapper system for CollTerm provides functionality for term tagging in plaintext documents, pre-processing of term-annotated documents and evaluation of *CollTerm* results for a given test corpus. As *CollTerm* requires pre-processed data, the wrapper provides all required pre-processing scripts.

The wrapper system has been created also in order to support varied length term extraction using *CollTerm*. As *CollTerm* supports only fixed length (from one to four tokens) n-gram extraction, the wrapper system executes *CollTerm* multiple times and combines the results in one output data file for each input document.

8.4.2.3 KEA Wrapper

In the multi-lingual NE and term mapper, we make use of *KEA* to tag terms for the English documents. The system is an existing tool and is not implemented within the ACCURAT project. The output of this system is, therefore, different from the input format of the NE and term mapper. The wrapper

- Enables that the output of *KEA* is of the same format as the input files to the mapper
- Provides a scenario to users where the mapper can be run on existing annotated data
- Enables the users to use other TE systems to prepare the input to the mapper.

8.4.2.4 Terminology Extraction for English and Romanian

The *Terminology Extraction* (TE) tool is designed to identify mono- and multi-word terminological terms in raw texts. It is designed to work for English and Romanian. In order for it to work, the application needs the input files to be pre-processed. To do so, it calls for the *TTL* web service (hosted at RACAI, *WSDL* file at http://ws.racai.ro/ttlws.wsdl). You can find more information about the technology it implements in Sect. 4.2.3. In order to properly work, the application needs that all the files given in the input file be part of the same domain, as it takes them all into account for having enough statistical relevance for computing the probabilities of various words and expressions of being terms.

8.4.3 Tools for Named Entity and Terminology Mapping

This section covers the tools that perform multi-lingual named entity and terminology mapping and are created within the ACCURAT project. The tools included in this section of the ACCURAT toolkit are

- *Multi-lingual named entity and terminology mapper*
- *NERA2*—language-independent named entities mapper
- *Terminology Aligner*—language-independent terminology mapper
- *P2G*—A tool to extract term candidates from aligned phrases.

8.4.3.1 Multi-lingual Named Entity and Terminology Mapper

We implemented a multi-lingual language-independent application (*MapperUSFD*) which aims to map named entities and technical terms in reports written in different languages to each other.

For named entity (NE) mapping, we implemented two scenarios. In the first scenario, the *NE mapper* takes as input two comparable documents in text format and outputs pairs of NEs with scores indicating their level of mapping. On both sides, we use *OpenNLP* (http://incubator.apache.org/opennlp/) to identify sentence boundaries. Next, on the English text, the mapper applies *OpenNLP NER* to extract English NEs. On the foreign text, it uses case information to identify candidates as foreign NEs. It treats all capitalised words as NEs and uses them for comparison with the English NEs. Consecutive capitalised words are treated as a single NE. For each word in the beginning of each sentence, we compare its lowercase variant with a list of lowercase words. If the lowercase variant is found in the list, then it is not treated as an NE. After having collected NEs in English and so-called NEs in the foreign language, we compare each English NE with all the other foreign NEs. The comparison is computed using cognate-based methods.

In the second scenario, the mapper uses proper NE identification on both sides. On the English side, it continues using the *OpenNLP NER*. On the foreign text side, it assumes that the NEs are identified using the NER systems described in previous sections. Having both lists of NEs with their types (*PERSON*, *LOCATION*, *ORGANIZATION*), it uses cognate-based methods to align them. However, instead of comparing every English NE with every foreign NE, it compares every English NE with type *X* with every foreign NE of the same type. We use cognate methods for the comparison.

The same cognate based approach as in NE mapping is applied to align terminologies. For English term extraction, the *KEA TE* extractor can be used (http://www.nzdl.org/Kea/). On the target one, the ACCURAT-specific tools are used. Extracted terminologies from both sides are aligned using cognate-based methods.

8.4.3.2 NERA2: Language-Independent Named Entity Mapper

The *Named Entity Aligner* (*NERA2*) tool is designed to map the named entities extracted from comparable or parallel documents. The algorithm is language independent, and the application is intended to work for any pair of languages as long as a translation equivalents table exists for that pair of languages for occurrence forms. As input, the application needs the corresponding documents (comparable or parallel) with named entities marked according to *MUC-7* style. The input of *NERA2* is perfectly compatible with *NERA1* output.

8.4.3.3 A Language-Independent Terminology Aligner

The *Terminology Aligner* tool is designed to map the terms extracted from comparable or parallel documents. The algorithm is language independent, and the application is intended to work for any pair of languages as long as a translation equivalents table exists for that pair of languages for occurrence forms. As input, the application needs the corresponding documents (comparable or parallel) with

terminology marked according to *MUC-7* style. The input of the *Terminology Aligner* is perfectly compatible with the output of *Terminology Extraction*. Methods applied in the *Terminology Aligner* have been describer in Pinnis et al. (2012b).

8.4.3.4 P2G: A Tool to Extract Term Candidates from Aligned Phrases

P2G (*PhraseTable2Glossary*) is a tool that extracts well-formed term candidates from phrase-aligned data, be it phrase tables or other outputs of phrase alignment (like *AnymAlign*, *PEXACC*, etc.).

The principal approach is to apply a series of filters to the input candidate phrases in order to output only the ones that can really be terms. Term candidates are brought into the right shape (lemmatisation, true-casing, gender and number agreement (in case of multi-words), etc.)

- First, the tool creates a lattice of *<lemma, POS>* pairs for each word of the input candidate, using a lemmatiser (and decomposer for German).
- This lattice is then compared to a filter of (single and multi-word) structures which allows only sequences having a 'legal' term structure to pass.
- This is done both for source and target candidates.
- In case of success, a proper term entry is created by lemmatising the head of the term into singular form, by true-casing all its parts (capitalising nouns, upper-casing acronyms, etc.), and by creating proper agreements between head nouns and modifying adjectives (using a noun gender–defaulting mechanism).
- Finally, a filter can be applied to filter out term candidates that are already known (e.g. from a general-purpose lexicon, stop words, etc.) and only the rest is output (not in the current version).

Tests have shown that in the best case (using *MOSES*-aligned data), the overall error rate of the *P2G* tool is about 5% (2–3% each coming from errors in German or English pattern extraction or term creation) and an additional 6% of errors result from incorrect phrase alignments by *MOSES*, so the overall error rate is about 11%. This is considered sufficient for human post-editing. The speed is about 100K phrase table entries per second, with about every 500th phrase table entry containing a well-formed term (in the automotive test, *P2G* created about 15.7K terms from a 6.9 million phrase table in 65 seconds).

8.5 Sisyphos-II: MT-Evaluation Tools

This is a set of tools for interactive[8] MT output evaluation. It supports the main non-automatic evaluation metrics used today which are

[8]The first version of Sisyphus was created by the Belgian METAL team in 1987, in pre-Windows times, to speed up system development. This kind of tool is still needed.

- Determination of the quality of an MT output, in terms of adequacy and fluency (called '*absolute evaluation*'). This answers the question: '*How good is the MT output?*'
- Determination of the quality of an MT output in comparison to another MT output (called '*comparative evaluation*'). It answers the question '*Which output (of two systems) is better?*' Note that it does not answer the question about the real quality of output.
- Determination of the distance of an MT output to a correct human translation (called '*post-editing evaluation*'). It answers the question about the effort needed to create a good translation from raw MT output, both in terms of edit distance and required post-editing time.

Three little stand-alone tools have been created to support these evaluations. They can be given to external evaluators (e.g. freelancers) together with a pack of evaluation data so that evaluators can process them offline and return the results. The main functionalities of the tools are

- Import of a new evaluation 'package'
- Interactive support of the evaluation procedure
- Creation of result files containing statistics.

The data flow is depicted in Fig. 8.10. The main files are the translation and evaluation XML files. Each tool works with two XML files, called '*translation-{abs|comp|post}.xml*' (created by the import function from the source and target language files produced by the MT systems) which stores the data to be evaluated, and '*evaluation-{abs|comp|post}.xml*' (created during interactive evaluation) which stores the evaluation result. The result of the evaluation is stored in the evaluation XML files; an overview file can be created containing basic statistics.

Absolute Evaluation For a given translation, its quality is determined. The translation is displayed, and users can evaluate the adequacy and the fluency of the translation. Each time a 4-point scale is presented, users select one of the options in both areas

- For adequacy, the options are {full content conveyed | major content conveyed | some parts conveyed | incomprehensible}.
- For fluency, the options are {grammatical | mainly fluent | mainly non-fluent | rubble}.

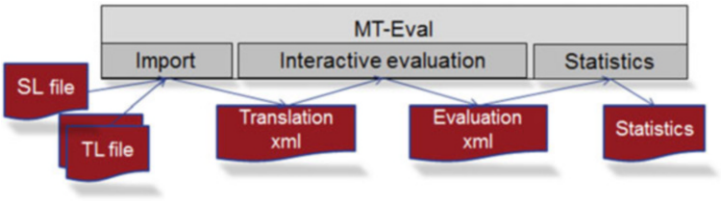


Fig. 8.10 Data flow

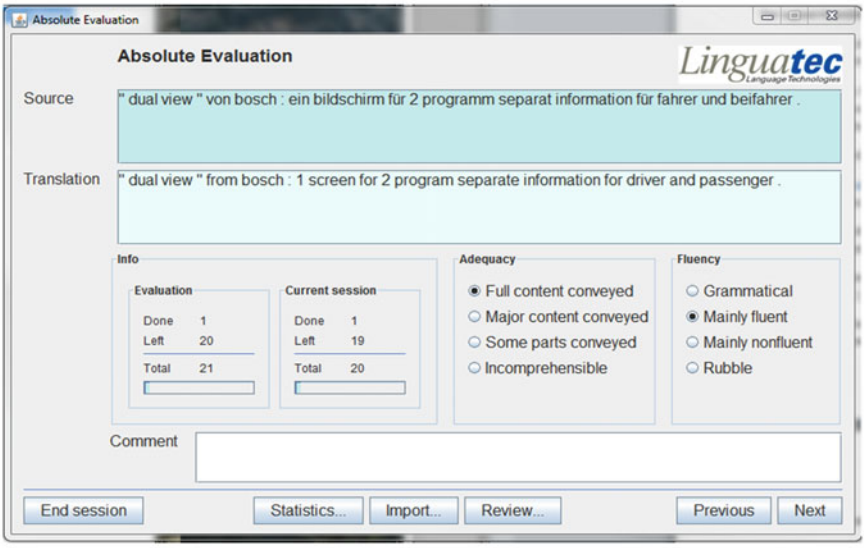


Fig. 8.11 Absolute evaluation

The result is stored by clicking on '*Next*', and the next sentence is presented. '*Previous*' displays previous evaluation data for corrections. The absolute evaluation interface is shown in Fig. 8.11.

Comparative Evaluation This tool compares the quality of two translations against each other. Two translations of a given sentence are displayed for comparison. Users can decide which one is better, on a 4-point scale.

Comparison options are {first translation better | both equally good | both equally bad | second translation better}.

The sequence of *translation1* and *translation2* is randomised to avoid biased evaluation (i.e. *translation1* is sometimes displayed first, sometimes second).

The result is stored by clicking on '*Next*', and the next sentence is presented. '*Previous*' displays previous evaluation data for corrections. The comparative evaluation interface is shown in Fig. 8.12.

Post-Editing Evaluation This tool measures the time needed to post-edit a translation output into a correct format (HTER—human-targeted translation edit rate). Afterwards, it can also be used to compute the edit distance. The translation of the source sentence is displayed. The translation field is editable, so users can edit the MT output.

The time from the first display of the sentence until the pressing of the '*Save*' button is stored (in seconds). There is also a '*comment*' field which can be used to give comments on the translation/post-editing. Navigation is done with the '*Next*' and '*Previous*' buttons. The post-editing evaluation interface is shown in Fig. 8.13.

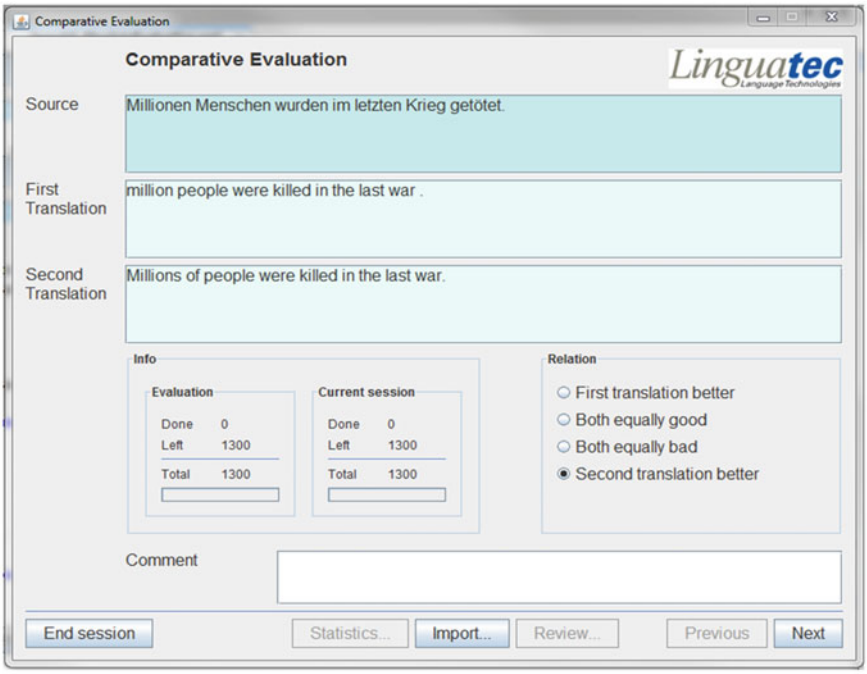


Fig. 8.12 Comparative evaluation

Evaluation Users have the option to see an overview of the evaluation at any time of their work. They can click on '*Statistics*', and then a statistics on the number of sentences and how they were evaluated is shown. Users can print this into a file. For more detailed evaluation, the evaluation XML files used by the tools must be consulted, like for inter-annotator agreement or for edit-distance computation. Examples of the evaluation files are given in Fig. 8.14 (for easier processing, all XML mark-ups are in separate lines).

From this XML file, the interesting data can be extracted:

- For Kappa calculation—sentence IDs, evaluator, evaluation results
- For edit distance calculation—translated text, post-edited text, etc.

Users should save the evaluation XML files separately from the working directory of the *MT-Eval tools*, in order to protect them from being overwritten by the next evaluation task.

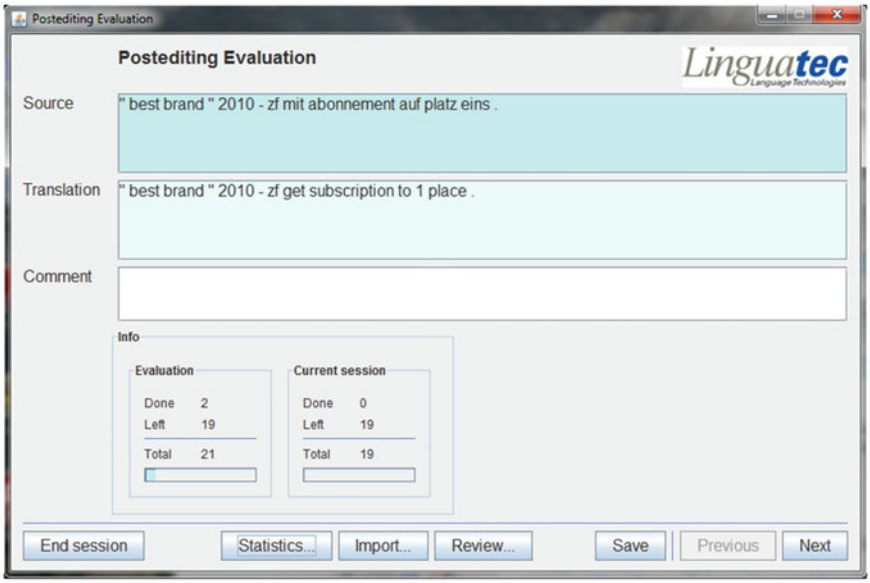


Fig. 8.13 Post-editing evaluation

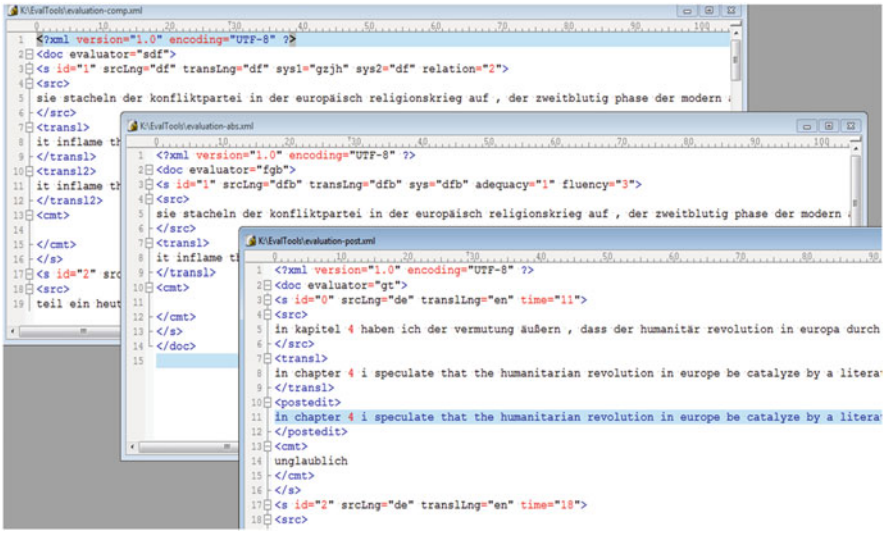


Fig. 8.14 Examples of evaluation files

8.6 Conclusions and Related Information

In this section, we described the ACCURAT toolkit containing tools for bi-/multilingual comparable corpora collection, multi-level alignment and information extraction from comparable corpora. In addition, tools for MT evaluation are presented. These tools are integrated into pre-defined workflows that are ready for immediate use. The workflows provide functionality for corpora collection and extraction of semi-parallel sentences, bilingual NE dictionaries and bilingual term dictionaries from comparable corpora.

The methods, including comparability metrics, parallel sentence extraction and named entity/term mapping, are language independent. However, they may require language-dependent resources, for instance, POS-taggers, Giza++ translation dictionaries, NERs, term taggers, etc.[9]

The ACCURAT toolkit is released under the Apache 2.0 licence and is freely available for download after completing a registration form.[10]

References

ACCURAT D2.6. (2012). *Toolkit for multi-level alignment and information extraction from comparable corpora*. http://www.accurat-project.eu

Adafre, S. F., & de Rijke, M. (2006). Finding similar sentences across multiple languages in Wikipedia. *Proceedings of the EACL Workshop on New Text*, Trento, Italy.

Baroni, M., & Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004* (pp. 1313–1316).

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics, 19*(2), 263–311.

Evert, S. (2005). *The statistics of word cooccurrences: Word pairs and collocations*. PhD thesis, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Ion, R., Ceauşu, A., & Irimia, E. (2011). An expectation maximization algorithm for textual unit alignment. *Proceedings of the 4th Workshop on Building and Using Comparable Corpora (BUCC 2011)* held at the *49th Annual Meeting of the Association for Computational Linguistics* (pp. 128—135), Portland, OR, June 24th, 2011. (C) 2011 Association for Computational Linguistics. ISBN: 978-1-937284-01-5.

Ion, R. (2012). PEXACC: A parallel data mining algorithm from comparable corpora. *Proceedings of LREC 2012,* May 21–27, Istanbul, Turkey.

Pecina, P. (2009). *Lexical association measures: Collocation extraction. Studies in computational and theoretical linguistics*. Prague, Czech Republic: Institute of Formal and Applied Linguistics.

Petrović, S., Šnajder, J., & Bašić, B. D. (2010). Extending lexical association measures for collocation extraction. *Computer Speech and Language, 24*(2), 383–394.

[9]Full requirements are defined in the documentation of each tool ACCURAT D2.6 (2012).

[10]http://www.accurat-project.eu/index.php?p=toolkit

Pinnis, M. (2012). Latvian and Lithuanian named entity recognition with TildeNER. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey.

Pinnis, M., Ion, R., Ştefănescu, D., Su, F., Skadiņa, I., Vasiļjevs, A., et al. (2012a). Toolkit for multi-level alignment and information extraction from comparable corpora. *Proceedings of ACL 2012, System Demonstrations Track*, Jeju Island, Republic of Korea, July 8–14, 2012.

Pinnis, M., Ljubešić, N., Ştefănescu, D., Skadiņa, I., Tadić, M., Gornostay, T. (2012b). Term extraction, tagging, and mapping tools for under-resourced languages. *Proceedings of the 10th Conference on Terminology and Knowledge Engineering (TKE 2012)*, June 20–21, Madrid, Spain.

Skadiņa, I., Aker, A., Giouli, V., Tufiş, D., Gaizauskas, R., Mieriņa, M., et al. (2010). Collection of comparable corpora for under-resourced languages. *Proceedings of the Fourth International Conference Baltic HLT 2010*, *Frontiers in Artificial Intelligence and Applications* (Vol. 219, pp. 161–168). IOS Press.

Ştefănescu, D. (2012). Mining for term translations in comparable corpora. *Proceedings of the 5th Workshop on Building and Using Comparable Corpora (BUCC 2012)* to be held at the 8th edition of *Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May 23–25, 2012.

Ştefănescu, D., Ion, R., & Hunsicker, S. (2012). Hybrid parallel sentence mining from comparable corpora. *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT 2012)*, Trento, Italy.

Su, F., & Babych, B. (2012a). Development and application of a cross-language document comparability metric. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey.

Su, F., & Babych, B. (2012b). Measuring comparability of documents in non-parallel corpora for efficient extraction of (semi-) parallel translation equivalents. *Proceedings of EACL'12 Joint Workshop on Exploiting Synergies Between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, Avignon, France.