Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 285 – 292
ISSN 1896-7094
© 2007 PIPS

Collecting Polish-German Parallel Corpora in the Internet

Monika Rosińska

Adam Mickiewicz University in Poznań
monika.rosinska@gmail.com

Abstract. Parallel corpora have recently become indispensable resources in multilingual natural language processing. Manual preparation of a bilingual corpus is a laborious task. Therefore methods for the automated creation of parallel corpora are currently a topic of concern for many researches. A number of sophisticated and effective algorithms for collecting parallel texts from the Internet have already been created. The aim of the research has been to verify the efficiency of existing algorithms for the collection of Polish-German parallel corpora, intended as a reference source for a Machine Translation system, and possibly, to propose a new algorithm – best suitable for the task.

1 Introduction

Parallel corpora – sets of corresponding texts in two or more languages – seem to be an indispensable resource for widely comprehended natural language processing. Melamed (2000) and Brown (1990) have proved the need for parallel corpora as an essential resource for the preparation of language statistical models in statistical machine translation. Gale and Church (1991) showed the importance of parallel corpora in automatic lexical acquisition.

Unfortunately, preparation of large high-quality corpora is not an easy task. "Handmade" corpora are usually limited in size, very expensive in preparation and not as representative as they should be. That is why most researches attempt to develop efficient and fully automated mechanisms that could yield demanded parallel corpora in relatively short time.

One of the problems that researches have come across is the appropriate choice of source of texts. Developers of latest mechanisms have chosen Internet as such a source because of its vast size, dynamic nature (the content of websites is continually updated), availability and multilinguality.

The most popular algorithms that use the Web as a repository of bilingual texts are: STRAND (Resnik 1999, Resnik and Smith 2003), BITS (Ma and Liberman 1999) and

PTMiner (Chen and Nie 2000). The authors of the algorithms have published results for various pairs of languages – however none of the tests included the Polish-German pair.

In our experiment we wanted to learn how existing algorithms would deal with the Polish-German pair and what innovations might be done to make the algorithms more efficient for this specific pair.

2 Definitions

2.1 Definition of the Corpus

The most general definition of a text corpus – "corpus is a set of texts" – seems not detailed enough for language researches. They are of the opinion that not every set of texts might be regarded as a corpus. That is why a more descriptive definition has to be specified by listing distinctive features of a text corpus.

Some of the criteria proposed by Tony McEnery and Andrew Wilson (1996) are the following:

- representativeness
- finite size
- machine-readable form

On the other hand, distinctive features of corpora considered by John Sinclair (1991) as the most important are:

- high quality
- infinite size
- detailed documentation.

2.2 Types of Corpus

We can distinguish various types of text corpus, depending on their structure, origin, source of texts, types of texts (specialized or general), purpose of creation. The most useful types for machine translation are:

Parallel corpus.

A parallel corpus is a set of texts in two or more languages that meets a criterion of text linkage (this feature distinguishes them from the collection of monolingual corpora in a few languages). Texts must be linked to one another on a defined level: document, paragraph, sentence phrase or word. That means that each part of a text (a document, a paragraph, a sentence a phrase or a word, depending on the level of linkage), in language A must correspond to an appropriate part of text in language B.

Indexed corpus.

An indexed corpus is a set of texts that contains additional data (called metadata) that stores information about the content of the corpus. The information may concern linguistic categories of individual words, the origin of the text or text formatting. The most popular standard for the annotation of parallel corpus is TMX (Translation Memory eXchange format `http://www.lisa.org/tmx/`).

2.3 Web Crawler

Crawler, called also a robot or an Internet spider, is a program that crawls through the Internet in search of reference materials on websites. Its main task is to gather information about visited websites and to download specified content of websites.

3 Parallel Corpus Finding Algorithms

The creation of a bilingual corpus consists of a few stages:

1. Determine required attributes of the corpus (languages, source of texts, typeof formalism, etc.). Those attributes significantly influence consecutive stages of the algorithm.
2. Collect pairs of texts that seem to be mutual translations in earlier determined languages.
3. Clean up collected texts by removing unnecessary information.
4. Align texts, i.e. link parts of text on the defined level.
5. Add crucial metadata and wrap up the texts into the right formalism.

The most difficult steps are 2 and 4. Here, we will discuss step 2 – collecting parallel texts. The first problem that has to be solved is to choose an appropriate source of texts. There are many possibilities (magazines that are translated into various languages, official European Union documents, etc.) but it has been verified by P. Resnik (1999), M. Liberman, X. Ma (1999) and many others, that the most promising source for obtaining appropriate texts is the Internet. In addition to other virtues (such as mentioned above: vast size, dynamic nature and multilinguality) texts in the Internet are easily available and have a computer-readable format. Thanks to computer-readable format the whole process of corpus creation can be made fully automatically.

The most commonly used algorithms for research purposes are STRAND (Resnik and Smith, 2003), BITS(Ma and Liberman, 1999) and PTMiner (Chen and Nie, 2000).

3.1 STRAND

STRAND is the algorithm created by Philipp Resnik. It is based on the assumption that corresponding texts have alike URL addresses and similar web page layouts (HTML tags architecture). A draft of STRAND (version from 1999) is presented below.

1. Use search engine (like AltaVista or Google) to find websites that might be parallel and in desired languages.
2. Verify language of each website
3. Look for candidate pairs in the set of collected websites: candidate pairs should have similar URLs (differing from each other only with the language marker, for example `http://google.com` and `http://google.de`), should be of similar sizes and should be written in different languages.
4. Verify the candidate pair – check if the HTML structure of both websites is identical.

STRAND (proposed in 1999) achieved very promising results while searching for English-French pairs (98% recall (The proportion of relevant documents that are retrieved, out of all relevant documents available.) and 97.4% precision (The proportion of retrieved and relevant documents to all the documents retrieved.)).
In 2003 Resnik came to the conclusion that his assumptions are too general and enriched STRAND with content-based matching functions (the source code is unpublished yet). Moreover, STRAND has now its own mechanism for candidate websites searching (crawler). Those changes allow better results for other pairs of languages achieving.

3.2 BITS

The BITS algorithm has been developed by Xiaoy Ma and Mark Liberman in 1999. The mechanism used for collecting candidate websites was relatively simple. The whole .de domain has been downloaded and for each web-site the algorithm tried to find a parallel page. A draft of BITS is presented below.

1. Download the whole **.de** domain.
2. Generate list of websites that are possibly bilingual.
3. For each element on the list verify if it has a likely correspondent in another language, if so, download the whole content of the correspondent.
4. Clean up collected texts by removing unnecessary information.
5. Create two lists: L1, containing only texts in the source language (German) and L2, containing texts in the target language,
6. For each element in L1 find the most similar element in L2 – using Levenshtein distance (Levenshtein, 1965) or LikeIt distance (Yianilos and Kanzelberger, 1997) .

The main difference between BITS and STRAND is that BITS uses more complex methods to define the parallelism between texts in two languages. The most efficient one uses the LikeIt distance – it is a refined form of the edit distance applied to determine the similarity between two texts (rather than words). Ma and Liberman have used BITS to prepare an English-German corpus. The size of created corpus was 63 MB. The results achieved by BITS were very good – 97.1% recall and 99.1% precision.

4 Research Description

4.1 The Aim of Research

The aim of the research has been to verify the efficiency of existing algorithms for the collection of Polish-German parallel corpora, intended as a reference source for a Machine Translation system, and possibly, to propose a new algorithm – best suitable for the task.

At the beginning we implemented an algorithm based on STRAND. The achieved results were not satisfying due to various factors:

- German translations of Polish web pages were usually of poor quality,
- the addresses of bilingual services seldom followed any naming convention,
- web pages were highly noised.

Therefore, in the second step we enriched the algorithm with some linguistic methods inspired by BITS. The results were better but the algorithm still needs improvements.

4.2 The Algorithms

During the research two specialized algorithms have been implemented. Both implementations have the same module architecture (the modules are presented in 4.2.1, 4.2.2, 4.2.3). However, they use different methods for finding pages and verifying the similarity of candidates.

The main steps in both implementations are following:

1. Create a set of starting URLs.
2. Crawl recursively through the Internet and find as many links as possible.
3. Add found links to the database.
4. For each link define its web page language and size.
5. Find candidate pairs in the database.
6. Verify similarity of candidate pages.
7. Clean up verified pages and add them to the corpora.

4.2.1 Parallel Pages Finder

In the first algorithm we followed the STRAND assumption: the URL addresses of parallel websites should differ with each other only on language suffix (that is why the Levenshtein distance (Levenshtein, 1965) between parallel URLs should not exceed 5). Unfortunately, in case of Polish-German web pages we observe a lack of a naming convention for parallel web pages. Only a part of parallel web-pages meet this criterion. Therefore in the second implementation the crawler looks for Polish and German pages that meet the only criterion of similar size. Their parallelness is verified in next stages.

4.2.2 Language Recognizer

LanguageRecognizer is the module that determines the language of a page. The method we have implemented consists of two main steps:

In the first step we use heuristic methods based on occurrence of national characters. For example, if one of the characters: ą, ć, ę, ł, ś appears in the text, it is assumed that the text is in Polish.

The second step of recognition consults a statistical language model, prepared beforehand. The model consists of trigrams of letters, and their probabilities of occurrence in texts of the modeled language.

4.2.3 Similarity Verifier

Firstly we have implemented a method based on the Resnik assumption that parallel web pages have similar HTML tags structure. Unfortunately, the achieved results were not satisfying. Inspired by methods used in BITS we tried to implement some content based methods to verify if candidate web pages are really equivalent.

One of the ideas that we implemented was the analysis of numbers. We verified whether the same numbers appear in the same order in the candidate pages.

In future we want to improve this method by dividing web page content into smaller pieces of text and then stemming and using dictionary to find out if those content are parallel.

5 Results

Firstly, we developed a crawler and used STRAND to filter candidates for parallel texts. The crawler has visited 9225 websites and has found there 521664 links. 46422 were marked as parallel ones. The results were not satisfactory – only about 10% found candidates were indeed parallel pages.

Therefore we decided to combine STRAND with methods inspired by the LikeIt distance applied in BITS (a source code of LikeIt has not been published). The

crawler visited 3583 websites and found 290264 links 104496 of them were recognized as parallel. This improved results significantly – the accuracy increased to about 40%. Unfortunately the speed of the improved crawler was much lower due to more complex methods that verify the similarity of candidate web pages.
Still, the results are clearly worse than those achieved by Resnik using pure STRAND for English and Basque or English and French.

6 Conclusion and Future Plans

The experiment draws us to the following conclusions:

- The accuracy of existing algorithms used for language pairs like Polish and German, where neither of languages is leading in the Internet is much lower than those obtained if one of the languages is English.
- Collection of Polish-German parallel corpora is hindered by the information noise appearing in German translations of Polish texts. The information noise has strong impact on all of linguistic methods used in the algorithm. The methods must be well adjusted to overcome the problem.
- The starting web links strongly influence the efficiency of a crawler.

According to these conclusions we intend to improve the similarity verifier and develop new methods for choosing the appropriate set of starting URLs for the crawler. The similarity methods will be enhanced by some latest metrics used to strings comparison and dictionary methods.

References

1. Brown P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, and P. Roossin. 1990. *A statistical approach to machine translation,* Computational Linguistics, 16(2):79-85.
2. Melamed I. Dan. 2000. *Models of translational equivalence among words,* Computational Linguistics, 26(2):221-249, June.
3. Gale William A. and Kenneth W. Church. 1991. *Identifying word correspondences in parallel texts,* In Fourth DARPA Workshop on Speech and Natural Language, Asilomar, California, February.
4. Resnik Philip. 1998. *Parallel strands: A preliminary investigation into mining the Web for bilingual text,* In Proceedings of the Third Conference of the Association for Machine Translation in the Americas, AMTA-98, in Lecture Notes in Artificial Intelligence, 1529, Langhorne, PA, October 28-31.
5. Ma, Xiaoyi and Mark Liberman. 1999. *Bits: A method for bilingual text search over the web,* In Machine Translation Summit VII, September. `http://www.ldc.upenn.edu/Papers/MTSVII1999/BITS.ps`

6. Chen Jiang and Jian-Yun Nie. 2000. *Web parallel text mining for Chinese English cross-language information retrieval,* In International Conference on Chinese Language Computing, Chicago, Illinois.
7. Levenshtein V. I.: *Binary codes capable of correcting deletions, insertions and reversals,* Doklady Akademii Nauk SSSR, 163 (No. 4, 1965).
8. Peter N. Yianilos, Kirk G. Kanzelberger *The LikeIt intelligent string comparison facility,* Technical Report, NEC Research Institute, May 1997.