Karlsruhe Institute of Technology

Getting Bilingual Information from the Web

Bachelor Thesis of

Michael Koch

At the Department of Informatics
Institute for Anthropomatics (IFA)

Advisor: Prof. Dr. Alex Waibel

Duration:: January 15th 2012 – May 14th 2012

0.1 Zusammenfassung

Folgende Ergebnisse sind im Rahmen einer Bachelor-Abschlussarbeit entstanden. Es wird der grundsätzlichen Frage nachgegangen, wie das Web genutzt werden kann, um zweisprachiges Textmaterial zu finden. Zwei Fälle werden betrachtet: Erstens, Suchen allgemein nach zweisprachigen, nicht unbedingt parallelen Material und zweitens, Suchen nach einer bestimmten Übersetzung für einen ganz bestimmten Ausdruck oder Wort.

Im ersten Fall wird nach Textmaterial gesucht, welches einer bestimmten Domäne angehört. In diesem Falle wird nach Forschungs- und Wissenschaftstexten gesucht. Sinn ist es einen allgemeinen Wortschatz innerhalb dieser Domäne zu gewinnen.
Benötigt sind Webseiten, welche Artikel in mehreren Sprachen anbieten. Solche Artikel können mit Hilfe von Syntaxinformationen wie auch Semantikinformationen gesucht werden. Syntaxinformationen sind Hinweise, dass weitere Webseiten mit einer entsprechenden Übersetzung existieren. Diese Informationen können zum Einen aus der URL der entsprechenden Webseiten gewonnen werden. Eine andere Quelle sind bestimmte Verlinkungen auf der Webseite . Diese Verlinkungen haben ein Wort oder Text, welcher dem Betrachter der Webseite signalisiert, dass die Verlinkung zu einer Übersetzung in eine bestimmten Sprache führt. Semantikinformationen werden durch teilweise Übersetzung des Artikel gewonnen. Webseiten, die viele Teile der Übersetzung beinhalten, können als mögliche Übersetzung angesehen werden. In dieser Arbeit wurde nur mit Syntaxinformationen gearbeitet.
Zuerst werden Webseiten diverser Universitäten aus Deutschland und Österreich durchsucht, um gleiche Artikel in Deutsch und Englisch zu finden. Die Artikel werden aus den Webseiten extrahiert, in Sätzen unterteilt und dann im Kreuzprodukt-Verfahren werden die Sätze zweier Artikel sich gegenseitig zugewiesen. Diese Satzpaare werden einem trainierten Support-Vector-Machine Klassifizierer gegeben, welcher die vermeintlichen Übersetzungspaare in korrekte und nicht korrekte Übersetzungen einteilt.
Die in dieser Arbeit vorgestellten Untersuchungen haben ergeben, dass Webseiten von Universitäten eine gute Quelle darstellen, um Übersetzungen im Bereich Forschung und Wissenschaft zu finden. Um intensiv nach solchen Artikeln zu suchen, benötigt es Methoden für Datenbankanbindung und Verfahren , um anfallende Probleme im Netzwerkbereich zu lösen.

Im zweiten Fall wird gezielt nach einer Übersetzung gesucht. Übersetzungen für bestimmte Worte können von Übersetzungsannotationen bestimmter Webseiten als auch über Kontext verwandte Wörter gefunden werden. Im letzteren Fall wird dann ein Lexikon benötigt, um verwandte Wörter vom Deutschen ins Englische zu übersetzen.
Statistische Methoden helfen, um Kontext relevante Wörter zu bestimmen. Als monolingualer Korpus (sowohl Deutsch als auch Englisch) dient das Web, welches mit der Bing API von Microsoft durchsucht wird. Die Websuchmaschine Bing API wird auch dazu benutzt, um Frequenzen von Wörtern und Wortpaaren im Web zu ermitteln. Diese Frequenzen werden zur Berechnung von Korelationsfaktoren benötigt.
Die durchgeführten Tests haben ergeben, dass das Finden von Kontext relevanten Wörtern mit einfachen Mitteln durchaus möglich ist, auch wenn es Verbesserungspotential gibt. Schwierig ist es von Kontext relevanten Wörtern auf die eigentlich gesuchte Übersetzung zu schließen. In dieser Arbeit angewandten Methodiken lieferten Ergebnisse, die eine verlässliche Benutzung leider nicht zulassen.

0.2 Abstract

This work has been created in the context of a bachelor thesis. The issue to investigate is which resources are offered by the Web in order to gather bilingual material which can be used for Statistical Machine Translation. There are two situations: In which way is it possible to find bilingual web pages to get a large corpus. The second one is whether and how well translations for specific words can be found within the Web
Finding bilingual content about a specific topic or of a specific domain helps to create a corpus and a dictionary. In this thesis, the domain is research and science. The aim is to obtain a set of common vocabulary for translating lectures hold at a university or college. To do so, web sites of German and Austrian universities are crawled to find bilingual web pages within. Text will be extracted, split up into sentences and aligned to sentences of the corresponding web page. To classify good translations and bad translations, a Support-Vector-Machine classifier will be applied.
Finding translations for specific expressions has been showed to be more complicated. The basic idea is to find related words from which the wanted translation can be concluded. In order to find related words, several co-occurrence measurements will be applied. To receive contexts of the specific word, a Web search engine (Bing) is been used. Also the Web search engine is used to receive parameters for calculating the co-occurrence scores.

Contents

0.1 Zusammenfassung . . . iii
0.2 Abstract . . . iv

1 Introduction **1**

2 Bilingual corpus **3**
2.1 Motivation . . . 4
2.2 Related work . . . 5
2.2.1 Syntax based searching . . . 5
2.2.1.1 URL (STRAND) . . . 5
2.2.1.2 DOM (Shi et al.) . . . 6
2.2.1.3 Anchor elements (Nie et al.) . . . 7
2.2.2 Semantic based searching . . . 7
2.2.2.1 Translation comparison (Ma et al.) . . . 7
2.2.2.2 Querying translations (Munteanu et al.) . . . 8
2.3 Methodology . . . 10
2.3.1 Precollecting based on URL . . . 10
2.3.2 Precollecting based on page intern references . . . 12
2.3.2.1 Crawling . . . 12
2.3.2.2 Resolving . . . 14
2.3.3 Text segment aligning . . . 14
2.3.4 Getting comparable sentences . . . 17
2.4 Test . . . 18
2.4.1 Language determination . . . 18
2.4.2 Dividing text of web page into segments . . . 19
2.4.3 Alignment . . . 19
2.4.4 Search based on URL . . . 21
2.4.5 Search based on intern page references . . . 21
2.4.6 Support-Vector-Machine classifier . . . 22
2.4.7 Applying crawling results to sentence alignment and classifier . . . 22
2.5 Outlook . . . 25

3 Translation for oov **27**
3.1 Motivation . . . 28
3.2 Related work . . . 29
3.2.1 Bilingual Context . . . 29
3.2.1.1 Maeda et al. . . . 29
3.2.1.2 Cheng et al. . . . 30
3.2.1.3 Zhang et al. . . . 31
3.2.1.4 Huang et al. . . . 32
3.2.2 Context deduction . . . 33
3.2.2.1 Rapp . . . 33

3.3 Methodology . . . 35
3.3.1 Extracting text out of documents . . . 35
3.3.2 Out-Of-Vocabulary Detection . . . 36
3.3.2.1 Single word query . . . 36
3.3.2.2 Context taking detection . . . 37
3.3.3 Co-occurrences measures . . . 37
3.3.3.1 Log Likelihood Ratio . . . 38
3.3.3.2 Mutual Information . . . 38
3.3.3.3 Modified Dice Coefficient . . . 38
3.3.3.4 χ^2 Test . . . 39
3.3.4 Finding related words . . . 39
3.3.5 Extracting translation . . . 39
3.4 Test . . . 41
3.4.1 Out-Of-Vocabulary words in documents . . . 41
3.4.2 Finding related words in German . . . 42
3.4.3 Finding translations . . . 43
3.5 Outlook . . . 45

4 Outlook **47**

5 Appendix **49**
5.1 Anchor elements crawling . . . 49

Bibliography **53**

1. Introduction

In recent years, the attention towards translation system based on machine arose. Several situations helped to bring the need of such translation systems forward. One big player is the European Union and its administration. Since there are many official and working languages in the EU, the effort of translating legislations, regulations and requests are immense. Replacing at least partly professional translators by machines would help to increase the efficiency of communication and to reduce costs. Translation system also are interesting in the context of globalization and the Internet. Even though English which has the biggest portion on the Internet, is the most spoken language in the world, there are many people which are not able to understand English but have access to the Internet. Creating a tool to translate web content into their language, it allows many people to find new information, share ideas and opinions and to get an insight into a different world.

In Machine Translation, different approaches have been developed. The most studied and most promising approach is the statistical one. It does not need the knowledge of an expert for a specific language, but tries to find rules by analyzing a corpus which is large enough. So the statistical techniques can be generalized and applied to other languages as well. Another benefit of this generalization is that only once costs and effort are made to create these statistical techniques and then costs and effort to apply it to other languages are considerably low.

A critical part for Statistical Machine Translation is to have enough information. Usually information are given by texts in a machine readable way. Acquiring such text files requires a lot of work and costs since they should be written in the languages of interest and also be nearly parallel. Two greater sources have been the European Union and the Canadian parliament. But they are quite limited to the domain of legislation and administration. Another source, the Web, has been growing steadily. It offers an incredible amount and variation of text documents.
Another advantage is that every day new documents, news, blogs, etc. are constantly added to the Web. So the Web contains the most updated and the most divergent information about widely used languages like English, German, French, Spain, etc. .
The tricky part actually is to find the wanted information on the Web. First of all the Internet is a decentralized network and neither the Internet nor the Web provides a mechanism to find a certain content within. Of course there are Web search engines which helps to find web pages, but they build up upon the Web and they are not essential for the Web to work.
Finding web pages about a specific topic has to be mastered, but finding bilingual web

pages about a specific topic makes it more difficult.
Additionally, there is another problem to overcome. Having found bilingual web pages containing the topic of interest, the information is still in a raw state. The rawness might vary strongly. Disturbances have to be removed and different preprocessing techniques have to be applied before a corpus comes out which can be used for Statistical Machine Translation. Still then, the quality of the corpus may vary because the informations are usually not from professional translators.

In the following work, the potential of the Web will be examined. Two approaches will be tested. First, a corpus should be collected focusing on topics from research and education such as Mathematics, Biology and others. The second approach is about finding a translation for a specific word.

2. Bilingual corpus

2.1 Motivation

One way to improve the translation part is to narrow down the general vocabulary as much as possible to the area for which the translation system is supposed to be used. In our case it is about research and academical education. Since the web originally was invented for the purpose of presenting and sharing scientific documents and research results, there are a lot of web sites containing information about various topics related to research and academical education. There are two problems remaining.

The first problem is how to find vocabulary which is the most general possible within the area of research and science. Just crawling the whole web from some start pages seems to be the first approach. But depending from where the crawler is starting it might take a very long time until a corpus has been collected which is large enough.
A more promising approach appears to focus on web sites of institutions which conduct research on various topics. They bring information about many topics in a small range. So crawling can be done effectively in an acceptable amount of time. Research institutes offer articles about the cutting-edge of technology, research news and the newest discoveries. Also one may find short articles about different topics. Since the web sites of a lot of research institutes like "Max-Plank Gesellschaft" or "IBM Research" offer news and update them frequently, these sites can be used to keep the corpus updated and to keep track of new technical terms and expressions.

The second problem is about to find translations. Even if web sites have been found which give texts about research and science, it is critical to find translations for these texts. International cooperation such as "IBM Research" have different offices around the world. The main page of "IBM Research" contains plenty of articles but all are in English and no translation of these articles is offered. The same can be said about the site of "IBM Deutschland". The site of "IBM Research Zurich" partly offers translation for their articles, "IBM Research Brazil" only has English pages whereas "IBM Research China" contains only Chinese articles. To sum it up, it is difficult to find web sites which offer translations for their pages. International institutions tend to have their web sites only on English since English is supposed to be understood by people with the required educations. It might be more promising to focus on national institutions inside a multilingual or non-English-speaking country. They offer news articles and informations in their native language for people from their own country. But for international people most of the information will be translated into English. National research institutions like the German "Max-Plank-Gesellschaft" or the Spanish "Consejo Superior de Investigaciones Científicas" provide information both in English as well as in German respectively Spanish.

Unfortunately such research institutions often restrict themselves to technical, natural and life science. Unless the focus for collecting a corpus mainly lies on these scientific areas, crawling sites of such institutions does not cover the entire spectrum of research and academical education. To include philosophical sciences, human sciences, etc. crawling web sites of universities appears to be a good start off. Research groups offer information about their work and almost the whole bandwidth of science will be covered. Additionally most of the information will be translated into English, especially if the university has a focus on international students or own a very good reputation world wide. Of course the translations will not be parallel because the different translation versions are intended for different target groups - local respectively international people. Hence they need sometimes slightly different informations, each version will take their need into consideration. Another reason why the chances to find parallel data will be thin, is that it takes more effort to translate as parallel as possible than just formulating the essential meaning in the target language.

2.2 Related work

The idea of using the web to collect a corpus has arrived briefly after the explosive growth of the web itself. Since then, several approaches have been carried out to exploit this source. Here we will give a short overview.

2.2.1 Syntax based searching

Searching based on syntax means that only information from the underlying HTML document or information unrelated to the specific content are taken. Here several ideas like examin the URL, comparing the DOM tree or looking out for anchor elements in the HTML documents are presented.

2.2.1.1 URL (STRAND)

Resnik and Smith proposed their project STRAND[RS03]. STRAND is an abbreviation for "structural translation recognition acquiring natural data". It is a system to find web pages which are mutual translations. The procedure to do so is based on an observation that authors tend to use the same document structure for translated web pages within the same web site. Resnik and Smith consider three steps to find parallel text.

1. Locate pages which might have parallel translations.
2. Pair web pages which might be translations of each other.
3. Filter out parallel documents.

Locating the pages is done by using the advanced search function of AltaVista. Firstly parent pages are searched for. These are pages which contain references to a document in different languages. Secondly sibling pages are searched for. These are pages which contain references to a translation of itself.
The second step is to get pairs of pages which might be translation of each other. If a parent page has been found, the references of this page will be paired together. If a sibling page has been found, it will be paired together with the reference on his page. In case several considered references have been found, all the URLs will be collected, then separated into the different languages by using language identification tools and eventually, the cross product will be built over two language sets of pages. To filter out pairs which are likely not be translations, STRAND takes use of the fact that many web sites are organized in a file system alike hierarchy where translations are stored in a certain parallel structure. Another filter method is comparing the length of the documents which are supposed to be translations.
At the end the remaining pairs will be aligned based on the underlying HTML format. First the HTML document will be "linearized", meaning tokenized into opening tags, closing tags and text chunks. For latter only the length of the text will be saved. After the alignment has been done, features will be calculated. Resnik and Smith used manually corrected thresholds to decide whether an aligned document is a translation or not.
In a second approach they used machine translation to determine the threshold values. They achieved 100% precision and 68.8% recall using the manually set thresholds. Having calculated the thresholds by machine learning, an average of 96% on precision and 84% of recall could be gained. On a third approach they substituted the alignment process of paired documents with translation similarity scores. This way an F-score of 0.875 could be achieved. In order to avoid crawling the web on a huge base, Resnik and Smith suppose to take advantage of databases which held information about the web in a large scale. Using such databases offers the advantage of having shorter IO waiting time as well operating on a finite network. They used The Internet Archive[Web] to accomplish it.

2.2.1.2 DOM (Shi et al.)

Shi et al. proposed a different approach[SNZG06] to find the actual translated phrases within two documents. Their aim is more restricted since they try to collect parallel data instead of comparable translations. Concerns on high bandwidth and slow download speed led Shi et al. to the conclusion that precollection of web pages has to be avoided. For this reason they dismissed the idea to find candidate pairs by testing their URLs on similarity. Because there are more comparable translations than parallel translations of pages are on the web, the procedure would waist time downloading all the comparable translated documents. A second issue Shi et al. wanted to address in their approach is that sentence alignment so far at the time were based on assuming a plain text document. Their concern is that a web page contains a lot of non translational and therefore non relevant phrases as well as out-of-vocabulary words. All this would make the alignment less accurate. To face these two concerns - unnecessary downloads and inaccuracy in the sentence alignment - Shi et al. suggested a stochastic alignment method based on the document object model (DOM) of web pages.
Starting at a given web site, the root web page and all web pages whose reference are on the root web page will be downloaded. Then these web pages are searched for trigger words in the text of hyper link references. Trigger words are words which indicate a possible translation.

English translation	English, English Version, ...
Chinese translation	Chinese, Chinese Version, Simplified Chinese, Traditional Chinese, ...

Table 2.1: Signal words for language anchors

The web site will be considered bilingual, if trigger words for the two languages of interest have been encountered on the web pages. The web pages will be paired together and then the pairs will be tested on whether both web pages in a pair are actual translation of each other. For each pairs of web pages which passed the verification test, the document object model tree of both web pages will be built. The aligned text chunks and hyper links will be extracted out of the document object model trees. For each aligned text chunk sentence alignment will be carried out and for each aligned hyper link references they will be passed over to the verification test mentioned above. From there on they go through the same procedure. The iteration process stops as soon as no more parallel hyper link references are left. The verification test will be done by a binary maximum entropy based classifier. Three features are used.

- file length ratio
- HTML tag similarity: all tags of a page will be concatenated and then the minimum edit distance between the concatenated tags of both web pages will be calculated. The HTML tag similarity then is defined as the ratio of match operation number to total operation number.
- sentence alignment score: ratio of the number of aligned sentences and the total number of sentences in web pages

Shi et al. tested their approach on English - Chinese bilingual web sites. For the precision of mined parallel documents they obtained 93.5% using URL pattern according to Resnik and Smith and 97.2% using their own method of DOM tree alignment. Evaluating the accuracy of the sentence alignment they recieved 86.9% precision and 79.4% recall without using a DOM tree and 93.4% precision and 86.6% recall using a DOM tree. In terms of download efficiency Shi et al. could show that mining based on DOM tree alignment

increases the parallel data acquisition throughput by 32% compared to mining based on URL pattern matching. Also regarding the ratio of downloaded pages per parallel pair, mining based on DOM tree alignment results in 2.26 whereas mining on URL patter matching only gives 19.38. As the authors mention correctly, it shows that the bandwidth usage is almost optimal. Comparing the amount of pairs of parallel web pages which have been discovered by using URL pattern matching and DOM tree alignment, it shows that using latter method covers 22.7% more. Using both should increase the throughput by 41%.

2.2.1.3 Anchor elements (Nie et al.)

Nie et al. tried to replace the Hansard corpus which is a collection of English-French parallel texts. It covers 8 years of Canadian Parliament debates. What they wanted to replace it with instead is parallel text collected from the web. Their approach is split up into two steps.
The first step is to select candidate web pages. To estimate whether a web site is bilingual and therefor has candidate web pages, the HTML text of the web pages will be searched for specific anchor text. The anchor texts should indicate a mutual translation. Text such as ”en français” or ”French” is likely to hint to a French translation whereas ”in English”, ”en anglais” or simply ”English” is very likely to hint to an English translation. If web pages within the same web site reference themselves via such anchors they are considered to be candidate web pages. Of course the references then must contain trigger words for different languages. To get proper starting seeds, Nie et al. suggest to use web search engines such as AltaVista or Northern Light. Looking out for documents in one language which also contains anchor text indicating the other language, a list of web sites is returned which then will be tested on bilingual web pages.
If a web site is determined as bilingual, parallel texts within the web site now have to be paired together. Instead of comparing each document with every other document within the web site, Nie et al. propose a heuristic method to pair documents together. First of all the URLs of the web pages will be tested on similarity quite like Resnik and Smith proposed[RS03]. On a preliminary evaluation test Nie et al. encountered a precision of over 95% and a recall of at least 50%. Secondly the HTML document structure of both web pages which are about to be compared, will be tested on similarity too. A third criterion Nie et al. used is a text length filter. On 1000 randomly selected candidate pairs the result of the text filter differed by 2% from the result using an alignment filter. Advantages of a text filter compared to an alignment filter are a simpler implementation and a faster performance.

2.2.2 Semantic based searching

Searching based on semantic information means that any structural information will not be considered. Only information from the actual content is taken in order to find bilingual web pages.

2.2.2.1 Translation comparison (Ma et al.)

Another architecture, BITS, was proposed by Ma and Liberman[ML]. According to a web survey which Ma and Liberman carried out 1997, the chances to find an German - English bilingual web site in the de domain lies at 1 to 10. They assume that for other domain like the Canadian domain ca the chances to find bilingual web sites are quite the same. Having found such a domain, a list of web sites for crawling can be obtained by querying some DNS server. For each web site the pages will be tested on how many languages can be identified. They limited the language identification to the pages which can be acquired

in at most 3 or 4 recursions. After filtering out all monolingual web sites, the web pages of the remaining web sites will be downloaded and converted into plain text. If the resulting plain text does not exceed a certain threshold of text size, it will be thrown away. The rest of the text files will be separated into documents in language L_1 and documents in language L_2. The separation again is done by using language identification tools. Having these two piles of text documents, matching translation pairs have to be found.
Ma and Liberman reject the idea of finding pairs by comparing the path segment in their URL on similarity. Since the idea is based on the assumption that web designers will name files with the same topic but written in different languages similarity, it appears to be not very robust knowing that this behaviours of web designers can change very quickly. Also the fact that there are different perspectives on the same topic for different viewers leads to the consequence that different aspects of the topic will be emphasized and hence it does not seem so likely to find matching pairs this way. Ma and Liberman also disapprove the approach of finding matching pairs by comparing the underlying HTML formats on similarities. The approach to compare the underlying HTML format is due to the expectation that related web pages have the same or nearly the same HTML structure. Even though the assumption is correct for many cases, it does not work that well. A lot of related web pages are not mutual translations but in fact use only the same template for the HTML document structure. Additionally mutual translation pages do not need necessarily a similar html document structure. So using this comparison method probably drops matching page pairs. Aside from that it will not work for web pages which have a simple HTML document structure.
Ma and Liberman propose a method which tries to imitate the way human beings recognize translation. This is by concentrating purely on the content and having a certain degree of knowledge about the languages of interest. Therefore Ma and Liberman developed an algorithm which counts how many translations of token in document A can be found in document B. If this value exceeds a certain threshold relatively to the number of tokens in document A, document A and document B are supposed to be mutual translations. For certain language pairs cognate words can be used instead of actual translations. To exclude wrong translations the algorithm applies a distance filter to consider only within a specific range which depends on the languages and the genre of the documents. In order to make the algorithm faster, a pre filter will be applied to the candidate pairs. If the size of the two documents A and B are too different, or proper nouns do not appear in both documents, etc. , then both documents will not be tested on similarity.
For the language identification Ma and Liberman used a tool which gave them a 100% accuracy for text over 500 bytes in 13 languages. Testing the translation pair finder resulted in 97.1% recall and 99.1% precision.

2.2.2.2 Querying translations (Munteanu et al.)

Munteanu et al. propose a strategy of getting parallel fragments of texts within non parallel, bilingual corpora[MM]. Previous works to extract translations out of the Web focused on documents which contains parallel text paragraphs. However, Munteanu et al. aim to find translations on sub-sentential level. So even though two sentences are not considered as mutual translations of each other, there still might be some fragments within the sentences which are translations. This scenario is quite realistic, since only a few documents will be translated with the purpose to have the exact information transformed into the target language. Often informations will be added, retained or extended depending on the interest of the target audience. For this reason corresponding information in bilingual documents are likely to be not sentence aligned.
Munteanu et al. start from a parallel corpus which is used to build two lexica. The first lexicon is done by running the GIZA++ implementation of the IBM word alignment models. The GIZA++ lexicon has a mediocre precision but a quite good completeness. So it

will be used to filter out roughly wrong alignments. The second lexicon is based on Log-Likelihood Ratio. It is more precise than the GIZA++ lexicon and contains information how likely two words are translation of each other as well as how likely two words are not translation of each other. Then from a set of comparable corpora, each document in the source language will be translated word by word and the translation will be used as a query to find matching documents in the target language. For each document in the source language the top 20 results will be paired together. All sentence pairs of each document pair will be applied to a filter which discards sentence pairs which have not enough translation words. For this filtering and the document matching the GIZA++ lexicon is used. In a last step the remaining sentence pairs will be employed to a fragment detection which results in parallel fragments. For the fragment detection the Log-Likelihood-Ratio based lexicon is used.
Testing their approach Munteanu et al. carried out experiments to translate from Romanian to English. The initial parallel corpus is taken from a workshop about statistical machine translation and the Romanian translation of the European Union's acquis communautaire. The collection of comparable documents are retrieved from online new sites and newspapers. As a Baseline system they used a previous approach which is virtually identical but operates on sentence level and not on sub-sentence level. In total they registered an improvement of 1 BLEU% having a confidence interval of 95%. Hence this improvement is statistically significant. Additionally they could show that going on subsentential level gives better results than operating on sentence level. Also Munteanu et al. found indications that a lexicon based on their Log-Likelihood-Ratio scoring performs better than a lexicon compiled by the GIZA++ implementation.

2.3 Methodology

To find comparable corpora in the area of research and academical education we exploit the fact that most sites of German universities offer a German and an English version of their pages. In fact the corpus will be comparable at most as we discussed before. A list of German universities can be found on Wikipedia[wika]. Following the links on the page leads to the Wikipedia page of the respective university where the actual link to the web site of the university can be found. Since Austria and a part of Switzerland are German speaking too, their universities also can be considered. A list of their universities can be found on Wikipedia too[wikb][wikc] . Of course the universities in the Italian and French speaking part of Switzerland are more likely to offer an Italian-English respectively French-English translation. Having collected the links to the web sites of the universities, the start point of the crawling process is defined and actual crawling can start.

Different approaches of crawling can be pursued. The first decision to make is how to determine bilingual pages. The proposal of Ma and Liberman [ML] may need a lot of memory and performance time if a web site contains a lot of web pages. For bigger universities this will inevitably be the case. Saving all the pages and then comparing each with every one might not be the best way to go since we want to do it for a lot of web sites which basically have many web pages and are expected to be bilingual. Generally spoken, a faster and less memory using method like the URL matching procedure[RS03] or the trigger word procedure[SNZG06][NSID99] is required.
Another issue is how to balance the memory usage and the amount of page downloads. Keeping the amount of page downloads down leads to higher memory usage. Since download calls are quite slow and several server systems have mechanisms in place to prevent high frequent automatic requests, it is preferable to keep the amount of page downloads low. On the other hand, we may find a lot of pages as discussed before. Our suggestion is to split the crawling process into a precollecting phase similar to Resnik and Smith[RS03] and the actual segment aligning phase. This means that all candidate web pages have to be downloaded twice. But since this happens with some time delay in between, it shouldn't cause any problem. After having aligned the segments we use a classifier to distinguish between correct aligned segments and wrong aligned segments.
There is some alternative for crawling the web. It has been suggested by Resnik and Smith[RS03]. Some databases has been set up to keep copies of part of the web. Google offers Google Cache in their search service to provide copies of the web sites. Usually these are snapshots of the last time the search crawler visited the web sites. Resnik and Smith used the Web Archive[Web] which has a large collection of web sites of the whole web. Resnik and Smith therefore could operate remotely on machines where the database was located. Consequently the time cost for IO calls were reduced significantly. Right now this service is not longer offered. Instead a JSON API has been implemented to get an alternative access to the web interface. Unfortunately the information about how to use the JSON API is missing on their web site. So for this thesis the suggestion of using the Web Archive cannot be pursued. Other archive services like the Google Cache or the Bing Cache are deeply integrated into the web interface and primarily supposed to be used via the web interface. So these services are not available either to replace the crawling procedure.

2.3.1 Precollecting based on URL

In a first attempt we try to follow the approach of URL similarity matching according to Resnik and Smith[RS03]. For that we build a simple crawling application which goes through a web page, collects all links, filter them, save the links which passed the filter and put them into a queue from where the next url is taken to get through the same

procedure. The filtering process is described in details below. It basically makes sure that first only URL references of the same web site will pass and second only web pages which contains HTML formated text will pass. Having the collection of all link references, the collection will be searched for trigger words. These are simple phrases which might indicate some existence of language version. If an URL reference contains an "en" as one of its path segments or a parameter "lang=en", it likely indicates an English version of the containing document. Doing so for two languages L_1 and L_2 gives two sets S_1 and S_2 of URL references. At this point one could follow the approach of Resnik and Smith to pair URL references from both sets based on their similarity. This is done by replacing in a URL reference of S_1 the occurrence of each trigger word of L_1 with every trigger word of L_2 and then looking for such an URL reference in set S_2. Another way to find matching URL references is a simple brute force method. For each URL reference in set S_1 we substitute the occurrence of trigger words of language L_1 by every trigger word of language L_2. For each URL reference in set S_2 we do the same but the trigger words will be replaced from L_2 to L_1. Then the new created URL references will be tested whether they exist or not. Fortunately the HTTP protocol supports a way of testing a URL reference without the need of downloading the actual content. By using the "HEAD" method only the header part of an http response will be delivered. In case a "HEAD" request results into a successful response we consider the URL reference of the request and the original URL reference as a candidate pair.

The procedure we described so far has some disadvantages but on the other side has a strong advantage and this is its simplicity to implement it and that it does not need any training or manually fixed bias values. Only the set of trigger words vary from language to language but still are easy to determine. Disadvantages are that a lot of page downloads will happen. Let us suppose that each URL reference in S_1 and S_2 contains only one trigger word and that we have t_2 trigger words for language L_1 and t_2 trigger words for language L_2. The amount of page downloads will be $t_1 \cdot |S_1| + t_2 \cdot |S_1|$. In best case there is a counterpart for each URL reference. This would mean that only $|S_1| + |S_2|$ "HEAD" requests will be successful. So at least $(t_1 - 1) \cdot |S_1| + (t_2 - 1) \cdot |S_2|$ page downloads will be unsuccessful by design and hence are unnecessary.
Another concern besides the unnecessary downloads is how many bilingual web sites will be missed if the bilingualism is determined only by the URL of the web pages. Nowadays many web sites use hashed names for their web pages. This is due to simpler maintenance of the web site and abstracting the URL of a web page from the actual relation which the web page has to the other web pages in the web site. All such web sites will not be recognized as bilingual. To collect these web sites the content of the web pages has to be analyzed. There is also another case where looking out for specific text snippets in the URL will not help to determine correctly. Some bilingual web sites use the equivalent word for the respective language versions. So on the web site of Karlsruhe Institute of Technology is a web page about the different departments. The URL for the German version is "http://www.kit.edu/studieren/fakultaeten.php". It contains the German word for "study" which is "studieren" and the German word for "department" which is "fakultaeten". The URL for the English version is "http://www.kit.edu/study/departments.php". In order to recognize the web pages as bilingual only based on their URL, a bilingual lexicon would be required and even then only translation entries in the lexicon could be identified effectively. Determining bilingualism solely by looking at the URL might not use the full capacity of the Web. Thus another concept of crawling will be pursued which takes advantage by analyzing the actual HTML document of the web pages.

2.3.2 Precollecting based on page intern references

To get a more reliable procedure of finding bilingual versions of pages, we going to analyse the hyper link references in an html page. If the linking reference is likely to lead to a bilingual version (either German or English), it will be considered as a candidate. If it happens that the page of the linking reference also contains a reference to the first page and the reference is likely to lead to a bilingual version, both pages then will be considered as bilingual version of each other. For each pair of pages we considered as bilingual versions, the text will be extracted out of the respective web page, segmented into sentences and then the cross product of will be built. So for one pair of pages each sentence of the first page will be aligned to each sentence of the second one.

Due to some problems which will be described below, the process has to be split up into three steps: Crawling, Resolving and Aligning. At the very end we get a corpus of aligned sentences, of the one side English and on the other side German. This corpus then will be used to find new vocabulary.

2.3.2.1 Crawling

The crawling process follows the usual way of crawling. Starting at a certain set of web pages, it will download a web page, analyse it and then follow all the references which have been found on the web page. To prevent crawling a page more than once, each crawled page has to be marked, meaning its URL will be saved. Only html formated pages are helpful, all web pages which are not html formated will be filtered out. To do so the http response for each requested web page will be examined whether the Content-Type header field contains some invalid type 2.2. If so the suffix in the path segment of the URL will be extracted and saved as an suffix for an invalid content type. Valid content types are:

text\html	indicates html formated text
text\raw	indicates raw text, but some web sites use it for the pages anyway
<empty>	an empty or not existing content type should be avoided according to http 1.1 protocol; in case it happens, it is recommended to guess the type based on the URL or the content. Since known suffixes of invalid content type has been filtered out at this step, the content simply will be considered as html formated text.

Table 2.2: Accepted content types for crawling process

Each URL which comes from the queue to get crawled will be tested on this set of suffixes and will be rejected if the test was positive. This way a lot of references to PDF formated files and image files which are quite frequent on web pages will be rejected and will not be marked as visited. If the http response indicates a redirection, this redirection will be registered and the triggering URL will be marked as visited. If the http response indicates an error, be it on client or server side, the URL nevertheless will be marked as visited to prevent requesting it a second time.

If the web page has been delivered successfully, the html structure will be tokenized. Unfortunately, the html standard is not that restrictive as the XML standard is. So it does not make sense to build a syntax tree. Instead parsing the content has to be done based on the sequence of the html tokens. Because often HTML pages are not pure machine built files - meaning someone has contributed a certain part of directly formulated html text - parsing the html content must be done in a way that it tolerates mistakes.

Language identification

Having all html tokens, the language of the content will be estimated. For this purpose all text from the web page will be extracted and filtered in such a way that only words without non-alphabetic characters remain. Then all words $< w_1, ..., w_n >$ will be matched against a collection $V_{\text{most frequent}}$ of the most frequent words in the language L. If a certain threshold δ_L has been exceed, text of the web page will be considered to be in the language L.

$$\delta_L = \frac{|\{w_i | w_i \in V_{\text{most frequent}}\}|}{n} \tag{2.1}$$

If the text of a web page exceeds the threshold δ_L for language L, the web page will be searched for link references which are likely referencing to pages with the same content but being not in language L. All these references then will be registered as candidates for bilingual pages where the references are the target and the current page is the source.

Determine bilingualism

The likelihood of being a reference to a page having the same content but being in another language - to say being a comparable translation - will be determined by the appearance of specific keywords in the tag attributes as well as in the text between the opening and the closing reference tag. This consideration is based on the experience that a link to a translated version is signed by the name of the language or the language code or an image of the flag of a country where the language is spoken. In case of latter there is often an alternative text which consists some hints about the language. The HTML5 standard contains several mechanisms to support crawling, but unfortunately most of these mechanisms are optional and the standard has not been widespread so far. If at least one keyword appears, then the reference is considered as a candidate. To keep the rate of wrong considered candidates low, the actual reference, the value of the "href" attribute, will not be searched for the keywords.

After all all links on the web page will be pushed into the queue of URLs to crawl as long as the haven't been filtered out. The filter will only passes URLs which do not have an invalid suffix and which belong to the domain of the current page. Belonging to a specific domain may also mean belonging to a sub domain. Thereby any leading "www" will be ignored(2.3).

domestic host	host of reference	sub-domain
www.kit.edu	www.isl.kit.edu	yes
www.kit.edu	isl.ira.kit.edu	yes
www.kit.edu	www.facebook.com	no
www.informatik.kit.edu	www.mach.kit.edu	no

Table 2.3: Examples for sub-domains

Crawling the web one problem generally occurs. It is the exponential growth of the amount of pages which will be searched. Let us consider a as the average amount of new pages if a page has been visited. Considering going only to the nth recursion it produces n' URLs to visit.

$$n' = \sum_{i=1}^{n} a^i - \sum_{i=1}^{n} i \tag{2.2}$$

If the sites we are searching have not many pages it is not such a big concern because the crawler might have visited all pages before the amount of new pages would have increased dramatically. For the university sites this does not work. To bound the exponential growth

to a certain limit each URL to crawl has been assigned a number of recursions left. If this number reaches zero, the references found on this page will not be followed any more. Each new found reference will get the recursions left number of it parent page minus one. However if the reference is one of the bilingual candidates it gets the same number of recursions left. This way we are going to shrink the search space those URLs we consider as potential candidates.

2.3.2.2 Resolving

Having all the information about potential candidates now they have to get confirmed. This is done by finding pairs of references which are symmetric version of each other. Since redirections happen during crawling, it is critical to know about all redirections before the process of finding matching pairs can start. If redirections are not known, it is not possible to be sure that the second part of each pair - the target of the reference - actually is a web page or just a redirection. Information about redirections have been gained in the crawling process before. But of course it contains only redirections which have appeared during the process. Other redirections will stay unknown and hence cannot be used in the resolving process.

To keep the memory usage low we take advantage of the fact that information about candidates and web pages which have been visited are in chronological order:
First all references of possible candidates are given and then some hint about the fact that the web page has been completely crawled is given. The information about a candidate pair will be processed in following way:
The target site (the actual reference) will be tested whether it is a redirection or not. If the test is positive - meaning it is a redirection -the target site will be replaced by the target of the redirection. After testing on redirection, the set of URLs which are linked to the source site so far will be searched for the target site. If it is found then a confirmed pair has been found. If not, the target site will be tested on whether it has been visited so far. If it has been visited, the candidate pair will be ignored, because there is obviously no symmetric reference on the target site. If it hasn't been visited, the source site will be linked to the target site, so when the target site will be visited, it will see which web pages are linked to it. Eventually the "web page has been visited" information appears and at this point all web pages which are linked to current web page can be removed.

2.3.3 Text segment aligning

For each confirmed pair of URL, the page will be downloaded, the text will be extracted, split into segments and then the cross-product is built over segments of the first and the second web page.

To obtain only the text which actually will be display on the web page, not all text from the web page can be used. Often the web page contains JavaScript code and Style code which is not of interest. Therefore text between specific opening and closing tags will be ignored **??**. To take advantage of the fact that HTML contains information about text blocks, these text blocks will separately be taken to divide into sentences and segments.

Apparently text on a web page consists not only of well formated sentences but also of short text snippets which are used often to label navigation elements. For that words in this text snippets have mostly strong semantic value. So these text snippets also will be considered besides actual sentences.
To get sentences out of the text blocks, they have to get segmented. Usually punctuation marks indicate sentence boundaries. The issue here lies in the fact that punctuation marks also indicate part of sentences as well as abbreviations and ordinal numbers. In addition to

tag areas to ignore	script, style, noscript, noframe, form, object, pre
tag areas indicating a text block	h1, h2, h3, h4, h5, h6, p, div, caption, dt, dd, blockquote, li, th, td
single tags indicating a separation between two text blocks	br, hr

Table 2.4: tags to ignore and to consider in the body area in order to acquire valid text blocks

that there are variations between different languages. So ordinal numbers are indicated in English by adding their two last letters to the digits whereas in German it is either written out (conventionally all numbers up to twelve) or a dot is added to the digits. Another different handling can be observed on dates. In German the day of the month and - if not written out - the month itself are ordinal numbers and hence dots are used contrary to the English way to write out a date.
One way to face this issue is to build a classifier which will be trained to recognize whether a punctuation mark is the actual end of a sentence or not. Using a supervised classifier[nlt], several features will be calculated:

- whether the next word is capitalized
  In many European languages - thus English and German too - the first word of a sentence has to be capitalized. But there are situations where the first word after a punctuation mark is capitalized, both in English and in German. In German all nouns have to be capitalized regardless of their position in the sentence. Especially in dates it happens that the name of the month is written out and is preceded by the dot of the day. This dot does not indicate the end of a sentence even though it matches the situation of a real sentence termination. In English proper nouns amongst others have to be capitalized. Being behind the dot of an abbreviation does not terminate the sentence either. So this feature is only a necessary condition and not a sufficient one. An exception is if the sentence is at the very end of the text and thus no word is following.

- the previous word
  Abbreviations in German are usually marked by a following dot. In British English and American English it varies. So knowing the most common abbreviations may help to determine whether a dot is indicating an abbreviation or the end of a sentence. If the previous word is a figure and the punctuation a dot, it is more complicated to determine whether the figure is an ordinal number or it only happens to be at the end of the sentence. To solve this problem it takes more information about the context. If necessary conditions are not fulfilled, it can be assumed to be an ordinal number. Otherwise more semantic information are required.

- the actual punctuation mark
  Only a few punctuation marks are used to mark the end of a sentence. In German these usually are dots, question marks and exclamation marks. The same is true for English. Colons will not be handled as sentence boundaries. For other languages such as Spanish there are different punctuation marks like inverted question marks or inverted exclamation marks. Fortunately such more unusual punctuation marks do not occur in English or German.

- wether the previous word has only one letter
  The chances that a word is an abbreviation is much higher if the word contains less letters. For words which contains only one letter it is very likely to be abbreviations.

Apart from variable names, there are only three words in English which consist of just one letter. These are ”a” (also appears as ”A”), ”I” and ”O”. In German no words are known which have only one letter. So if an one-letter-word precedes a dot it is very likely to be an abbreviation. But it is not sufficient since the word could be a variable name or some case like ”It was us, John and I.”.

In this feature set some cases are not covered. It will not be gathered whether a space character is following the punctuation mark or not. Since we are shrank to exactly two languages, German and English, the important punctuation marks are known and other punctuation marks do not need to be observed. Instead of a classifier we use a decision tree 2.1. It does not need any training and can be put directly into code. It basically test whether the punctuation mark is a dot, semicolon, question mark or exclamation mark. If so it test whether it is at the end of the text. If it is in the middle of the text and followed by a space character, it will be accepted as sentence boundary unless the punctuation mark is a dot. If the dot is preceded by a word with more than a specific amount of letters, it is accepted as sentence boundary.

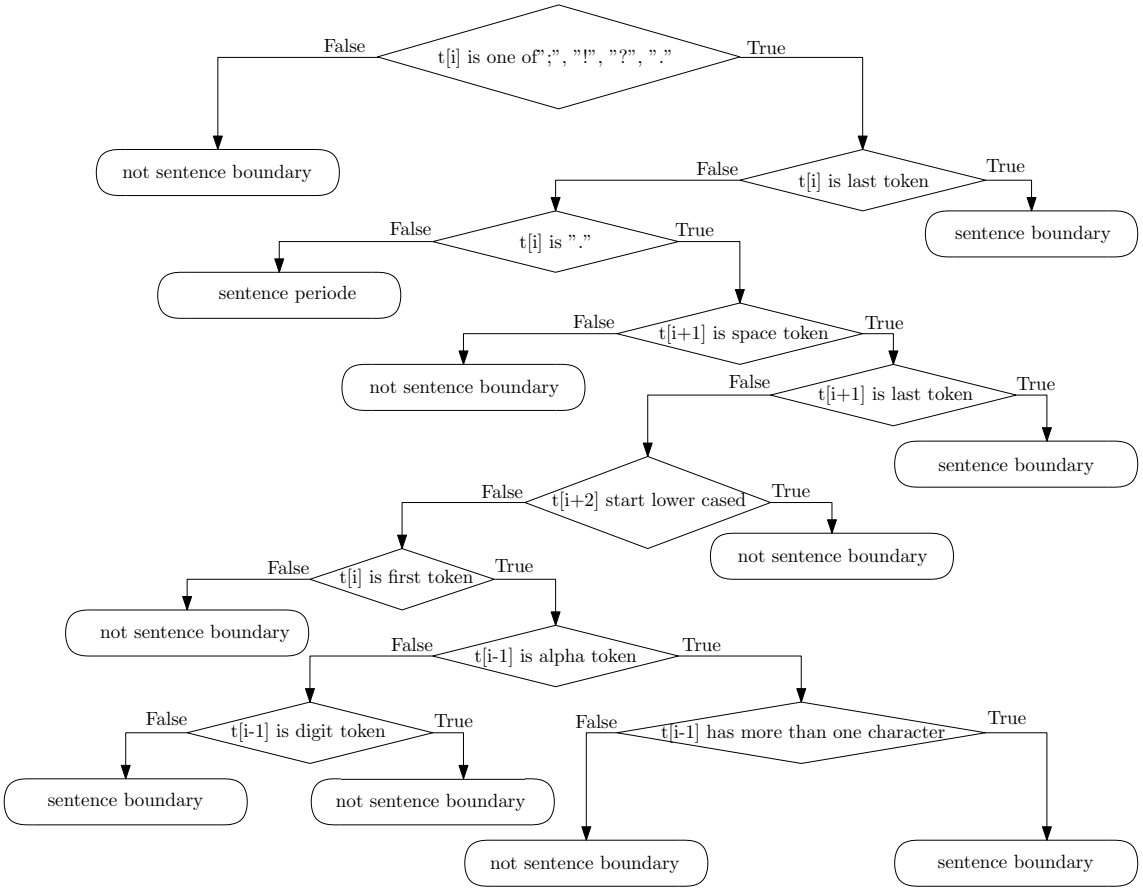

Figure 2.1: Decision tree to segment text block into sentences

The small text snippets mentioned above usually do not contain any punctuation marks which help to indicate the end of a sentence. So they will be treated as one sentence respectively.

Building the cross product - on the one hand - makes sure that no correct alignment will be missed. On the other hand, a lot of incorrect alignments will appear. Considering a certain amount of sentences m on one web page and n on another web page, the amount of correct sentences will be limited by the maximum of m and n. This limitation is only valid as long as parallel elements in sentences are aligned to only one sentences on one of

both sides. So if one sentence contains information which is represented by two sentences on the other language, but in the same time on the other language they same phenomena appears, this limitation does not work generally any more. But for further calculation, such cases will be excluded. The cross product will produce c alignments.

$$c = m \cdot n \tag{2.3}$$

With our consideration at most d alignments can be correct.

$$d = max\{m, n\} \tag{2.4}$$

By design, at least e alignments will be wrong.

$$e = c - d = m \cdot n - max\{m, n\} \tag{2.5}$$

For bigger m and n, the amount of wrong alignments e can be quite high. In fact the gap e is proportional to the squared minimum of n and m. Lets consider i the minimum of m and n and δ the difference between the maximum and minimum of m and n.

$$\delta = max\{m, n\} - min\{m, n\} \geq 0 \tag{2.6}$$

We can reformulate the amount of wrong alignments e.

$$e = i \cdot (i + \delta) - (i + \delta) = i^2 + i \cdot (\delta - 1) - \delta \tag{2.7}$$

If both m and n increase, the gap grows in terms of the squared minimum. The distance δ between maximum and minimum has almost no influence but rather helps to increase the growth if the minimum is big enough.

The calculations have shown that the cross product will produce wrong alignments in a super linear way. Thus the collection of sentences should be preferably small in order to avoid wrong alignments.

2.3.4 Getting comparable sentences

After dividing text snippets into sentences the corpus will be preprocessed. The preprocessing is done according to the Statistical Machine Translation system of Karlsruhe Institute of Technology[HMNW]. This involves:

- normalizing special symbols
- smart-casing the first word of a sentence
- removing long sentence and sentences with length mismatch

The preprocessed corpus then will be evaluated whether the aligned sentences are mutual translations. For the classification task a Support Vector Machine is used. We used a Support Vector Machine implementation from Karlsruhe Institute of Technology[HMNW]. Following features are taken into account:

- IBM 1 score from source language to target language
- IBM 1 score from target language to source language
- fraction of unaligned words in source language
- fraction of unaligned words in target language
- maximum fertility ratio from source language
- maximum fertility ratio from target language
- length ratio

According to Herrmann et al.[HMNW] a Support Vector Machine gives the best results amongst Regression, Logistic Regression and Maximum Entropy in classifying correct and incorrect translations.

2.4 Test

To estimate how well the crawling and translation extraction procedure is going, several tests will be performed. First the language identifier will be examined on how well web pages in German and English can be determined. Then the performance of segmenting a web page into phrases will be analyzed. Also the alignment process of phrases of two web pages will be evaluated. At the very end, both crawling based on URL and crawling based on translation links will be tested.

2.4.1 Language determination

	German	English
training size in web pages	753	754
positive training entries in web pages	417	337
negative training entries in web pages	336	417
mean ratio of matched words in positive training entries	0.501291	0.614665
variance ratio of matched words in positive training entries	0.00721628	0.0130792
mean ratio of matched words in negative training entries	0.289594	0.135713
variance ratio of matched words in negative training entries	0.00319375	0.00292017
calculated threshold	0.383245	0.298921
testing size in web pages	394	504
correctly accepted web pages	189	166
wrongly accepted web pages	13	17
correctly rejected web pages	174	317
wrongly rejected web pages	18	4
precision	0.935644	0.907104
recall	0.913043	0.976471
true negative rate	0.90625	0.987539
accuracy	0.92132	0.958333

Table 2.5: Language detection results

In order to determine the language of a web page the ratio of words matching a set of most frequent words in the language and the total amount of words on a web page will be calculated. If the ratio exceeds a certain threshold, the web page will be supposed to be in the language. The threshold used in crawling process which will be described below is obtained by training a Gaussian classifier. The training set is a collection of URLs and a notation whether the web page behind the URL is in the language or not. The collection of URLs has been gathered by crawling university web sites without looking out for mutual translation web pages. Then the web pages has been assigned manually to German, English or a different language. A fraction of this labeled collection is declared as test set and the rest is used to train the classifier.
A list of the top 1000 most frequent words in German and English has been taken from University Leipzig[Unia] [Unib]. The training set for the German language classifier has 753 web pages. There are 417 web pages in German and 336 web pages not in German. The training result shows that 38% of matched words in the web page indicates a German web page. The test set contains on 394 web pages of which 207 web pages are in German and 187 web pages are not. Testing 2.5 the threshold obtained by the training gives a precision of 93.56% and a recall of 91.30%. The same training set has been used to train a classifier for the English language but of course having replaced the annotations so that now English web pages are labeled as positive and not English web pages are labeled as negative. Some web pages couldn't be downloaded during the training process. Due to that the amount of considered web pages for training slightly differs. The resulting

threshold for detecting English web pages is about 30%. Testing this threshold gives a precision of 90.71% and a recall of 97.64%.

2.4.2 Dividing text of web page into segments

Another step in gathering a corpus is to divide the text of a web page into approbriate segments. The structure information given by the underlaying HTML document allows to narrow down the whole text of the web page into smaller text chunks which then will be split off into sentences. To test how well the designed algorithm performs, 10 randomly selected web pages are taken out of 463477 web pages which have been crawled before. The hypothesis corpus were made by simply downloading the web pages and the passing it on to the algorithm which gives all segments. The reference corpus were made by downloading and exporting the web pages into plain text files via the application *lynx*[Lyn]. After that the plain texts were formed manually into sentences and simple text snippets.
As illustrated in 2.4.2, the intersection of hypothesis and reference corpus is 1412 entries, 27 entries of the reference did not occur at all in the hypothesis and 48 entries of the hypothesis could not be assigned to any entry in the reference. This results in a precision of 96.7% and a recall of 98.1%.

correct hypothesis segments	1412
incorrect hypothesis segments	48
missing hypothesis segments	27
precision	0.967
recall	0.981

Table 2.6: Text segmentation results

The precision and recall figures show that the algorithm performs quite well. But one has to bear in mind that most of the web pages contain a lot of navigation elements and graphical elements such as tables to visualize information rather than plain written text. So results may vary if web pages which are about to get examined have more text content. The not matching entries in the reference corpus may be caused by the *lynx* application. Not matching elements in the hypothesis corpus partly comes from errors done during sentence dividing. The algorithm fails to recognize abbreviation which are finished by a dot and have more than one character. One of the abbreviation which accidentally is understood as sentence boundary is Dr.̈ To improve the algorithm regarding this misunderstanding a set of common abbreviation might help to identify most of the abbreviations with more than one character. Another issue is to distinguish cardinal numbers and numbers which happen to be placed at the end of a sentence. The designed algorithm always assumes a cardinal number, so consequently does not see the sentence boundary. Evaluating the context might uncover hints to find out whether it is a cardinal number or simply a number at the end of a sentence. In a more sophisticated algorithm it might be wise to consider the different rules in different languages. The question of cardinal number or regular number actually is misplaced in English since cardinal numbers are marked differently and without punctuation marks.

2.4.3 Alignment

The alignment process uses the brute force method of building the cross product between two sets of sentences. To retrieve some information about the performance the alignment procedure will be monitored for a test set. The test set contains 10 pairs of URL randomly picked up from 2473 pairs of mutual bilingual web pages. These pairs were collected by crawling university web sites and looking out for bilingual web pages. The test set were

controlled manually to make sure that the used pairs actually are mutual translations. For each URL pair, the correct alignments are counted manually and then mean and variance are calculated. As discussed before in Methodology 2.3.3, we can assume a certain upper boundary of correct aligned sentences. So in the results **??**, both the assumed upper boundary and the amount of alignments produced by the cross product will be considered. By design the recall is 100%. Taking the amount of alignments produced by the cross product as the baseline, in average there are about 1% correctly aligned sentences with a variance of 0.006%. Taking the upper boundary as the baseline, there are 82% alignments correct in average with a variance of about 5%.

mean of correct alignments (cross product)	0.0117
variance of correct alignments (cross product)	0.000065
mean of correct alignments (upper boundary)	0.8165
variance of correct alignments (upper boundary)	0.0576

Table 2.7: Alignment results

The mean of correct alignments considering the upper boundary shows that some improvement can be done. If looking into the pairs of web pages in 2.8, it shows for which pair the alignment went well and badly. Inspecting the alignments of No. 10, the German web page contains more information than the English web page. In another case (No.7) it is the opposite situation: The English web page contains much more information than the German web page. The alignment No.6 has a balanced amount of text content. What makes it so unequal is that on the German web page there are significantly more navigation elements. The remaining web pages show a precision of above 90%.

No.	German	English	cross product	correct	correct (cross product)	correct (upper boundary)
1	49	48	2352	47	0.0200	0.9592
2	196	192	37632	187	0.0050	0.9541
3	108	108	11664	105	0.0090	0.9722
4	159	157	24963	148	0.0059	0.9308
5	167	158	26386	158	0.0060	0.9461
6	97	39	3783	35	0.0093	0.3608
7	38	63	2394	30	0.0125	0.4762
8	109	105	11445	105	0.0092	0.9633
9	32	32	1024	32	0.0313	1.0000
10	103	66	6798	62	0.0091	0.6019

Table 2.8: Alignment results for each test web page

Of course after all the precision has to be calculated considering the result of the cross product. It basically performs quite badly but this is due to the design. A better approach needs to take the web page structure into consideration to reject unlikely alignments.

To estimate the average amount of sentences per web page, 247 randomly selected pairs are taken from the crawling process discussed below. Then the sentences are extracted by the sentence dividing algorithm and counted. The maximum of both counts for each pairs is considered. The mean is 130.696 sentences with a variance of 11230.7. It clearly shows 2.4.3 that the amount of sentences per web page is dispersed highly. Also the average size of each cross product will be calculated on this pair set. As it is for the maximum sentences, the variance displays a high dispersion.

Amount of samples	247
Mean amount of maximum sentences	130.696
Variance amount of maximum sentences	11230.7
Mean amount of cross product	19773
Variance amount of cross product	$9 \cdot 10^8$

Table 2.9: Average amount of sentences per web page

2.4.4 Search based on URL

As discussed before, the crawling process will start from web sites of universities in Germany and Austria which can be obtained from Wikipedia [wika][wikb][wikc]. Swiss universities have been kept out to in order to not have to deal with the web site of universities in the Italian and French speaking part. In total, web sites of 444 different universities could be obtained. To enrich the set of start seeds, each URL were put into Bing Search and all resulting URLs which are sub domains of the respective query URL were collected. Merging the collected URLs with the set of start seeds gave 9485 URLs. Extending the set of seeds makes sure that sub domains which are not directly accessible from the main web site are registered too and will be visited.
Starting from this set of seeds, the crawler application gave 287012 URLs after a time of about one and a half hour.

visited web pages	287012
URL containing German text snippets	12708
URL containing English text snippets	2731

Table 2.10: Results of crawling based on URL

Filtering the URLs gives a set of web pages whose URL contain language identification text snippets 2.10. For English indexing text snippets, 2713 URLs could be found. For German indexing text snippets, 12708 URLs could be found. English indexing text snippets could be found in 2731 URLs.

Carrying out the replacement test for the URLs with German text snippets had to be aborted because it needs too much time. Also there were some URLs whose replacement still referred to the same web page instead of a Web page in English. Because of these facts, this approach has not been continued but an alternative way has been tried. This way is presented in the next chapter.

2.4.5 Search based on intern page references

The second approach to crawl the web sites is based on internal language related references. The crawling process has been carried out with a recursion depth of 10 and with the threshold for language detecting calculated above. The same set of seeds for crawling based on URL has been used for this approach. The seeds has been divided into 9 subsets of which 8 have 50 seeds and 1 has 44 seeds. For each of these subsets an own crawling process were started and were running for three days. Crawling the seeds resulted into 2473 pairs of bilingual web pages for which 146627 web pages were visited. 2.11

From the 2473 pairs a tenth - 247 pairs - is picked randomly to evaluate the correctness. The 247 pairs are passed over to a graphical web browser to verify manually or better to say visually whether a pair refers to each other by a language link. Of the samples 98% were correct. Wrong pairs refer to the main page instead of referring to the corresponding page in the other language. Also they happen to refer to a subpage instead of the actual page. 2.12

seed URLs	440
visited web pages	146627
found redirections	50663
found candidates	2473

Table 2.11: Results of crawling based on internal references

Correct pairs	242
Incorrect pairs	5
Total pairs	247
Correctness	0.979757

Table 2.12: Testing correctness of crawling output

2.4.6 Support-Vector-Machine classifier

The Support-Vector-Machine classifier were trained and tested on a parallel corpus. This corpus were translated manually from a record of a lecture hold at Karlsruhe Institute of Technology. For training 1500 lines were added by reordered alignments of these 1500 lines to a total of 6000 lines. The reordered alignments make sure to have negative examples within the training. The test set contains 420 correct lines and 1260 incorrect lines. The incorrect lines were obtained the same way as for the training set. Testing the trained classifier gave a precision of 95.15% and a recall of 93.33%. 2.13 Even though the results

positive training size	1500
negative training size	4500
positive test size	420
negative test size	1260
precision	0.9515
recall	0.9333
accuracy	0.9714

Table 2.13: Training and testing the SVM classifier

are good, the training and testing set might be different in nature from the content of web pages. Unfortunately there was no confirmed set of parallel corpus gathered from web pages.

2.4.7 Applying crawling results to sentence alignment and classifier

Taking the pairs from crawler based on internal references, the sentence alignment and the trained classifier were applied to get the corpus of mutual translation pairs. From those 2473 pairs of web pages, the sentence alignment process created 48326600 hypothetical alignments. An estimated limit of correct translation alignments can be calculated by considering the correctness of the cross-product alignment. It already contains mistakes of the sentence segmentation. Additionally not all web page pairs are correct. As demonstrated, about 2% are wrong paired. So we can assume that 50 of 2473 pairs might be wrong.

$$2473 \cdot (1 - 0.979757) = 50.060939 \tag{2.8}$$

Supposing that the average size of the cross product alignment is 19773 sentence pairs, we'll have presumably about 990000 wrong sentence pairs. Of course having a wrong pair of web pages, it does not mean that we will not find any correct translation pair. But to

keep the calculation simple, we do not consider this aspect.

$$50.060939 \cdot 19773 = 989854.947 \tag{2.9}$$

Finally we bring the correctness of the cross-product into calculation. Subtracting the wrong translation pairs and multiplicating it by the correctness of the alignment process gives an estimation of how many translation pairs actually might be correct. It is about 550000 pairs.

$$(48326600 - 989854.947) \cdot 0.0117 = 553839.917 \tag{2.10}$$

This is only a rough calculation and the reality might differ from it, since the big variances and other aspects have been kept out. On the other hand it offers an orientation of how well the classifier works on the gathered corpus.

Applying the corpus of hypothetical alignments to the classifier returns about 220000 translation pairs. This is about 0.5% of the original corpus. 2.14

	absolute	percentage
translations	217818	0.45%
non translations	48108782	99.54%

Table 2.14: Applying Support-Vector-Machine classifier to corpus

Considering the precision of the trained Support-Vector-Machine, about 5% of the 220000 pairs will be wrong, this is 10670 pairs. Inspecting the 220000 pairs shows some wrong translations. Some are because there are only two words on both side and only one of them is known. The classifier now assumes that the unknown words are mutual translations which they are not. To prevent such situations, the alignment has to predict better based on the document structure.

In a second step the rejected German corpus has been applied to a decompounder to make it easier for the classifier 2.15. So additional 0.06% could be filtered out by the classifier. It shows that compounded words which often appear in German are some difficulty for the classifier. The decompounding process was carried out as it is described in Herrmann et al.[HMNW].

	absolute	percentage
translations	31341	0.06%
non translations	48077441	99.93%

Table 2.15: applying Support-Vector-Machine classifier to decompounded rejected corpus

Inspecting the out coming files many entries appear more than once. Filtering out redundant entries, the file of correct translations keeps 22484 alignments whereas the file of rejected translations contains about $12 \cdot 10^6$ alignments. Inspecting the set of accepted translations again, some lines of dates and names appear which are not useful to have. In order to remove such lines, a length filter will be applied 2.16. If some entry has not the required minimum of words on both sides then it will be rejected by the length filter.

minimum length of words per entry	0	3	5	7	10
remaining entries	22484	13223	7331	6132	5051

Table 2.16: Accepted translations applied to length filter

These are considerably good results regarding the incorrectness of the cross product. The estimation based on the outcome before filtering out duplicates though shows some improvement can be done.

2.5 Outlook

As mentioned before, some improvement can be done. In the alignment process, we have taken a quite simple algorithm: just building the cross product between the sentences of two documents. Since the actual formation of a document is given, this information can be included in the alignment calculation. Resnik et al.[RS03] used an alignment algorithm based on a dynamic programming algorithm by Hunt and McIlroy. In exploiting the fact that bilingual pages have almost the same HTML structure, corresponding text segments will be aligned. This way, the precision of the alignment process can be increased dramatically even though the recall probably will decrease. The then aligned text segments have to be analyzed whether they could be divided again into smaller segments like sentences.

Another approach would be to look for smaller pieces of entities instead of sentences. Munteanu et al.[MM] suggested rather to take sentences as basic entity to identify parallel fragments within sentences. For most situations, it represents reality much better, since every translation also is an interpretation. So information will be dismissed, split or added. However, Munteanu et al. observed that taking fragments instead of sentences does not bring much improvement. This could be because of two points.
First one is that they rely heavily on lexicon data. It might be interesting to investigate if a different source potentially topic related helps to increase the fragment approach. Also one could try to reduce the dependency on the lexicon data and instead to use more data from the actual text. Position within a paragraph, proper names and phonetically close related words might help to identify parallel fragments as well as numbers and references. The second point is that the test corpus might be almost parallel on sentence level due to its nature. Only on the test corpus derived from the BBC web page, Munteanu et al. could see a significant improvement. It might be interesting to test it on two web pages which discuss the same topic but are not from the same company or web site. This would make sure that they have not been created by translating sentence by sentence and hence they will not be parallel at all. But since they reflect the same issue, they might contain parallel fragments which will not be typical expression of a language but of the topic.

We have evaluated only the approach. Putting it into action requires more engineering. The following section is discussing issues which should be considered.
Many can be done parallel. Since each crawling process is supposed to stay on its own web site, there is almost no mutual influence. But using one process per web site will not exhaust all potential possible, even though it might be attempting because of its simplicity. The collection of invalid file types can be shared. Also sharing threads might help to increase the efficiency because analyzing the web pages and archiving the results takes some time which can be done during waiting on a network response. Using more than one process thread for crawling one web site may increase efficiency too. This can lead to complications if the whole process of finding link references on web page and confirming such cross references should be merged into one step.
The next point of enhancing our approach, is about to merge finding link references on web pages and confirming these into one step. It will bring more handiness and better usability. Also it can decrease the usage of memory. Once a web page has been visited, all references to other web pages are registered. Any reference from another page back to the visited page which does not match the already registered references can be rejected. As mentioned before, the situation gets complicated if a web site should be crawled by more than one process thread. The steps of sharing the progress between the threads and synchronizing correctly information requires a sophisticated and a careful designed algorithm.
Trying different parameters for the crawling process we've encountered the problem of memory exhaustion. Since all data are held in the main memory this can be solved only by taking use of data bases. For our application we had to split up the collection of

seeds into smaller heaps and to limit the deepness of recursion. Having a data base would allow to push all seeds into one process and to increase the limit or perhaps to lift the limit. Going deeper into a web site might unveil more bilingual web pages with more essential text content. Bilingual web pages directly referred by the home page are often contact, directory and overview pages. These usually do not provide much text not to say interesting text.
The last suggestion is to build a system which can run easily for longer time and to research visited web pages. For web sites which get updated quite frequently like news portals, new content is provided on the same web page after a certain amount of time. Searching such web sites, a system needs to run for longer and to detect which page gets updated and which not. For university web pages, this suggestion probably will not help to improve because of the slow income of new material.

Since crawling takes some time to find enough content, it makes sense to build up a data base so that getting bilingual web pages of two arbitrary languages can be handled quickly. It would be similar to an ordinary Web search engine but specialized to deliver only bilingual or multiple-lingual web pages. Rather than putting a word into the search engine, the web site - that is to say the URL of the home page or simply the Web server - has to put into the search engine in order to obtain all bilingual web pages within the web site. Crawler processes in the background would help to keep the data base up to date. Such service would provide good support for constructing a machine translation system as well as make sure that effort will not be put into the same work twice.

3. Translation for oov

3.1 Motivation

At Karlsruhe Institute for Technology (Germany) and Carnegie Mellon University (USA) the project Lecture Translator was launched to build a reliable simultaneous translation system for lectures[Lec]. The aim is that students and everyone who attends a lecture get a simultaneous translation on what the docent is saying. The system consists on two parts: speech recognition and the actual translation done by a machine.
In order to train or adopt the system of the machine translation, a lot of data about the content of the lecture is required. The easiest way of getting such data is to analyze lecture slides, exercise sheets and so on.

An important part in training a good translation system, is to predict which words or phrases and their translations have to be known in order to achieve a good understanding of the lecture. One way is to collect a set of vocabulary which covers the domain of what the lecture is about. Therefore methods which have been discussed in the chapter before, can be used to obtain such a set.
Another way to increase the understanding of translations is to predict specific words which are very likely to appear and to find their translations. Some lecture about Economics usually has its technical expressions and these differ from the typical vocabulary set of a lecture about law. But we can go even further and say that a lecture about the Gross National Product has its own typical expressions and they are different to expressions of a lecture about Accounting. So finding a translation for such typical expressions should help to increase the understanding.
It also can be extended to words whose translation is already known. Often words have different meaning and hence different translation depending in which context they appear. The German word ”Körper” usually means in English ”body”. The meaning remains in Biology where it refers to the physical appearance of a human being or an animal. In different disciplines like Geometry or Physics, the meaning of the word ”Körper” has been abstracted to a general term for any limited three dimensional entity. So depending on the context ”Körper” has to be translated to ”body” (Biology) or ”field” (Algebra) or ”object” (Physics).
To find a translation of some specific expression we need a large corpus in order to extract the wanted translation. The Web seems quite useful, because it covers a wide range of topics, it contains information in different languages and it is machine readable and hence does not require any manual prework.

In this chapter we going to try to find translations for words we have not known any translation so far. To do so we want to take advantage of the web and his opulence of text documents.

3.2 Related work

Some effort has been done in the area of Cross-Language Information Retrieval based on Web search engine. Works about two different ways will be discussed and evaluated. The first way is about bilingual content and second one is about deducting a translation from the context.

3.2.1 Bilingual Context

Sometimes documents contain small information of some translation. Often relevant words are annotated with their translation. In the following section different works are presented which try to exploit such documents to disambiguate between translations (3.2.1.1 3.2.1.3 3.2.1.2) and to find new translations(3.2.1.4).

3.2.1.1 Maeda et al.

Maeda et al.[MSYU] proposed a disambiguation method for dictionary-based query translation. In their work they focus on Japanese-English translation. The source phrase in Japanese will be segmented into words based on a morphological analyzer. Several combinations are possible since Japanese does not know word boundaries. Each combination of word sequences will be translated into English. If more than one translation are found the longest is taken. If overlapping matches are found, all of them will be taken into consideration.

To disambiguate between multiple possible translations, co-occurrence tendency measures are applied in order to find out which word sequence is most likely to belong together. To operate on a large corpus Maeda et al. use Web search engines. Special commands help to find the correct results. Using the AND operator will deliver results with both query words in it. So searching for "distributed AND network" will give web pages which contains both "distributed" and "network" and hence can be used to find out the co-occurrence frequency of both words. Using the OR operator helps to get web pages which contains at least one of the words. To obtain web pages which contain one of the words but not the other can be achieved by the operator AND NOT. Applying these logical operators brings the necessary informations to calculate the co-occurrence measures.

Maeda et al. used four co-occurrence measures. The first co-occurrence measure is Mutual Information. It gives the quantity of mutual dependency. The second co-occurrence measure is Dice Coefficient. It calculates the similarity between two sets. Maeda et al. adopted the Dice Coefficient to improve the accuracy. This adaptation is done by adding a weight which is based on the co-occurrence frequency. The third co-occurrence measure is Log Likelihood Ratio. Log Likelihood Ratio tests a hypothesis. The hypothesis used here is the null hypothesis, expecting that each word is independent and does not appear more often with one word than another word. Obviously this is not correct. Opposing this null hypothesis to the reality gives a degree to which certain words appear more often together than independent words. The last co-occurrence measure is Chi Square Test. Chi Square Test also tests variances of measurements to a given hypothesis. It assumes a χ^2 distribution.

Testing the different co-occurrence measures, Maeda et al. couldn't observe any significant differences. Differences of the average precisions were lower than 0.1%. For their disambiguation system they could acquire a coverage of 97% of manual translation case in terms of the average precision.

Different to our case is that Maeda et al. are not interested in finding a way to obtain a translation for an out-of-vocabulary term. Besides that Japanese and German differ in various way which might eliminate some challenges Maeda et al. had to face. But on the other side it might create new ones.

3.2.1.2 Cheng et al.

Cheng et al.[CTC+] proposed a translation method for unknown queries using web search engines. Their approach rests on the fact that the web contains a lot of text documents which are designed for different languages. In such documents there is one main language and a so called auxiliary language, often English as it is the most spoken language internationally.

Often documents being a mix of two languages have parallel or comparable terms in both languages. Terms in both languages can be navigation simplifications for users who haven't got sufficient skills of the main language, or they can be the most relevant terms in a document. Also anchors sometimes have been appended their English translation. The phenomena of translations in the same document appears in highly technical or scientific documents too where the technical terms origins from a different language than the main language.

Cheng et al. carried out some experience to find out how often such translation pairs could be gathered from the Web. From a real search engine log they got 430 popular English query terms and translated them manually into Chinese. They also picked randomly 100 query terms from the top 19,124 query terms in the search engine log and translated them into Chinese. Putting the queries into Google Search gave them following result. More than 95% of the translations of the popular query terms could be found in the top 30 to 40 summary snippets given by Google Search. Also the results of Google Search could cover about 70% of the translations of the set of randomly selected query terms.

To get translations out of the summary Cheng et al. perceived two issues. The first one is to find terms with correct lexical boundaries and to minimize the amount of terms with incorrect lexical boundaries in order to keep the noisy information down. Therefore Cheng et al. propose a new association measure. It takes use of symmetric conditional probability and context dependency. The symmetric conditional probability gives information about the cohesion within an n-gram. The context dependency helps to clarify how much the appearance of a n-gram is dependent on a certain phrase within the n-gram itself. Cheng et al. also mention that their association measure can be combined with a local maxima algorithm or with a PAT-Tree.

The second issue is to actually find correct translations or translations which are correct in terms of semantically close. Translations should be found without needing too many search results and wasting too much time for waiting on a web request. The difficult part is to calculate the similarity of each translation candidate and the query term only having the result of the Web search engine as additional information. Cheng et al. suggest two ways of doing so. The first one is based on estimating a co-occurrence score. Cheng et al. use the χ^2 test and take the necessary parameters from the page count of querying a Web search engine. Thereby logical operators will be applied as discussed above. The second way is to build up a so called context feature vector for each candidate term. Similarity then is determined by the distance between two context feature vectors. The features are contextual terms constituting the search result pages. To calculate the similarity between the source query term and a candidate term cosine similarity can be used. Cheng et al. observed that the χ^2 method performs better for high frequency terms instead of low frequency terms. The Context Vector Method gives better results for low frequency terms compared to the χ^2 method. However the Context Vector Method depends more on the quality of the search results which makes it more unstable. To get the best out of both methods Cheng et al. propose a combined approach which basically is a linear combination weighting scheme.

Cheng et al. experienced for English to Chinese a top-1 inclusion rate of 46 percent using the web to find the translation whereas using a domain specific parallel corpus

instead resulted in a inclusion rate of at most 2.5 percent. They also carried out some tests for English to Japanese (top-1 inclusion rate at 35 percent) and English to Korean (top-1 inclusion rate at 32 percent). This approach seems to be feasible for queries of technical terms and proper names. Also it might helpful for language pairs which use different characters such as English (Latin Alphabet) and Japanese(Hiragana, Katakana) since every proper name and technical term has to be transcripted.

3.2.1.3 Zhang et al.

Trying to improve the approach of Cheng et al. and various other suggestions, Zhang et al. [ZV] proposed an extension to improve the disambiguation process. In their work they regarded Chinese-English translation. The disambiguation process involves four steps. These are detecting Chinese out-of-vocabulary terms, extracting appropriate text from the Web, making calculations of co-occurrence statistics about the extracted texts and after all selecting a translation out of the extracted texts.

The first stage is detecting out-of-vocabulary terms. Since the source language is Chinese and Chinese does not know any word boundaries it is essential to decide which symbols belong together and where a new word is starting. If the meaning of all symbols and their translation is known, this is rather an easy task and can be done the way Maeda et al.[MSYU] suggested. But this can only be applied directly to out-of-vocabulary terms. So these out-of-vocabulary terms have to be known in beforehand. However, the challenge lies in identifying the out-of-vocabulary terms first. Zhang et al. therefor propose a Hidden Markov Model which estimates such unknown out-of-vocabulary terms. A Hidden Markov Model perfectly models the necessary approach for such a challenge. The actual intention of the Hidden Markov Model used by Zhang et al. was to have a proper disambiguation technique. Since the Hidden Markov Model has to calculate the correlation probability of the Chinese symbols, it can also be used to detect likely out-of-vocabulary terms. If the correlation probability is lower than a certain threshold probability, an out-of-vocabulary term can be assumed.

The second step is about extracting good text snippets from the Web in order to find a translation for the out-of-vocabulary term. Zhang et al. use Google search to get a corpus consisting of result descriptions. The query thereby is the Chinese out-of-vocabulary term itself. Thanks to Google's good work the resulting corpus can be considered to be good quality and hence keeps noisy information low. The resulting descriptions (titles and descriptions) then will be searched for substrings which contains both the Chinese query term and a following or preceding English phrase. Because Chinese and English use different alphabets this can be done quite easily and reliably.

In the third step co-occurrence information will be gathered from the text snippets which were collected before. For each English phrase which appears before or behind a substring of the origin Chinese query, the frequency, all associated Chinese sequences and their lengths as well as the co-occurrence frequencies will be gathered.

The last step eventually results in a translation phrase for the out-of-vocabulary term. First the longest Chinese substring is searched for. Then the English term which occurs the most often with the longest Chinese substring will be registered as translation for this Chinese substring. After that the most frequent English term is searched for and the Chinese substring which appears the most often with the English term. If both of them are different to the first pair of English-Chinese translation, then they will be registered as translation too. Zhang et al. maintain that for most cases two translation pairs were enough to have a translation for the out-of-vocabulary term. Often only one translation pair was given anyway.

For translation from Chinese to English Zhang et al. experienced encouraging results even though they operated on a small set of data. In 50 queries 8 were found to contain Chinese out-of-vocabulary terms and for 7 out of 8 correct translation could be found out. For another set of 60 queries 25 out-of-vocabulary terms have been found and for 18 correct translations(72%) could be delivered. For translation from English to Chinese four different experimental runs were carried out. In the first run all Chinese equivalents of English out-of-vocabulary terms were kept out to see how well the disambiguation technique is doing without being troubled with out-of-vocabulary problems. The second run was done without any English out-of-vocabulary terms. In the third run the Chinese equivalents of the English out-of-vocabulary terms were added and in the last run the English out-of-vocabulary terms were included. Without the English out-of-vocabulary terms 86.7% of a monolingual result could be achieved. Adding the English out-of-vocabulary terms resulted in 77.1% of the monolingual result. Even though only appropriate translations for 88% of English out-of-vocabulary terms could be found it is a dramatic improvement considering that other approaches only achieve 72% and need manual inspection.

3.2.1.4 Huang et al.

Huang et al.[HZV] propose an approach to mine the Web for translations of specific short terms. The starting problem very similar to the problem we want to solve is to find translations of out-of-vocabulary terms out from the Web. The motivation is that nouns and technical terms are one of the most information-bearing linguistic structures. Finding a translation for such out-of-vocabulary terms will help to augment the quality of language based systems. Unfortunately proper nouns or terms might be context dependent and therefore will not be covered by a regular bilingual lexicon. Crawling the web might be unrewarding. Huang et al. suggest a way to look for translations of specific terms.

The same phenomena Cheng et al. experienced is the base for their approach. A lot of web pages contain bilingual information in a way that certain terms and their translation appear close to one another. Also Huang et al. observed that the appearance of such phenomena are often accompanied by translation pairs which are semantically related to the first term. So their approach contains four stages.

First semantically related terms in the source language have to be found. Such terms must follow certain criteria. They should occur very often with the query term. Huang et al. suggest to take the most frequent terms. Also the terms have to be reliably translatable. So they must be covered by the used bilingual lexicon. But if too many translations are known for a specific term, the term will not be used since the chances are higher to pick the wrong translation than the appropriate translation. The last criterion is that the term should be translated into a noun or a phrase of nouns. It helps to keep the focus on noun translations. The top words which satisfy the criteria and have the highest frequency will be selected to build new queries. For each selected word its translation and the original query term builds a new query term.

After sending the queries to a Web search engine like Google, the returned web page snippets will be preprocessed. Underlying HTML format will be removed. Special HTML encoding will be transformed into their respective character. Then a word segmentation is applied. This is necessary because Huang et al. regarded Chinese-English translation and as discussed before Chinese does not know word boundaries. Punctuation marks will be replaced by a special phrase separator character. All non-query words which are in the source language will be replaced by placeholder mark. Counting the placeholder marks between a phrase in the target language and the query term gives the word distance.

For all phrases in the target language features will be calculated. Since most of the out-of-vocabulary terms are names and names are phonetically transliterated into Chinese,

a similarity score based on a transliteration model will be estimated. The next feature reflects the semantic equivalence between a Chinese term and an English candidate. To compute the translation probability, the IBM model-1 will be used. Based on translation probability for both direction the NE (???) translation cost is calculated which represents the semantic equivalence. The last feature captures how often and how close the English candidate term appears together with the Chinese out-of-vocabulary term. If an English candidate term appears more often and/or closer to the Chinese out-of-vocabulary term then it is more likely to be an appropriate translation. Based on these features a translation will be selected amongst the candidate terms.

To test their approach Huang et al. compared it to other known strategies. First translation were tried to be found by searching any web pages which contain the query term. Then translations were tried to be found by searching only English web pages which contain the query term. For searching any web pages a maximum inclusion rate of 85.8% could be achieved. Searching only English web pages resulted an inclusion rate of 89.7% using 32 mixed-language snippets and 95.2% using 165 snippets. Using an average of 165 snippets per query gave Huang et al. a top-1 translation accuracy of 80%. Considering the top-5 results brought an accuracy of 90%.

Even though the approach describes a solution for a problem very similar to our ones, some differences remain. Chinese needs a word segmentation process. This will not be necessary for German. Another difference is that Chinese and English can be easily distinguished solely on the syntax, to be more precise the used characters. For German-English it will not be the case and it will need some different techniques to determine whether a word is German or English.

3.2.2 Context deduction

Starting from the out-of-vocabulary word, context related words in the same language are collected. These words then will be translated into the target language. From the context of these translated words, the wanted translation will be deducted. Here the work of Rapp will be presented in detail.

3.2.2.1 Rapp

An approach to find new translations for German-English were proposed by Rapp[Rap]. His intention was to extract translation out of comparable or - worser - unrelated monolingual texts from both source and target languages. By imitating professional translators and interpreters who read texts issuing a certain field in both languages in order to prepare terminology in this field, translation pairs should be extracted out of the monolingual documents. Since a machine is not able to semantically understand a text contrary translators and interpreters, statistical techniques will be applied instead.

To acquire information about semantic relation between words in texts of the first language and words in texts of the second language, the fact that translation pairs appear quite often with other translation pairs which are semantically close related will be exploited. It is the same phenomena which has been discussed several times before. For this undertaking a monolingual corpus for both languages is required. Furthermore a possibly small bilingual dictionary is needed. Rapp aims to expand this dictionary through the process of his approach. The corpora Rapp is using are editions of the German newspaper "Frankfurter Allgemeine Zeitung"(1993 to 1996) with about 125 million words and of the English newspaper "The Guardian"(1990 to 1994) with roughly 163 million words. These corpora are preprocessed in order to reduce disk space and processing time. The pre-process involves removing all function words and lemmatizing. Function words are found by comparing to

a special list of function words in German (600 entries) and in English (200 entries) as well as by calculating the word frequency derived from the used corpora. Rapp assumes only a lose of little information since function words usually are for syntactic reasons and therefore occur very often. Also they are generally highly ambiguous which makes it more difficult to find an appropriate translation. Lemmatizing the corpora accommodates the fact that German is highly inflectional. So more disk space can be saved and the frequency of words can be calculated more effectively. For the bilingual dictionary a lexicon of about 16000 entries were used.

Rather than using fixed window sizes for counting word co-occurrences, Rapp decided to count the co-occurrences considering the distance. So assuming a window size of two the frequency of words being two words ahead, being one word ahead, being one word beneath and being two words beneath will be counted separately. These frequencies then make up a co-occurrence vector. The best working window size Rapp could detect were three based on preliminary experiments. To emphasize more the association between two words than the co-occurrence the relation between observed co-occurrences and expected co-occurrences will be used. To represent this relation Rapp used the Log-Likelihood-Ratio which worked out to be best for sparse data. Based on the Log-Likelihood-Ratio the association vector is computed.

The association vector is used to determine likely related words in the source language, then to get their translations via the bilingual dictionary and then to predict the translation based on which words are most likely related to the translations. To get a relation score between words a similarity measure is applied. Rapp decided to use the city-block metric which calculates the similarity between vectors A and B by adding the absolute differences of the corresponding vector positions. $s = \sum_{i=1}^{n} |A_i - B_i|$ Testing his approach Rapp gained an accuracy of 72% for manually checking the first translation candidate. Considering the top 10 of translation candidates gave an accuracy of 89%. Rapp made out various problems in his approach. One of the problems is a strong dependency of the used corpora. An expected translation for the German word ”Kohl” is ”cabbage”. Unfortunately the newspaper contained a considerable part about politics and hence delivered ”Major”, ”Kohl”, ”Thatcher”, ”Gorbachev” which are names of political leaders during the time the editions are from. Another problem is the strong morphology of the German language. The German word ”weiß” were translated by ”know” rather than by ”white”. The word ”weiß” is an inflected form of the verb ”wissen”, in English ”to know”. The word ”weiß” has not been removed by the lemmanizing process and so distorted the out coming result.

Like Zhang et al. does Rapp offers a way which is quite similar to ours. Instead of having collected a corpus beforehand, we want to take advantage of the web which has some advantages like the size, offering text for more domains and faster searching. Another difference is that our domain already is limited to academical and scientific phrases.

3.3 Methodology

There are different ways to get text information from the internet. In the following we want to explore these different ways and evaluate the results. To find translations we are going to use the Web as a corpus. The Bing Search API will be used to effectively search the Web.

Using a web search engine does not cover the whole web. It even does not cover a fraction of the web. According to Bergman[Ber] only 0.03 % are indexed by search engines. Only a rough estimation helps to determine the size of the web. Currently (February 2012) the size of the indexed web is considered to be between 8 billion and 55 billion web pages[wor] Google itself announced to have encountered more than 1 trillion unique URLs[goo]. Assuming the web is growing exponentially[**?**] the majority of the web cannot be reached by any web search engine. However the indexed part still offers a rich source of information and web search engines give immediate access to it. Also there is no good alternative. Building an own crawler will face the same problems and surely will not be better than the already existing search engines of Google and Microsoft. Another way of finding information in such a complex and big network like the Web has yet to be invented.

Our aim is to extract translations from the web. To be more specific, we want to extract the translations out of text documents from the web. The set of all phrases for which we want to find some translation are obtained by filtering the text documents(presentation papers, homework papers, exercise sheets, ..) of a specific lecture for unknown vocabulary. Due to their nature these phrases often consist only of one word and they are such special terms so that they are not covered by some general dictionary. To get the most out of the web in a short amount of time, a web search engine appears to be the best way.
Having the set of out-of-vocabulary phrases, we build specific queries for each phrase, put it into a web search engine and extract possible translations out of the search result.

3.3.1 Extracting text out of documents

Basically the documents we want to search for out-of-vocabulary terms are lecture related. If docents use presentation assistance such as Power Point or Latex Beamer, the text inside the files can be used to find out-of-vocabulary terms. In the same way exercise sheets can be used. A lot of docents offer their presentation slides as PDF formated files. It is considered as the most undependable rich document format and therefore is supported by most of the used operating systems. A lot of tools for various platforms exist to handle PDF formated files in almost every way. Another file format widely used for presentation slides is Microsoft Power Point. Since its format is changing constantly it is not so easy to find tools which are up-to-date to the newest file format on Non-Microsoft platforms. On Linux there a several tools to handle the conversion of different text documents.

- pdftotext converts from PDF to plain text
- unoconv converts any document from and to any OpenOffice supported format
- catdoc converts Microsoft Word file into plain text
- catppt converts Microsoft Power Point file into plain text
- Apache Tika converts from over 1200 file formats like PDF, any Microsoft Office format and ODF to plain text
- wvare converts from Microsoft Word format into other formats like PS, PDF, etc.

To give some picture about the differences, we converted a set of presentation slides of a medicine lecture into plain text. The slides are all in German and have a lot of technical

terms. The tools pdftotext and Apache Tika have been used for the conversion. Using isutf8 we made sure that the conversion were made correctly. Then we removed all space, control, punctuation and digit characters to tokenize it into words 3.1. Using pdftotext we encountered less words than using Apache Tika, but more words with length greater 3. The set of differences between both word sets contains words which are actually two words but have been accidentally put together. The tool pdftotext seems to work better even though the difference is quite small.
For both tools however some issue occurred converting German documents. Due to font-technique reasons, German umlaut marks will not be converted correctly. As it appears for many PDF documents which were built by LaTeXthe umlaut marks will not be rendered by the equivalent character but by the basic character and the overlying dots separately. So the *ä* will be rendered as a *a* and at the end of the line ¨ is appended with the notation to draw it above the *a*. To prevent this systematic conversion error, the converted documents will be corrected manually.

	pdftotext	Apache Tika
words	3373	3706
unique words	1361	1376
words with length > 3	2436	2400
German unique out-of-vocabulary words	595	610

Table 3.1: Comparing the tools pdftotext and Apache Tika

Besides orthographic mistakes a document might not be in the source language or some part of it might not be in the source language. If these parts or documents have been written in the target language and no translation could be found, it will be taken as it is and hence does not cause any error. Since it might happen that we find by mistake some translation for a phrase which is in the target language, filtering the document against a lexicon of the target language might help to prevent it. Another approach is to determine the language of the document beforehand and only let pass the ones which are in the source language. In our work we make sure to use only documents which are basically in German.

3.3.2 Out-Of-Vocabulary Detection

After converting the documents into plain text, all unknown words have to get identified. This is done by matching all words against a collection of words for which translations are already known. If no translation is known, then the Web will be searched to find some.

3.3.2.1 Single word query

Probably the simplest method to detect a translation for an out-of-vocabulary term is combine it together with a flag indicating to look out only for web pages in the language the translation should be in and then sending the query to a Web search engine. The resulting description of the different web pages then will be searched for translation candidates. Even though it is simple as it does not need any calculation, it has several disadvantages.

Because we look out only for one word, we loose information about the context. It makes it more difficult to find translation candidates. This can be explained by the fact that the less information we have to find something we do not know, the fewer is the chance to find it. In terms of information theory we can argue that putting more information into the system makes the entropy low. Putting only one word into the system means a higher entropy and hence it makes it more difficult to find the right translation. If the

word - independent of the context - has several translations, it also makes it more difficult to disambiguate it.

To test how well this simple method performs all entries of a bilingual dictionary which only have one word have been put into the bing search. We built n-grams(up to 5-gram) of the top 50 results for each one-word entry and then counted how often a possible translation of the specific word appeared as an n-gram 3.2. In total we had 100785 one-word entries. The dictionary[Dic] is topic independent. For the Bing search we used the meta operator "language:en" to indicate that only English sites should be given out as result. As the figures show there is a substantial amount of web pages which contains English translations of German words but it would be difficult to find the correct translation candidate in the result and to determine whether there is a translation at all. We also can see that one German word may be translated into a phrase with more than one English word. Taking phrases with up to three words into account almost doubles the amount of finding a candidate. But considering phrases with more than three words does not make much difference any more.

n-gram	found translation	not found translation	found translation in percentage
1	26092	74693	26
1-2	39360	61425	40
1-3	41386	59399	41
1-4	41792	58993	41
1-5	41869	58916	42

Table 3.2: Finding translation depending on size of search window (n-gram)

3.3.2.2 Context taking detection

To overcome the lack of information about the out-of-vocabulary word, a better description about what to look for (meaning the translation we yet do not know) has to be created. This description can be derived from two areas. The first one is the actual context where the out-of-vocabulary word has been found. If the word is part of a quote or a set expression, it might to help to find the correct translation. In case the actual context only consists of regular prepositions, conjunctions or other function words, this method probably will fail.

The second area is the semantic word field. Putting related words and their translation into the web search, helps to clarify what we are looking for. Going this way to find translation is quite common[MSYU][Rap]. To find semantically related words the Web will be searched.
At first related words have to be found in German. Their translation then will be put into a Web search engine to get all words of their context. Those words then will be scored according to their co-occurrence, hoping that the appropriate translation appears often in the context of the translation of the related words. Also the original out-of-vocabulary word can be put into the latter query to obtain web pages which might contain bilingual information.

3.3.3 Co-occurrences measures

In order to compute the co-occurrence of two words A and B, four parameters are required:

- f_{AB}: How often do both words A and B appear together.
- $f_{A\neg B}$: How often appears A without B.
- $f_{\neg AB}$: How often appears B without A.

- $f_{\neg A \neg B}$: How often appears neither A nor B.

These parameters can be calculated by using logical operator like "AND" and "NOT". The count of found web pages then is taken from the meta data. The overall frequency of A then is f_A and the one of B is f_B.

$$f_A = f_{AB} + f_{A\neg B} \tag{3.1}$$

$$f_B = f_{AB} + f_{\neg AB} \tag{3.2}$$

The overall frequency of web pages which have not A in it is $f_{\neg A}$. The overall frequency of web pages which do not contain B is $f_{\neg B}$.

$$f_{\neg A} = f_{\neg AB} + f_{\neg A \neg B} \tag{3.3}$$

$$f_{\neg B} = f_{A\neg B} + f_{\neg A \neg B} \tag{3.4}$$

To normalize all results the sum of all web pages f_Σ is needed.

$$f_\Sigma = f_{AB} + f_{A\neg B} + f_{\neg AB} + f_{\neg A \neg B} \tag{3.5}$$

The co-occurrence measures we are using are:

- Log Likelihood Ratio
- Mutual Information
- Modified Dice Coefficient
- χ^2 Test

3.3.3.1 Log Likelihood Ratio

The first measure, Log-Likelihood Ratio, describes how much the observed co-occurrence differs from the expected one. Expecting that every word is equally distributed and independent from each other, the observed co-occurrence tells us if word pairs are more probable to appear together or not. Rapp[Rap] uses this measure in his approach argumenting that it is theoretically well justified and more appropriate for sparse data than the χ^2 Test.
The formula for log-likelihood ratio is the same Rapp is using.

$$C_{\text{LLR}} = f_{AB} \log \frac{f_{AB} f_\Sigma}{f_A f_B} + f_{A\neg B} \log \frac{f_{A\neg B} f_\Sigma}{f_A f_{\neg A}} + f_{\neg AB} \log \frac{f_{\neg AB} f_\Sigma}{f_{\neg A} f_B} + f_{\neg A \neg B} \log \frac{f_{\neg A \neg B} f_\Sigma}{f_{\neg A} f_{\neg B}} \tag{3.6}$$

3.3.3.2 Mutual Information

Mutual Information is used to determine the mutual dependency between two random variables by using the joint probability $P(A, B)$ and the marginal probabilities $P(A)$ and $P(B)$. Applying to the scenario of finding related words, the probability outcomes will be calculated using the frequency of the specific cases and the total amount of web pages.

$$C_{\text{MI}} = \log \frac{P(A,B)}{P(A) \cdot P(B)} = \log \frac{\frac{f_{AB}}{f_\Sigma}}{\frac{f_A}{f_\Sigma} \cdot \frac{f_B}{f_\Sigma}} = \log \frac{f_{AB} f_\Sigma}{f_A f_B} \tag{3.7}$$

3.3.3.3 Modified Dice Coefficient

Dice Coefficient is used to calculate the similarity of two samples. In terms of finding related words, it can be understood as how many web pages two words are sharing. Maeda et al.[MSYU] suggest to use a modified version to give the frequency of both words occurring more weight.

$$C_{\text{MDC}} = (\log f_{AB}) \frac{2 \cdot f_{AB}}{f_A + f_B} \tag{3.8}$$

3.3.3.4 χ^2 Test

χ^2 test is a measure to compare the actual observation to the null hypothesis. The null hypothesis assumes a χ^2 distribution for both random variables A and B. χ^2 test gives a hint how dependent A and B are from each other. We are using the formula by Maeda et al. [MSYU] which includes Yate's correction for frequencies smaller than 5.

$$C_{\text{CHI}} = \begin{cases} \frac{f_\Sigma \left(|f_{AB} f_{\neg A \neg B} - f_{A \neg B} f_{\neg AB}| - \frac{f_\Sigma}{2} \right)^2}{f_A f_{\neg A} f_B f_{\neg B}} & \text{if } \min(f_{AB}, f_{\neg AB}, f_{A\neg B}, f_{\neg A\neg B}) < 5 \\ \frac{f_\Sigma (f_{AB} f_{\neg A \neg B} - f_{A \neg B} f_{\neg AB})^2}{f_A f_{\neg A} f_B f_{\neg B}} & \text{otherwise} \end{cases} \tag{3.9}$$

3.3.4 Finding related words

The first step is to find related semantic words of the out-of-vocabulary word. Therefore the out-of-vocabulary word is put into a query combined with a flag to look out only for German web pages. Then the returning descriptions are taken as the context. For each description text monograms, bigrams and trigrams are built and all grams which contain the out-of-vocabulary word are filtered out. For each of the grams the co-occurrence will be calculated. As described in the section about the different co-occurrence measurements, an estimation of how often two words appear or do not is taken from the meta data of the Web search engine. These figures forms then a score for the co-occurrence. Taking the top results should deliver related words.

An important part is to have translations for the related words found by the Web search

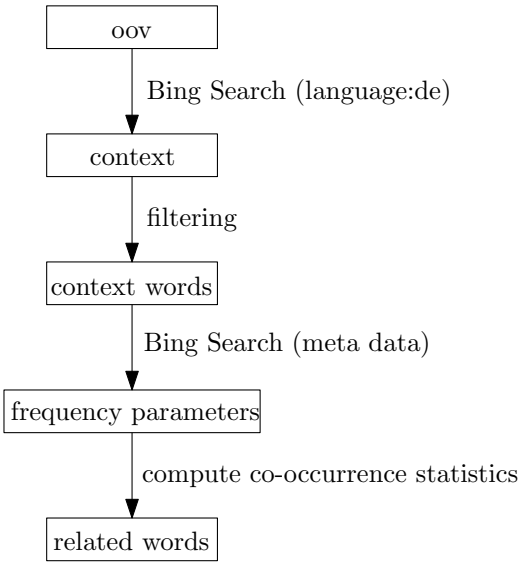

Figure 3.1: From Out-Of-Vocabulary word to related words

engine. Having a lexicon, the collection of related words can be filtered on letting pass only words known by the lexicon. It is then critical to have a lexicon which provides many words which are in the context of the out-of-vocabulary word. If not, it can happen that helpful results are filtered out and only low scored words are taken.

3.3.5 Extracting translation

Having a set of related words in English, it is now time to find the wanted translation. Basically the idea is that words which appear often together in German, they appear also often together in English.

First a set of context words is gathered. Every related words are put into a Web search

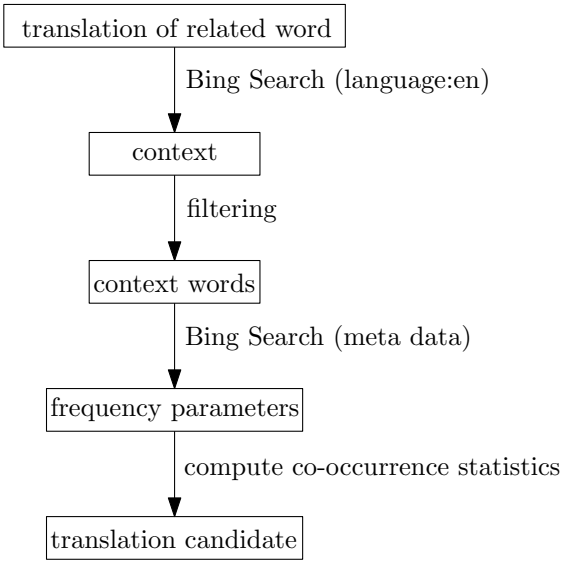

Figure 3.2: From Translations to Candidate

engine adding a flag which indicate to look out only for English web pages. From the returning description texts, monograms, bigrams and trigrams are built and all grams which contain the related word are filtered out. Then co-occurrence measurements between the grams and the related word are calculated. At the end for each related word all words within its context are scored. To get one score for a word which appears in the context of more than one related word, all its scores have to be merged in a certain way. In our approach, three different merging procedures are tested:

- taking maximum score
- calculating geometric mean
- calculating arithmetic mean

3.4 Test

In this chapter, out-of-vocabulary words will be tested on which kind they are, how successful the co-occurrence approach is to obtain related words and how well translations can be found based on translations of related words. Since the tests involve a lot of manual inspection and evaluation only a case study can be done based on documents of one lecture.
Out-of-vocabulary words make about 62% of words in all documents which display the nature of these documents and how much it is different to regular text. Related words can be found, but results for our approach are humble. The unreliable step is finding the translation based on related words in English. We use a simple approach and it shows that this way it needs a lot of rethinking because the figures are quite low.

3.4.1 Out-Of-Vocabulary words in documents

To get some inside about the nature of the documents which are taken to find out-of-vocabulary words, a set of documents will be examined. Therefore all documents related to a lecture hold at Karlsruhe University is collected. The lecture is about different paradigms of programming languages with a focus of functional programming languages. Documents are taken from exercise sheets, presentation slides and lecture transcriptions. Converted by *pdftotext* all documents made a text size of about 500 KB. The known Umlaut error in the conversion is corrected manually. Then all words were matched against a set of known words. The set of known words were obtained from the EPPS, NC, BTEC and TED corpora. Results **??** show that in total 20065 unknown words are found. Removing duplicates 3677 words remain. The out-of-vocabulary words in total make 57% and considering only unique words, the out-of-vocabulary make 62%. Surprisingly, the

Words in total	35250
Unique words	5893
Out-Of-Vocabulary words in total	20065
Unique Out-Of-Vocabulary words	3677
Unique Out-Of-Vocabulary words after decompounding	2986

Table 3.3: Out-Of-Vocabulary words in lecture documents

fraction of unknown words is quite high. It shows the very specialised vocabulary of lecture documents. There are not so many regular text parts which usually contains a lot of function words. Also technical terms occur extremely tightly in the documents like ”Ableitungsregeln” or ”Datenbankmanipulation”. Additionally German grammar allows to compound words to create new ones. Inspecting the Out-Of-Vocabulary words unveils that many compound words are in it like ”Folgekonfiguration” or ”Funktionstypoperator”. Running a decompounder application with the list of Out-Of-Vocabulary words, the list can be reduced to 2986. Inspecting again the Out-Of-Vocabulary words, however, it shows that the decompounder application leaves many compounded words untouched.

Next samples of the out-of-vocabulary words will be examined of which kind they are 3.4. They get tested on whether they are German or English, a technical term or not recognizable at all. The first three categories will are divided into correct and incorrect spelling. The test is based on 480 randomly selected out-of-vocabulary words. Technical terms are usually in English, but it is more appropriate to use them than the German translation. The high fraction of English words is probably due to the nature of the lecture which is about programming languages. Code examples are usually completely in English. Also they contain many function words, arbitrary abbreviations and compounding of words. Filtering these code examples out probably helps to focus on out-of-vocabulary words

Language class	Correct spelling	Incorrect spelling	Total
German	158	60	218
English	156	57	213
Technical term	13	0	13
Unknown	-	-	36

Table 3.4: Distribution of 480 randomly selected Out-Of-Vocabulary words

which have more semantic value and whose translations increases the understanding. Incorrectly spelled words may come from mistakes caused by the author or done by the PDF to text conversion application.

3.4.2 Finding related words in German

From the set of out-of-vocabulary words retrieved by the way discussed above, 300 words are selected randomly and labeled according to the categories described above. From the category of correctly spelled German words 20 ones are taken out by random to be used for further tests. For each of the 20 words semantically related words are searched and scored according to their co-occurrence measures. The average correctness of the top-1 and top-5 results are calculated considering every word as well as considering only out-of-vocabulary semantic words. As shown in **??**, for top-1 results the measure Mutual Information performs the best with 73%. Also did the measure Mutual Information perform best in the top-5 results category with about 70%.

Measure	Average of finding related words	Average of finding related words considering only semantic words
LogLikelihood Ratio (top-1)	0.58	0.69
LogLikelihood Ratio (top-5)	2.65	2.92
Mutual Information (top-1)	0.73	0.73
Mutual Information (top-5)	3.3	3.48
Weighted Mutual Information (top-1)	0.42	0.42
Weighted Mutual Information (top-5)	1.75	1.75
Modified Dice Coefficient (top-1)	0.65	0.58
Modified Dice Coefficient (top-5)	2.85	2.77
χ^2 Test (top-1)	0.65	0.69
χ^2 Test (top-5)	2.76	3.02

Table 3.5: Related words of German Out-Of-Vocabulary words with filtering function words

The recommended measures for finding semantically related words of rare words were Log-Likelihood Ratio and χ^2 Test. Even though they were in the top, Mutual Information did better. Nevertheless some improvement can be done. Also the fact that the top-5 bring worser results than top-1 is not expected to be this way.

In a second approach the experiment is repeated in exactly the same way except that this time only semantic words are considered which are known by a lexicon built by Moses from the EPPS, NC, BTEC and TED corpora. The figures in 3.6 show that the average hits of Log-Likelihood Ratio, Mutual Information and χ^2 Test are lower than before. The average hits for Weighted Mutual Information and Modified Dice Coefficient however are increasing. In fact Modified Dice Coefficient gets 72% hits in the top-1 category and 58%

hits in the top-5 category. Here again, the figures for top-5 are lower than the figures for top-1.

Measure	Average of finding related words	Average of finding related words considering only semantic words
LogLikelihood Ratio (top-1)	0.25	0.26
LogLikelihood Ratio (top-5)	1.2	1.24
Mutual Information (top-1)	0.13	0.12
Mutual Information (top-5)	1.13	1.12
Weighted Mutual Information (top-1)	0.57	0.62
Weighted Mutual Information (top-5)	2.13	2.28
Modified Dice Coefficient (top-1)	0.7	0.72
Modified Dice Coefficient (top-5)	2.77	2.92
χ^2 Test (top-1)	0.43	0.44
χ^2 Test (top-5)	1.617	1.64

Table 3.6: Related words of German Out-Of-Vocabulary words with filtering function words and dictionary entries

The results again display the gape of possible improvement to find related words. Also it indicates a strong dependency of the different measurements on which words are considered as possible related words. But for more reliable figures, more lectures and lexicons have to be tested.

3.4.3 Finding translations

Having found related words, the next step is to try to find a translation based on the translations of the related words. Therefore the results of the last step were taken and verified manually. The translations then should lead to the wanted translation. In a first approach, the query for obtaining related words contains the German words whose translation we are looking for. For merging gathered scores of a possible translation, three methods will be inspected. In the first only the maximum will be considered. The second one is calculating the geometrical mean the last one is calculating the arithmetic mean. The best performances in the top-1 category do not exceed 8%. In the top-5 category the best result is 8% from Mutual Information taking the maximum (3.7).

It is obvious that the different ways of merging have not so different outputs. This is due to that in average only 2 or 3 words are used to find the translation. Also considering the top-5 instead of the top-1 results does not bring any improvement. Looking into the translation candidates, a considerably high part actually is German. It is quite difficult to distinguish generally between German and English if not using a set of known words. Unlike Japanese and English, German and English share almost the same character set and can't be distinguished syntactically.
Also almost all candidates which are not correct are semantically related at least. Most of the candidates are monograms even though bi- and trigrams have been considered.

In a second approach, the experiment is repeated but without the word whose translation is searched for and with a flag which indicates to look out only for English web pages(**??**). Again three different merging methods - taking the maximum, geometric mean and arithmetic mean - are tested and the top-1 and top-5 results are considered. Considering only the top-1 result, the best performing method is Modified Dice Coefficient with 16%

Measure	Average of finding translation (top-1)	Average of finding translations (top-5)
Log-Likelihood Ratio (maximum)	0.04	0.16
Log-Likelihood Ratio (geometric mean)	0.04	0.2
Log-Likelihood Ratio (arithmetic mean)	0.08	0.36
Mutual Information (maximum)	0.04	0.4
Mutual Information (geometric mean)	0.04	0.12
Mutual Information (arithmetic mean)	0.04	0.12
Modified Dice Coefficient (maximum)	0.08	0.28
Modified Dice Coefficient (geometric mean)	0.04	0.24
Modified Dice Coefficient (arithmetic mean)	0.08	0.36
χ^2 Test (maximum)	0.08	0.12
χ^2 Test (geometric mean)	0.04	0.12
χ^2 Test (arithmetic mean)	0.04	0.08

Table 3.7: Finding translation with German Out-Of-Vocabulary word an translation of related words, based on 1,2,3-gram

Measurement	Top-1	Top-5
Log-Likelihood Ratio (maximum)	0	0.08
Log-Likelihood Ratio (geometric mean)	0.04	0.12
Log-Likelihood Ratio (arithmetic mean)	0	0.12
Mutual Information (maximum)	0	0.08
Mutual Information (geometric mean)	0	0.08
Mutual Information (arithmetic mean)	0.04	0.08
Modified Dice Coefficient (maximum)	0.16	0.32
Modified Dice Coefficient (geometric mean)	0.12	0.36
Modified Dice Coefficient (arithmetic mean)	0.12	0.36
χ^2 Test (maximum)	0	0.08
χ^2 Text (geometric mean)	0.04	0.12
χ^2 Test (arithmetic mean)	0.04	0.12

Table 3.8: Finding translation without German Out-Of-Vocabulary word solely on merging related words of all translations, based on 1,2,3-gram

hits. Taking the top-5 results into consideration, Modified Dice Coefficient with 7% hits. Compared with the first approach(3.7), this one gives worser figures.

The results show that a lot of improvement has to be done to effectively find translations. As it happened in the first approach, many related words occur as a candidate. But these words are often quite special like names of enterprises. Another source of the poor results could be the used related words. Putting up a list of related semantic words by hand, is not that easy since every related word happens to be often in its context. So synonyms often replace the actual word instead of having it in the context.
Another issue is the limitation of Bing API. In an experiment all related words translation were taken to calculate the co-occurrence with a translation candidate. Running it several times ended always after the same amount of requests.

3.5 Outlook

The tests in the chapter before have unveiled pretty clearly that improving and rethinking has to be done in order to build up a system which delivers reliable results. The basic concept seems quite reasonable: Finding related words and concluding the translation based on the related words found before.

First, the mechanism of finding related words has to be redesigned. The basic idea remains: putting the out-of-vocabulary expression into a Web search engine in order to obtain a list of web pages which contain the out-of-vocabulary expression. Web search engines are by far the most effective way to find high quality web pages. Instead of taking the description snippets as the context, it might be more appropriate to take the actual web page as the context window. In our tests we encountered the phenomena that an out-of-vocabulary word actually has important meaning in different areas. So it is important to find all those web pages which are related in any way to the source of the out-of-vocabulary words. Estimating the relation between the content of a web page and the source documents of the out-of-vocabulary can be done by calculating different frequency features and similarity measurements. Similar work has be done by Maergner et al.[**?**]. It is essential to collect a large set of highly related words because their translations are required. Unfortunately we have to assume a quite small and topic unrelated lexicon. The bigger the set of related words is the bigger is the likelihood to find a related word with a translation in the lexicon. Also the related words must be strongly related to the out-of-vocabulary word in order to have a good chance to conclude correctly its translation.
Going back from related words to the wanted translation on the English side, it gets more complicated. One single word or expression has to be found. As in our approach it is to be found in the intersection of all queries of the translated related words. Here again, the actual web pages can be taken instead of the description snippets. Making sure that the English web pages are related to the topic of interest cannot be done by relying directly on the source documents. One way would be to compare the web pages to a set of topic related English words. So web pages which are about something completely different can be filtered out. It would be perfect to have a mechanism to make sure that web pages are topic related. On the other hand a good diversity helps to decrease the intersection of all documents, so finding the right translation will be easier. Another issue which arises here, is that the wanted translation has not to be within every web page.
The other approach of concluding the wanted translation was by relying on bilingual web pages. It is based on the fact that sometimes a word appears together with its translation on a web page. Different suggestions building upon this fact have been described at the beginning of this chapter. It was easy for them in a certain perspective because they regarded language pairs which have different alphabet sets. So the different language portions can be distinguished simply by the characters. For languages which use the same alphabet set more sophisticated methods have to be developed to determine the right language. Also the different fractions can be evaluated by position features or bracket patterns rather than by co-occurrence statistics.

Also it might be interesting to investigate finding out-of-vocabulary words. So far documents will be tested on out-of-vocabulary words by comparing them to a set of already known words. Detecting an out-of-vocabulary expression where the single words are known needs another approach. This task can be modeled by an Hidden-Markov-Model. The transition between two words shows how likely they appear together. To find the transition probability is in some way tricky because a lot of word transitions has to be known. Unfortunately out-of-vocabulary expressions can have an arbitrary length. Theoretically the model needs to shrink or to extend the amount of states arbitrarily. But due to feasibility the amount of states has to be limited. So a compromise has to be done which recognizes a needed minimum of the expression but allows fast computation.

Another important stage is the actual documents in which out-of-vocabulary words are searched for. Already in the conversion step, unhelpful content like mathematical equations, footnotes and diagrams can be filtered. But it is questionable whether such information are saved within the document format. Unfortunately formats like PDF or Power Point do not provide such a clear and open structure as HTML or XML do. Apart from recognizing those unwanted content parts it takes some effort to generally parse these files. It might be also interesting to consider not only material directly connected to the lecture but also secondary literature. Services like Google Books or the digital archive of the university library offer standard references which can be used to extend the search.

For our basic idea of finding out-of-vocabulary words, it is necessary to have documents which are about the lecture. But one crucial problem we encountered was that often professors and teachers lock their document with a password due to copyright issues. If no material nor abundant description is given, it might be quite difficult to predict possible out-of-vocabulary words. A campus wide solution and extend communication might help to overcome this issue.
But for some situations, getting some material machine readable is difficult. Some lectures still uses to teach the students with a blackboard rather than with digital presentation slides. In this case the possible context has to be constructed by all information given by the course catalog. This requires more work expecting worser results.

4. Outlook

As illustrated, the Web has an enormous capacity of translation examples. Nevertheless, it takes some effort to extract these. Crawling for translated articles appears quite interesting because of its simplicity and independence of language. Extending the idea to a broader spectrum in which the crawler crosses web sites and tries to find translations outside of the web site, helps to imagine which potential in Web still could be uncovered.
Also the basic idea of finding translations for specific words sounds quite reasonable. So far, more sophisticated techniques could help to find translations more effectively. Some techniques which might help are already out there and have been named in this work. Other techniques have to be developed.

After all, the Web offers a lot of advantages for Machine Translation. But one reason of its advantages causes a big disadvantage in the same time. The reason is the simplicity of creating content on the Web. This is only possible because the required technical formality is limited to a certain extend. So people can create content which is highly diversified in every dimension. For parallel corpora like these from the European Council or from the Canadian parliament, much effort had to be put into. For such corpora, professional translators have to be employed. Because of this, there are not many big parallel corpora. Regarding the Web, translations are applied only to small articles and in a way which pleases the translator. But since people can create content quite easily, there might be more of such small articles out on the Web. In a consequence, summing all these small articles might exceed the size of the parallel corpora.
Compared to the parallel corpora, the individual small articles probably are of different quality and often of lower quality. So one disadvantage is that the different qualities have to be taken into consideration in order to optimize the benefit.
Another challenge for machine translation or data mining in general lies in how to uncover translational relations between the diversified content of the Web. The traditional parallel corpus is divided into two or more languages and each text of such a corpus is divided into sections which can be easily assigned to a section of a text in another language. This pattern can be found on the Web only to a certain extend. Getting more out of the Web means also to develop methods which helps to find such translational relations. One way displayed in this work is, to deduct a translation from the Context.

At the end, the challenge for mining the Web is more about to do it extensively rather than intensively. So the lesson is not to concentrate on how much do I get out of a set of data but how do I enlarge this set of data.

5. Appendix

5.1 Anchor elements crawling

Listing 5.1: Crawl and collect all redirections and possible bilingual web pages

```
1  // set of all redirection which occurr
2  R: Set<(URL,URL)> = ∅
3
4  // set of all references which indicate language L1
5  L1: Set<(URL,URL)> = ∅
6
7  // set of all references which indicate language L2
8  L2: Set<(URL,URL)> = ∅
9
10 // set of invalid suffixes
11 I: Set<Text> = ∅
12
13 // set of texts which indicate language L1
14 T1: Set<Text> = {...}
15
16 // set of texts which indicate language L2
17 T2: Set<Text> = {...}
18
19 // function to determine wether a text belongs to language L1
       or not
20 p1: function = ...
21
22 // function to detemrine wether a text belongs to language L2
       or not
23 p2: function = ...
24
25 // set of starting seeds
26 S: Set<URL> = {...}
27
28 // starting point for crawling
29 for each s ∈ S do Crawl(s)
```

```
30
31  procedure Crawl(u:URL)
32  do
33    // try to download web page
34    r:Response = Download(u)
35
36    // test wether url is http redirection
37    if IsRedirection(r) then
38    do
39      R = R ∪ {(u, GetRedirectionTarget(r))}
40    done
41
42    // test wether url refers to valid web page
43    else if IsSuccessful(r) then
44    do
45
46      // test wether content of web page is valid
47      if IsValidContentType(r) then
48      do
49        L̃1:Set<URL> = GetLanguageLinks(r, T1, p1)
50        L̃2:Set<URL> = GetLanguageLinks(r, T2, p2)
51
52        for each l ∈ L̃1 do L1 = L1 ∪ {(u,l)}
53        for each l ∈ L̃2 do L2 = L2 ∪ {(u,l)}
54
55        L̃:Set<URL> = GetLinks(r)
56        for each l ∈ L̃ do Crawl(l)
57      done
58      else
59      do
60        suffix:Text = GetSuffix(u)
61        I = I ∪ {(suffix)}
62      done
63    done
64  done
65
66  // test wether an HTTP response is redirecting
67  function IsRedirection(r:Response):bool
68  do
69    return r.status >= 300 and r.status <= 399
70  done
71
72  // test wether an HTTP response was successful
73  function IsSuccessful(r:Response):bool
74  do
75    return r.status == 200
76  done
77
78  // test the content type of a web page
79  function IsValidContentType(r:Response):bool
80  do
81    return r.headers['ContentType'] is '' or
```

```
            r.headers['ContentType'] starts with 'text/html' or
            r.headers['ContentType'] starts with 'text/raw'
done

// takes all language links
function GetLanguageLinks(r:Response, T:Set<Text>, p:
    LanguageIdentification):Set<URL>
do
  if p(r.body.text) then
  do
    A:Set<HTMLAnchor> = GetHTMLAnchors(r)
    return {a.url | a ∈ A ∧ (a.rel ∈ T ∨ a.rev ∈ T ∨ a.hreflang ∈ T ∨
        a.lang ∈ T ∨ a.title ∈ T ∨ a.xml::lang ∈ T ∨ a.alternate ∈ T ∨
        a.innerText ∈ T)}
  done
  else return ∅
done

// collect all links from a web page
function GetLinks(r:Response):Set<URL>
do
  A:Set<HTMLAnchor> = GetHTMLAnchors(r)
  return {a.url | a ∈ A}
done

// takes all anchor elements from html document
function GetHTMLAnchors(r:Response):Set<HTMLAnchor>
do
  return r.body.anchors
done
```

Listing 5.2: Resolve and find symmetric pairs

```
// set of http redirections
R:Set<(URL,URL)> = {...}

// set of link references of language L1
L1:Set<(URL,URL)> = {...}

// set of link references of language L2
L2:Set<(URL,URL)> = {...}

// resolved links for pairs where the source page is in
    language L1
L̄1:Set<(URL,URL)> = Resolve(L1)

// resolved links for pairs where the source page is in
    language L2
L̄2:Set<(URL,URL)> = Resolve(L2)

// bilingual web pages (the result)
Result:Set<(URL,URL)> = {(a,b) | (a,b) ∈ L̄1 ∧ (b,a) ∈ L̄2}

```

```
19  // resolve any redirections, so that the outcoming pairs have
        reference to actual web pages
20  function Resolve(L:Set<(URL,URL)>, R:Set<(URL,URL)>):Set<(URL,
        URL)>
21  do
22    return {(s,t) | (s,t) ∈ L ∧ ∄x:(t,x) ∈ R} ⋃ {(s,t) | (s,y) ∈ L ∧
          (y,t) ∈ R}
23  done
```

Bibliography

[Ber] M. K. Bergman, "White Paper: The Deep Web: Surfacing hidden Value."

[CTC+] P.-J. Cahng, J.-W. Teng, R.-C. Chen, J.-H. Wang, and W.-H. Lu, "Tranlsation Unknown Queries with Web Corpora for Cross-Language Information Retrieval."

[Dic] [Online]. Available: www.dict.cc/

[goo] [Online]. Available: http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

[HMNW] T. Herrmann, M. Mediani, J. Niehues, and A. Waibel, "The Karlsruhe Institute of Technology Translation Systems for the WMT 2011."

[HZV] F. Huang, Y. Zhang, and S. Vogel, "Mining Key Phrase Translations from Web Corpora."

[Lec] [Online]. Available: http://isl.ira.uka.de/english/127.php

[Lyn] [Online]. Available: http://lynx.isc.org/

[ML] X. Ma and M. Y. Liberman, "BITS:A Method for Bilingual Text Search over the Web."

[MM] D. S. Munteanu and D. Marcu, "Extracting Parallel Sub-Sentential Fragments from Non-Parallel Corpora."

[MSYU] A. Maeda, F. Sadat, M. Yoshikawa, and S. Uemura, "Query Disambiguation for Web Cross-Language Information Retrieval using a Search Engine."

[nlt] [Online]. Available: http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html#sec-further-examples-of-supervised-classification

[NSID99] J.-Y. Nie, M. Simard, P. Isabelle, and R. Durand, "Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts from the Web," 1999.

[Rap] R. Rapp, "Automatic Identification of Word Translations from Unrelated English and German Corpora."

[RS03] P. Resnik and N. A. Smith, "The Web as a Parallel Corpus," 2003.

[SNZG06] L. Shi, C. Niu, M. Zhou, and J. Gao, "A DOM Tree Alignment Model for Mining Parallel Data from the Web," *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pp. 489–496, 2006.

[Unia] [Online]. Available: http://wortschatz.uni-leipzig.de/Papers/top10000de.txt

[Unib] [Online]. Available: http://wortschatz.uni-leipzig.de/Papers/top10000en.txt

[Web] [Online]. Available: http://www.archive.org/

[wika] [Online]. Available: http://de.wikipedia.org/wiki/Liste_deutscher_Universit%C3%A4ten

[wikb] [Online]. Available: http://de.wikipedia.org/wiki/Liste_der_Universit%C3%A4ten_und_Fachhochschulen_in_%C3%96sterreich

[wikc] [Online]. Available: http://de.wikipedia.org/wiki/Schweizer_Universit%C3%A4ten_und_Fachhochschulen

[wor] [Online]. Available: http://www.worldwidewebsize.com/

[ZV] Y. Zhang and P. Vines, "Using the Web for Automated Translation Etraction in Cross-Language Information Retrieval."