D3.3.2: Corpus Modelling Module (ver. 2)

Grant Agreement Number	ICT–248307
Project Acronym	PresemT
Project Title	**P**attern **RE**cognition-based **S**tatistically **E**nhanced **MT**
Deliverable Title	**D3.3.2: Corpus Modelling Module (ver. 2)**
Responsible Partner	**NTNU** (Björn Gambäck)
Dissemination Level	Public
Due Delivery Date	31 December 2011 (+60 days)
Actual Delivery Date	17 January 2012

Project Coordinator Name	**Dr. George Tambouratzis**
Project Coordinator Site	**Institute for Language and Speech Processing / RC 'Athena'**
Tel	+30 210 6875411
Fax	+30 210 6854270
E-mail	giorg_t@ilsp.gr
Project Website Address	www.presemt.eu

Executive summary

The PreseMT Corpus Modelling Module is an off-line module. It takes as input an annotated text corpus in the target language. From this, it infers a corpus model, which is an abstraction of certain aspects of the text corpus. Since the text corpus is a sample from the target language, a corpus model is also a language model, which is a more conventional term. The task of the Corpus Modelling Module is to support the Translation Equivalent Selection Module in the translation of individual phrases, which primarily involves word translation and word ordering.

Evaluation of such language models is unfeasible without a full implementation of the PreseMT translation system and accompanying translation evaluation procedures. Instead, initial work has focussed on one particular aspect of the translation process for which it is easier to evaluate the contribution of different language models. This is the task of Word Translation Disambiguation (WTD), which amounts to selecting the best translation(s) given a source word instance in its context (e.g., a sentence) and a set of target word candidates (e.g., from a bilingual dictionary). One of the advantages of this is that we can reuse the framework from the word translation tasks for several language pairs in SemEval 2010. The work experimental evaluations described in the present deliverable concern the English-to-German part of the SemEval Cross-Lingual Word Sense Disambiguation.

One of the features that distinguishes the PreseMT approach to MT from mainstream statistical MT is that it tries to avoid relying on large parallel text corpora for training purposes, a resource that is both scarce and expensive. Instead, it aims at learning patterns in the source and target language, and the mapping between them — from very large annotated monolingual corpora only.

The work on translation disambiguation without parallel text has proceeded along three lines. The initial approach relies on Vector Space Models and is based on the assumption that the meaning of a word can be inferred from its usage, i.e., its distribution in text. What makes this approach particularly attractive in the context of PreseMT is that it does not require any external knowledge resources besides a large text corpus and that it is fully unsupervised (i.e., no need for human annotation).

The second approach aims at a straight-forward practical solution based on statistical n-gram models, which are currently the *de facto* language models in NLP, including (statistical) MT. Statistical n-gram models allow for a "generation and ranking" approach to translation which consists of generating alternative word translations and word orders, and subsequently ranking these alternatives according to their perplexity in order to find the best translation. Even though n-gram models is an established technology, constructing such models on the basis of text corpora containing billions of words poses interesting challenges in the area of parallelization and high-performance computing.

Thirdly and most recently, we have investigated the use of the use of Kohonen's Self-Organising Map (SOM) model in order to model the TL language. Self-Organising Maps is an unsupervised classification strategy which aims to convert a high-dimensional input space of training samples into a low-dimensional representation. SOMs are employed to determine the semantic relevance of a translation candidate with respect to its context, and thus allow a quantitative comparison among all the available alternatives that are suggested as candidate translations by a bilingual

dictionary. One additional advantage of the SOM approach is that the models are small in terms of memory and can thus be processed very quickly and efficiently.

The work package has delivered a number of concrete results. Firstly, new methods for efficiently constructing statistical n-gram models from very large corpora (billions of words) have been developed using parallel processing techniques. The use of n-gram models for translation disambiguation has been experimentally evaluated. Secondly, innovative approaches to translation disambiguation without relying on parallel text corpora — based on Vector Space Models and Self-Organising Maps — have been proposed, developed, evaluated and reported. So far, the VSM approach has also been described in a scientific publication (more publications on all approaches are expected). Thirdly, both the VSM-based and SOM-based disambiguation approaches have been implemented in the PreseMT MT system. Their contribution to the overall translation quality is currently assessed.

Version history

This document relates to the second version of the Corpus Modelling Module. The first version of the module was reported on in Deliverable D3.3.1: "Corpus Modelling Module (ver. 1)" created at Month 12 of the project. Clearly, there is a reasonable amount of overlap between the two versions of the module, and even more so between the two versions of the textual deliverable. The following section summarises the main changes undertaken when creating the present text, for each chapter of the deliverable:

1 Introduction: Rewritten according to the many changes in the rest of the document.

2 Related work: No major changes.

3 The SemEval word translation tasks: Merges parts of Chapter 3 and Chapter 4 of the earlier deliverable, adding new text on the evaluation measures used.

4 Vector Space Modelling: Contains many minor changes and updates, and reports new experimental results.

5 Statistical N-gram Modelling: Changes mainly in introduction. Addition of a final section with experimental results on WTD.

6 Modelling with Self-Organising Maps: Completely new chapter.

7 VSM-based disambiguation in the PreseMT MT system: Completely new chapter.

8 Discussion and future work: Rewritten according to the changes in the rest of the document.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Corpus Modelling Module

The on-line PreseMT translation system comprises three major steps. First source language text is linguistically annotated, which includes the usual steps of tokenization, lemmatization and POS tagging, followed by lookup in a bilingual lexicon. Second, the *Structure Selection Module* determines the global structure of the translation by reordering phrases towards the order required in the target language. Third, the *Translation Equivalent Selection Module* takes care of the translation of individual phrases, which primarily involves resolving word translation choices, word ordering and morphological generation. The role of the *Corpus Modelling Module* is to support the Translation Equivalent Selection Module in accomplishing its tasks.

The Corpus Modelling Module is an off-line module. It takes as input an annotated text corpus in the target language. From this it infers a *corpus model*, which is an abstraction of certain aspects of the text corpus. For instance, a *word n-gram model* is an abstraction of the language limited to the probability of word sequences. Since the text corpus is a sample from the target language, a corpus model is also a *language model* (LM), which is a more conventional term. The goal of modelling can be to focus on a particular aspect of the language (e.g., word order) and/or to compress/encode relevant information in the text corpus to make access computationally feasible. In the remainder of this text we will use the terms "corpus model"/"corpus modelling module" and "language model"/"language modelling module" interchangeably.

Since the PreseMT Corpus Modelling Module delivers language models that can be used by the Translation Equivalent Selection Module, the design and implementation of the Corpus Modelling Module depends on the requirements of the Translation Equivalent Selection Module. However, at the time work on both modules started, this posed some practical problems. As work on the Translation Equivalent Selection was in progress, it was difficult to explicitly specify its requirements regarding the language models. At the same time, work on language modelling in isolation was difficult without an application context like the PreseMT translation system to serve as a framework for evaluating models. As a solution to these problems, work on language modelling has proceeded along two lines: a short term and a long term approach.

The first approach aimed at a short term practical solution based on statistical n-gram models. It acknowledges that statistical n-gram language models are currently the *de facto* language models in NLP, including (Statistical) Machine Translation. They allow for a "generation and ranking" approach to translation that first generates alternative word translations and word orders and then ranks these alternatives according to their perplexity in order to find the best translation. Statistical n-gram models partly addressed the direct needs of the Translation Equivalent Selection Module, enabling initial implementation work on translation selection to continue relatively independent from the work on corpus modelling. Even though n-gram modelling is an established technology, constructing such models on the basis of text corpora containing billions of words poses interesting engineering challenges in the area of parallelization and high performance computing. In addition, the n-gram models serve as the state-of-the-art baseline on which we aimed to improve in the second line of work.

The second approach targets development of new language models according to the PreseMT "Description of Work" (Annex I to the PreseMT Grant Agreement), in order to measure semantic similarity between word translations. This addresses the problem that, according to a bilingual dictionary or some other translation model, a source language word can often have several translations in the target language. For instance, the English word *knight* may be translated as the Dutch word *ridder* in the context of medieval history, but as *paard* in the context of a chess game. We can define this subtask in the translation process as follows:

> **Word Translation Disambiguation (WTD)**
> Given a source word instance in its context (e.g., a sentence) and a set of target word candidates (e.g., from a bilingual dictionary), the task of Word Translation Disambiguation is to select the best translation(s).

One of the features which distinguishes the PreseMT approach to MT from mainstream (statistical) MT is that it tries to avoid relying on large parallel text corpora for training purposes, a resource that is both scarce and expensive. Instead, it aims at learning patterns in the source and target language, and the mapping between them — from large annotated monolingual corpora only. In a similar vein, most empirical approaches to WTD crucially depend on word-aligned parallel text. In contrast, our goal is to develop data-driven methods for WTD that do not require any parallel text, but rely solely on the combination of bilingual dictionaries and large-scale monolingual corpora. Even though it may be unrealistic to expect that such methods would exceed those relying on parallel text in terms of performance, we ultimately aim to bridge the gap in performance between the two.

So far primarily three different strategies for WTD have been investigated within the PreseMT project. The first one is to exploit statistical n-gram models of the target language for this task. The second is to use Vector Space Models (VSM) to measure the similarity between the context of source word and the contexts of translations candidates in a target language corpus. The third approach uses Self-Organising Maps (SOM) to measure semantic similarity between consecutive translation candidates.

As argued earlier, evaluation of language models is unfeasible as long as we lack a full implementation of the PreseMT translation system and accompanying translation evaluation procedures. Yet, waiting for a fully-functional PreseMT MT system would have been equally

unfeasible. One of the advantages of working on Word Translation Disambiguation was that we could reuse the framework from two closely related tasks from SemEval 2010, namely, the Cross-Lingual Lexical Substitution task (Mihalcea et al., 2010) and the Cross-Lingual Word Sense Disambiguation task (Lefever and Hoste, 2010b). This provided an experimental framework with test data, an evaluation method, baseline scores, and scores of competitive systems. The SemEval Cross-Lingual Word Sense Disambiguation task offers data sets for several language pairs, namely translation from English to German, Dutch, French, Spanish, and Italian. The work on WTD described here reuses the English-to-German part of the CL-WSD task as targeted language pairs in the PresemT context. (This may be extended in WP9 to include English-Italian data, as Italian is one of the target languages in the project's final year.)

In the second year of the project, the focus has gradually shifted towards integrating the corpus modelling approaches in the PresemT MT system. This involved implementing the corpus modelling module and interfacing it with the other modules, primarily the translation Equivalent Selection module. This required generalizing and scaling up the approaches and finding ways to make them computationally feasible, for example, by parallelisation.

1.2 Deliverable outline

The rest of the deliverable is structured as follows: the first two chapters give background and establish the framework in which the experiments were run. Hence Chapter 2 starts out by discussing related work — in particular different approaches to word translation — and Chapter 3 then gives an overview of the two SemEval 2010 word translation tasks, the Cross-Lingual Lexical Substitution task and the Cross-Lingual Word Sense Disambiguation task, that provide us with an experimental framework.

One of the advantages of reusing the SemEval framework is that it includes an evaluation method. However, it is not without problems. The evaluation criteria form the topic of Section 3.3 which proposes some modifications to the SemEval criteria that are needed in the PresemT context.

The chapters thereafter go into detail on the three lines of research on language modelling within Task T3.4 ("Design and implementation of the Corpus modelling module") of PresemT WP3, "Corpus extraction & processing algorithms". Chapter 4 discusses the use of Vector Space Models in Word Translation Disambiguation. The chapter reports experimental results on applying this approach to the English-to-German part of the SemEval Cross-Lingual Word Sense Disambiguation task. Chapter 5 details how n-gram models have been created from corpora mined from the web and applied to Word Translation Disambiguation. Similarly, Chapter 6 describes the creation of Self-Organising Maps for the disambiguation task and reports some results with this approach.

The corpus modelling module in the PresemT system incorporates the VSM-based WTD as one of its appraoches. Chapter 7 details the actual module, both its off-line preprocessing and its on-line component.

Finally, Chapter 8 concludes the discussion of the present version of the Corpus Modelling Module and points to some future directions of research that could be pursued.

Chapter 2

Related work

Koehn and Knight (2001) compare different methods to train word-level translation models for German-to-English translation of nouns. These methods cover a logical range of conceivable approaches to data-driven word translation. It is therefore a good starting point to map work on word translation/disambiguation and to get a notion of the relative scores obtainable by different approaches.

1. **Using parallel corpus and lexicon**

   A bilingual lexicon is used to extract word-level noun translation pairs from a parallel corpus. Using context words as features, supervised machine learning techniques (e.g., decision lists) can then applied to predict the correct translation of a source word in its context. This method gave the best scores in Koehn and Knight's (2001) experiments.

2. **Using parallel and monolingual corpora and lexicon**

   This method uses Yarowsky's (1995) bootstrapping algorithm in combination with a German monolingual corpus to bootstrap training. However, bootstrapping did not lead to any performance improvement.

3. **Using only parallel corpus**

   This applies the standard SMT as a noisy channel approach, using GIZA for word alignment, but without word alignments being restricted by a lexicon. Performance dropped significantly, especially for less frequent words.

4. **Using monolingual corpora and lexicon**

   The first approach here is to simply always choose the translation candidate which occurs most frequently in the target language corpus. The second approach is to build a language model and use it to pick the most probable word sequence in the target language. The third approach relies on monolingual source and target language corpora in combination with the Expectation Maximization algorithm to learn word translation probabilities. Performance of the latter two is comparable to that of using only a parallel corpus.

5. **Using only monolingual corpora**

Table 2.1: Accuracy for various word translation methods as given by Koehn and Knight (2001)

Knowledge source:	Method:	Accuracy (%):
Parallel corpus + lexicon	most frequent	88.9
Parallel corpus + lexicon	decision list	89.5
Parallel corpus	Giza	76.9
Monolingual corpus + lexicon	most frequent	75.3
Monolingual corpus + lexicon	language model	77.3
Monolingual corpus + lexicon	EM	79.0
Monolingual corpus	identica	11.9
Monolingual corpus	spelling + context	38.6

This involves various attempts to bootstrap a translation dictionary from monolingual corpora. Words that are identical in both languages serve as a seed to the bootstrap process. Several heuristics are then used to extend the lexicon: similar context, similar spelling, similar co-occurrence relations, and similar frequency. Interesting as they may be, performance is really low.

A quantitative comparison of these methods is given in Table 2.1. As is to be expected, word translation — including its subtask of word translation disambiguation — is significantly harder without access to parallel text. With access to monolingual corpora only, a good lexicon is absolutely required.

Since one of the main goals of the PRESEMT project is to avoid using parallel corpora — and since there is a huge body of work on word translation and related matters — the discussion of related work in this chapter will be restricted to the fourth approach above, that is, to translation using monolingual corpora in combination with bilingual dictionaries.

2.1 Lexical acquisition using vector space models

Rapp (1995) proposes a method for extracting word translations from unrelated monolingual corpora. It is based on the idea that words that frequently co-occur in the source language also have translations that frequently co-occur in the target language: If, for example, in a text of one language two words A and B co-occur more often than expected from chance, then in a text of another language those words which are translations of A and B should also co-occur more frequently then expected. First, word co-occurrence matrices are constructed for source and target language. Next, rows/columns of one matrix are permutated to make its counts most similar to those in the second matrix. This results in both matrices having similar, i.e., translationally equivalent, words along their rows/columns. Although Rapp's (1995) goal is automatic acquisition of word translations, the concept of exploiting the distributional similarity between translations in the form of a vector space is similar to our approach (see Section 4.1).

Fung and McKeown (1997) and Fung and Yee (1998) formulate Rapp's method in terms of a vector space model and use it to extract translation equivalents from comparable text, introducing a seed lexicon to make it computationally feasible.

Rapp (1999) continues along the same line — using a seed lexicon — to describe a practical implementation with good results. Using a target language corpus, a word co-occurrence matrix is computed whose rows are all word-types occurring in the corpus and whose columns are all target words appearing in the bilingual lexicon. Given a source language word, whose translation is to be determined, a source language corpus is used to construct a co-occurrence vector for this word. All known words in this vector are translated to the target language. As the seed lexicon is small, only some translations are known. All unknown words are discarded and the vector positions are sorted in order to match the vectors of the target-language matrix. This vector is compared to all vectors in the target language corpus. The vector with the highest similarity is considered to be the translation of the source language word.

In many respects, this approach is almost identical to the PreseMT use of a vector space model for WTD (which is discussed in Section 4.1). The crucial difference is a difference in goal. Rapp's (1999) goal is to bootstrap a bilingual lexicon, whereas our goal is to disambiguate word translations. As a result, Rapp's input consists of a source word in isolation for which contexts are retrieved from a source language corpus, while our input consists of a source word in a particular context.

Chiao et al. (2004) explore a very similar method with domain-specific comparable corpora of limited size. In addition, they re-score translation candidates in the target language by applying the same translation algorithm in the reverse direction and re-ranking them according to the harmonic mean score.

Rapp and Zock (2010) claim a significant improvement over the previous algorithm (Rapp, 1999): when creating the co-occurrence vector for a source word, only the 30 most strongly associated words are kept and all others are eliminated.

2.2 Estimating word translation probabilities using Expectation Maximization

Koehn and Knight (2000) propose to use an n-gram model of the target language to select translations candidates that occur in the most likely candidate sequences (as in the PreseMT short-term approach outlined in Chapter 5), reporting an improvement in accuracy of about 2% on German to English translation of nouns. The language model is then used in combination with a bilingual lexicon and a monolingual corpus to estimate word translation probabilities. This is accomplished with a form of the Expectation Maximization algorithm.

Monz and Dorr (2005) also employ an iterative procedure based on Expectation Maximization to estimate word translation probabilities. However, rather than relying on an n-gram language model, they measure association strength between pairs of target words, which they claim is less sensitive to word order and adjacency, and therefore data sparseness, than higher n-gram models. Their evaluation is only indirect as application of the method in a cross-lingual IR setting.

2.3 Query translation in Cross-Lingual Information Retrieval

Kishida (2005) reviews state-of-the art techniques for Cross-Lingual Information Retrieval (CLIR), in which users search documents written in a foreign language with a query written in their own language. The most widely used strategy is translation of the query to the target language using machine machine-readable dictionaries. This gives rise to a term ambiguity problem which is very similar to word translation disambiguation in MT, except that search queries are often sets of keywords rather than proper linguistic utterances. The problem is that if all translations listed in the dictionary are used as search terms, irrelevant terms are likely to harm precision. Among the disambiguation techniques developed in CLIR, most relevant to our discussion are those based on co-occurrence statistics. These are based on the idea that correct translations of terms are more likely to co-occur in documents than incorrect translations. Numerous researchers have taken this idea and implemented some version of it.

Ballesteros and Croft (1998) is one of the first studies about translation disambiguation using co-occurrence statistics: "The correct translations of query terms should co-occur in target language documents and incorrect translations should tend not to co-occur." Their algorithm for resolving translation ambiguities is basically as follows. Given two tagged source terms t_1 and t_2, they retrieve all available translations from a dictionary. Next they generate all possible pairs of translations (a, b) such that a is translation of t_1 and b is a definition of t_2. The importance of co-occurrence of the elements in a set is measured by the *em* metric (Xu and Croft, 1998), a variation on mutual information (Church and Hanks, 1989) which does not favor infrequent co-occurrences. It essentially measures the percentage of the co-occurrences of a and b within a window (250 words) in the target corpus, corrected for the number of expected co-occurrences. Each set is ranked by its *em* score and the highest ranking set is taken as the appropriate translation. Ballesteros and Croft (1998) compare co-occurrence and parallel corpus methods for term disambiguation w.r.t. translation accuracy and find that the former performs significantly better than the latter (47 out of 60 correct vs. 39 out of 60 correct). Essentially the same approach is also found in Lin et al. (1999).

Jang et al. (1999) continues in the line of Ballesteros and Croft (1998), focusing on pruning translations. Given the source terms in the query, they first calculate the mutual information (MI) between consecutive translation candidates by searching for co-occurrences in window of 6 words (but without crossing sentence boundaries) in the target corpus. Heuristics are then used to prune translations. The translation pair with the highest MI is selected first and serves as the point of departure from which the connected translations with the highest MI values are chosen. An experiment shows improvement on IR results, but Jang et al. (1999) do not report numbers on translation accuracy as such.

Maeda et al. (2000) use the web as corpus for scoring mutual information using a document as the window size. They generalize mutual information for word pairs to mutual information between an arbitrary number of words. Other measures tested include a modified Dice coefficient, Log likelihood ratio and Chi-square. Their procedures for selection of translations are described in detail, and basically rely on exhaustive search for the best translations in combination with frequency based pruning. Experiments showed no significant differences w.r.t. IR between these measures. Slight variations on this approach can be found in Sadat et al. (2002).

Gao et al. (2000, 2002) is also similar to Ballesteros and Croft (1998), using a similarity measure which combines mutual information with the distance between terms (measured in words, within the window of a single sentence). Similarity is not only calculated between consecutive translation pairs, but between all the target candidates. A greedy algorithm is used to find the best translations.

Qu et al. (2003) present work on WTD in the framework of CLEF-2002 (the 'Cross-Language Evaluation Forum'). They compare three methods:

Web method: Query the web for trigrams of translation candidates and use the number of hits as a coherence score. Select the best-scoring translations.

Corpus method 1: Constructs all possible translations and use each of them to retrieve documents from the target corpus. Compute the sum of the similarity scores of the top N retrieved documents as the coherence score for the sequence.

Corpus method 2: Construct all possible trigrams of translation candidates. Compute mutual information for term pairs in the trigram, and add these to get the coherence score for the trigram. Select as translation for the first word in the trigram the alternative which gives the best coherence score.

Experimental results show better IR performance, but Qu et al. (2003) do not report numbers on translation accuracy as such.

Kishida (2007) provides an empirical comparison of different similarity measures and different algorithms for selecting the best translation (in addition to pseudo-relevance feedback techniques) over several data sets/language pairs. Although there are no significant differences in terms of IR, cosine similarity in combination with a best sequence algorithm tends to give best performance.

Chapter 3

The SemEval word translation tasks

The work on Word Translation Disambiguation (WTD) in PresesMT partly reuses the framework from two closely related tasks from SemEval 2010, namely, the Cross-Lingual Lexical Substitution task and the Cross-Lingual Word Sense Disambiguation task. This provides a platform for evaluation in the form of trial and test data, an evaluation method, baseline scores, and scores of competitive systems (relying on parallel data). This chapter first reviews relevant parts of both of these shared tasks. Modifications required to make it suitable for the WTD task are discussed in the final section of the chapter.

3.1 SemEval-2010 Task 2: Cross-Lingual Lexical Substitution

The Cross-Lingual Lexical Substition (CL-LS) task[1] (Mihalcea et al., 2010; Sinha et al., 2009) is based on the earlier English Lexical Substitution task from SemEval-2007 in which systems had to find an alternative (synonym) substitute word or phrase for a target word in its context (McCarthy and Navigli, 2007). In the 2010 Cross-Lingual Lexical Substitution task, however, only the source is English while the target word is Spanish. This makes it almost identical to Word Translation Disambiguation except that the set of translation candidates is not given in advance. The task may be envisioned as consisting of two steps:

1. candidate selection, which involves finding all possible translations;
2. candidate ranking, which involves finding the most likely translation among the candidates.

In contrast to the Cross-Lingual Word Sense Disambiguation task described in the next section, there is no intermediate layer of senses.

[1] `http://semeval2.fbk.eu/semeval2.php?location=tasks#T24`

3.1.1 Data

The data consist of instances of nouns, verbs, adjectives and adverbs in a single sentence context. The development set consists of 30 words (10 instances per word, 300 instances in total) and the test set consists of 100 words (10 instances per word, 1000 instances in total). Four annotators, all native Spanish speakers, provided as many adequate translations for each word in its context as they could think of. The annotation includes for each candidate the number of annotators that choose it (i.e., minimally 1 and maximally 4).

3.1.2 Evaluation

Participating systems produce one or more translations, where the order is significant (most likely translation first). The evaluation basically measures the fit between the system's translations and the annotators' translations in terms of precision and recall, using two scoring variants. The 'best' (Best) score measures the ability of the system to come up wih the best translations, and penalizes for additional guesses. System translations are given credit depending on the number of annotators that picked each translation, while being punished for any non-matching translations. The 'out-of-ten' (OOT) score allows up to ten system responses without punishment for non-matching translations. This takes into account that there may be good translations that the annotators had not thought of.

For both best and out-of-ten scores, there is also a 'mode' score, which is calculated against the mode from the annotators responses. The 'mode' is the target term with the highest frequency, and is not defined if two or more terms share this distinction. Note that the best 'mode' score is not penalized by the number of submitted terms. The 'mode' criterion measures the ability of the system to include the term with highest count of inter-annotator agreement.

The Best criteria are defined by McCarthy and Navigli (2007) using the following formulae.

$$\text{Precision} = \frac{1}{|A|} \sum_{a_i;i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|a_i| \cdot |H_i|} \tag{3.1}$$

$$\text{Recall} = \frac{1}{|T|} \sum_{a_i;i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|a_i| \cdot |H_i|} \tag{3.2}$$

And the OOT criteria as the following.

$$\text{Precision} = \frac{1}{|A|} \sum_{a_i;i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|} \tag{3.3}$$

$$\text{Recall} = \frac{1}{|T|} \sum_{a_i;i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|} \tag{3.4}$$

Using the following terms (here described in an informal manner):

- a_i are the answers submitted by the system,
- $|A|$ is the number of answers,
- $|T|$ is the number of test items,
- $|H_i|$ is the amount of inter-annotator agreement, i.e., the sum of annotator votes for all gold terms for this test item, and
- $freq_{res}$ is the number of annotator votes for this particular system answer.

3.1.3 Results

Baselines are calculated by taking the (ordered) translations from an online dictionary. Only 4 out of the 14 systems submitted have a best score above the baseline. The best system (UBA-T) is essentially Google Translate complemented by some additional dictionaries (Basile and Semeraro, 2010). Results are somewhat better for the out-of-ten score, but this appears to be mainly due to the trick of adding duplicates. Virtually all systems (except for the SWAT and TYO systems) rely on parallel text. This suggests that the task is harder without parallel corpora.

3.2 SemEval-2010 task 3: Cross-Lingual Word Sense Disambiguation

The Cross-Lingual Word Sense Disambiguation (CL-WSD) task[2] (Lefever and Hoste, 2010b, 2009) is very close to the Cross-Lingual Lexical Substitution task. The main difference is that there is an intermediate level of sense clusters during the annotation stage. Annotators are therefore not free to pick just any translation for a given source word, but first have to select the appropriate sense cluster, and from that cluster must select up to three adequate translations. See the original papers for a motivation of this strategy.

3.2.1 Data

The source language is English and there are five target languages: Dutch, French, Spanish, Italian and German. In contrast to the CL-LS task, only lemmatized nouns are considered. The annotation process has two steps. First, a sense inventory is created. This is based on the word-alignment of the EuroParl corpus (Koehn, 2005). Alignments involving the source word are manually checked. The corresponding target words are clustered into sense clusters. Target words are also manually lemmatized.

Second, trial and test data is extracted from two independent corpora (JRC-ACQUIS and BNC). The development set consists of 5 nouns (20 instances per noun, 100 instances in total per language) and the test set consists of 20 nouns (50 instances per nouns, 1000 instances in total per language). For each source word, annotators were asked (1) to pick the contextually

[2] http://semeval2.fbk.eu/semeval2.php?location=tasks#T8

appropriate sense cluster and (2) to choose their three preferred translations from this cluster. Translations are thus restricted to those appearing in EuroParl. The sentence-aligned parallel text from which the sense clusters were derived was made available. The sense clusters are available for the trial data, but not for the final test data.

Below is a sample from the trial data in XML format, where each context element provides an English sentence which contains a surface form of the lemma 'bank'.

```
<?xml version="1.0" ?>
<!DOCTYPE corpus SYSTEM "clls.dtd">
<corpus lang="english">
 <lexelt item="bank.n">
   <instance id="1">
     <context>AGREEMENT in the form of an exchange of letters between
     the European Economic Community and the <head>Bank</head> for
     International Settlements concerning the mobilization of claims
     held by the Member States under the medium-term financial
     assistance arrangements</context>
   </instance>
   <instance id="2">
     <context>The BIS could conclude stand-by credit agreements with
     the creditor countries' central <head>banks</head> if they should
     so request.</context>
   </instance>
   <instance id="3">
     <context>CONSIDERING the importance of the existing links between
     the Community and the Palestinian people of the West
     <head>Bank</head> and the Gaza Strip, and the common values that
     they share</context>
   </instance>
 ...
 </lexelt>
</corpus>
```

The sample below of the gold standard for German lists the preferred translations corresponding to the above instances.

```
bank.n.de 1  ::  bank 4;bankengesellschaft 1;finanzinstitut 1;
                 kreditinstitut 1;zentralbank 1;
bank.n.de 2  ::  bank 4;finanzinstitut 1;kreditinstitut 1;
                 nationalbank 1;notenbank 1;zentralbank 3;
bank.n.de 3  ::  west-bank 1;westbank 2;westjordanien 2;westjordanland 2;
                 westjordanufer 3;westufer 2;
        ...
```

This means, for example, that for the first instance of the English word 'bank', four translators thought German *bank* to be a correct translation, and at least one of each translators also considered *bankengesellschaft, finanzinstitut, kreditinstitut* or *zentralbank* to be correct.

3.2.2 Evaluation

Evaluation is almost identical to that in the Cross-Lingual Lexical Substitution task, except that the out-of-ten score (OOT) is replaced by an out-of-five score (OOF).

3.2.3 Results

Baselines were constructed by selecting the most frequent translation(s) of the source word according to the word-aligned EuroParl corpus. There were 16 submissions from five teams. About half of the systems achieved a best score below the baseline. This was even worse for the out-of-five score, where none of the systems outperformed the baseline for Spanish and Dutch, whereas only one system was above the baseline for French, Italian and German. All systems relied on parallel data.

3.3 Evaluation criteria

One of the advantages of reusing the word translation task framework from SemEval 2010 in the PresemT approach to Word Translation Disambiguation is that the SemEval set-up includes an evaluation method. However, the original evaluation measures appear to have some deficiencies. We therefore adopt some alternatives.

3.3.1 Drawbacks of the SemEval scoring criteria

One drawback with these scoring criteria is that the maximum score obtainable for the target terms may often be very low in absolute terms. It is our opinion that evaluation criteria should give near perfect systems a score near the top of the scale and that the distance between two scores should have a reasonable interpretation as difference in system quality. The original SemEval CL-WSD criteria will potentially give low scores to very good systems. For example, if the annotators have selected ten target terms among them and a system has submitted all those ten and only those, the score will be the normalised sum of the words divided by the number of submitted terms, i.e., one divided by ten (scores are reported as percentages, i.e., 10.0). While if a system only delivers the top word and this has half of the annotator votes, it will receive a score of 50.0. Consider item 20 in the German gold set for 'Bank' (the numbers behind the gold terms are the inter-annotator agreement counts):

```
bank.n.de 20  ::  bank 4;bankgesellschaft 1;finanzinstitut 2;geschäftsbank 1;
                  handelsbank 1;kreditinstitut 1;
```

Here submitting the top word according to the annotators will give a score of 40.0 for this item, while submitting all correct items will give a score of 16.0.

One may debate if favouring systems submitting fewer terms in this manner is a reasonably scoring system for the task, but we consider it problematic that the distribution of the score weightings are dependent on the number of gold terms and how annotator agreement is distributed among them. This has the effect of making the scaling differ between the test items, and it is unclear whether this affects the score in an inappropriate manner.

A similar issue is the case for the OOF score: depending on the distribution of the annotator frequency counts, a large chunk of the full score may be unobtainable by any system since most target terms have substantially more than 5 gold candidates and all of them are part of the normalization weight. Consider the German gold file for 'Plant' where 14 out of 20 test items have more than five gold terms, and the minimum total score in the terms remaining after the top five are removed is about 17. The maximum score attainable is by consequence correspondingly lower. This is illustrated in the 'perfect system' experiments discussed in the next section. There is also substantial variation in the unavailable test item scores with the standard deviation being 10 over the items with more than five gold terms.

The lack of scaling between the theoretical maximum and minimum score is a clear drawback of these criteria. But what may be a more serious problem is the manner in which the scaling varies with the distribution of frequency counts among the gold terms of a test item. In other words, the scores are not normalized across words. One way to visualize this is to consider that the score is penalized by dividing by the number of submitted terms, which gives the system that submits a number of terms close to the amount that cover most of the weight of the annotator agreement for this particular item an advantage in the scoring, an effect which is debatable at best and which varies over the test items.

3.3.2 'Perfect system' scoring

In order to illustrate the maximum score attainable with the SemEval evaluation method, we simulated a perfect system for both the Best and OOF criteria. As can be seen in Table 4.5, the scores varied from around 0.20 to 0.50 for Best and 0.80 to 0.95 for OOF. This makes it difficult to compare scores and analyze improvements over time or over target terms. It may also make the mean of scores rather meaningless as a test statistic.

The perfect OOF system submits the five target terms with the highest annotator agreement weight, while the perfect Best system submits the single top annotator agreement term, minimizing the penalty for submitting multiple candidates. It might be possible to construct a slightly better perfect Best system by carefully studying the inter-annotator agreement distributions, but we believe that the difference will be slight, if it exists at all.

Looking at the 'perfect system' scores alongside the published baseline we can see the differences in score range which has to be taken into consideration when analyzing results by these evaluation criteria. One should also note that the more robust OOF or simple Best 'mode' score might be more easily understood in terms of system improvement. But the Best 'mode'

Table 3.1: The Best and OOF scores for the target terms from the published baseline and our simulated perfect system

	Bank	Movement	Occupation	Passage	Plant
Best baseline	2.49	3.91	13.45	4.58	11.70
Perfect Best	42.71	28.78	30.40	37.29	29.58
OOF baseline	23.23	20.34	32.78	27.35	21.06
Perfect OOF	95.60	82.62	93.58	89.57	81.97

score hinges on the system selecting the single 'mode' term, and the OOF score encourages the system to aggressively submit candidates which would be undesirable behaviour from the PRESEMT Corpus Modelling Module.

In the context of development of the Corpus Modelling Module, Recall is not very relevant for any of the scoring criteria since the WTD aims to produce a set of target terms for any head. As a result the Recall will always be equal to the Precision, since coverage is 100% by design.

3.3.3 Alternative evaluation measures

As discussed above, the Best measure has some deficiencies, most importantly that it is not normalized between 0 and 1. This results in a very uneven spread of scores, both among different target words and among the individual test sentences for each word, making it difficult — or not even meaningful — to judge differences in system performance by looking at average scores. Hence rather than using the original Best score, we adopt the normalized variant proposed by Jabbari et al. (2010), here referred to as Best_{JHG}. The formal description of these measures below also closely follows the formalisation by Jabbari et al. (2010).

For each sentence t_i, ($1 \leq i \leq N$, N the number of test items), let H_i denote the set of human translations. For each t_i there is a function $freq_i$ returning the count of how many annotators chose it for each term in H_i (0 for all others) and a value $maxfreq_i$ for the maximum count for any term in H_i. The pairing of H_i and $freq_i$ constitutes a multiset representation of the human answer set. Let $|S|^i$ denote the multiset cardinality of S according to $freq_i$, i.e., $\sum_{a \in S} freq_i(a)$, the sum of all counts in S. For the first example in discussed in Section 3.2.1: $H_1 = \{$bank, bankengesellschaft, kreditinstitut, zentralbank, finanzinstitut$\}$; $freq_1$(bankengesellschaft) $= 4$, $freq_1$(bank) $= 1$, etc; $maxfreq_1 = 4$; and $|H_1|^1 = 8$.

The Best_{JHG} measure is defined as follows

$$\text{Best}_{JHG}(i) = \frac{\sum_{a \in A_i} freq_i(a)}{maxfreq_i \times |A_i|} \tag{3.5}$$

where A_i is the set of translations for test item i produced by the system. The optimal score of 1.0 is achieved by returning a single translation whose count is $maxfreq_i$, with proportionally lesser credit given to answers in H_i with smaller counts. In principle a system can output several candidates in order to "hedge its bets", but there is a penalty for non-optimal translations, so the best strategy appears to be to output just one. The systems in our experiment always

produced a single translation for the Best_{JHG} score, so $|A_i| = 1$ always. In the first example of Section 3.2.1, the system output $A_1 = \{\text{bank}\}$ would give $\text{Best}_{JHG}(1) = 1.0$ whereas $A_1 = \{\text{bankengesellschaft}\}$ would give $\text{Best}_{JHG}(1) = 0.25$ and $A_1 = \{\text{ufer}\}$ would give $\text{Best}_{JHG}(1) = 0.0$.

The Out-Of-Five (OOF) criterion, which measures how well the top five candidates from the system match the top five translations in the gold standard, can be formalized in the same notation:

$$OOF(i) = \frac{\sum_{a \in A_i} freq_i(a)}{|H_i|^i} \tag{3.6}$$

In this case systems are allowed to submit up to five candidates of equal rank. It is a recall-oriented measure with no additional penalty for precision errors, so there is no benefit in outputting less than five candidates. With respect to the previous example from Section 3.2.1, the maximum score is obtained by system output $A_1 = \{\text{bank, bankengesellschaft, kreditinstitut, zentralbank, finanzinstitut}\}$, which gives $OOF(1) = (4+1+1+1+1)/8 = 1$, whereas $A_1 = \{\text{bank, bankengesellschaft, nationalbank, notenbank, sparkasse}\}$ would give $OOF(1) = (4+1)/8 = 0.625$. One remaining problem with the OOF measure is that the maximum score is not always one, i.e. not normalized, because sometimes the gold standard contains more than five translation alternatives.

For assessing overall system performance, the average of Best_{JHG} or OOF scores across all test items for a single source word is taken. The "mode" variant of both scores were not used in the evaluations for reasons explained by Jabbari et al. (2010).

3.4 Baseline and maximum scores

Two baselines were calculated on the Semeval CL-WSD data for German using Best_{JHG} and OOF measures. In addition, maximum scores were calculated to serve as an upper bound on system performance. These scores are listed in Tables 3.2 and reftab:oof-baseline-scores.

The first baseline (MostFrequentBaseline) does not rely on parallel corpora. It consists of simply selecting the translation candidate(s) whose lemma occurs most frequently in the deWaC corpus. It therefore completely ignores the context of the words. This results in generally low scores on the Best_{JHG} measure, even though the OOF scores for *bank* and *occupation* are high. The low scores may be due to differences between predominant translations in Europarl and in deWaC. Another factor which may reduce the efficiency of target side frequencies is that the word counts can be "polluted" because a certain German word is also the translation of another very frequent English word, a problem discussed by (Koehn and Knight, 2000, Section 3).

The second baseline (MostFrequentlyAligned) does rely on parallel corpora and was also used in the original CL-WSD shared task. It is constructed by taking the translation candidate most frequently aligned to the source word in the Europarl corpus with manually corrected source word alignments. As expected, the Best_{JHG} scores are consistently much higher than those of

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
MaxScore	100.00	100.00	100.00	100.00	100.00	100.00
MostFreqAlignBaseline	6.25	19.17	35.83	15.00	40.00	23.25
MostFreqBaseline	1.25	5.00	2.50	1.67	10.26	4.14

Table 3.2: Best$_{JHG}$ baseline and maximum scores on CL-WSD trial data

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
MaxScore	95.60	82.62	93.58	89.57	83.22	88.92
MostFreqAlignBaseline	23.23	20.34	32.78	27.25	21.06	24.93
MostFreqBaseline	31.69	14.17	40.02	6.63	20.04	22.51

Table 3.3: Out-of-five (OOF) baseline and maximum scores on CL-WSD trial data

the first baseline. However, this is not so with regard to the OOF scores, which are lower than the first baseline for *bank* and *occupation*.

The maximum Best$_{JHG}$ score is 100 by definition. However, the maximum OOF score varies per word depending on distribution of translations in the gold standard. This drawback of the OOF score was discussed in Section 3.3.1. Maximum OOF scores on the CL-WSD data for German vary between 82 and 96 percent.

Chapter 4

Vector Space Modelling

4.1 Introduction

One of the major challenges in machine translation is that, according to a bilingual dictionary or some other translation model, a source language word can often have several translations in the target language. For instance, the English word *knight* may be translated as the Dutch word *ridder* in the context of medieval history, but as *paard* in the context of a chess game. Determining the correct translation in a given context is called called Word Translation Disambiguation (WTD).

WTD can be regarded as a ranking and filtering task. It is akin to word glossing or word-for-word translation provided that all translations candidates can always be retrieved from a bilingual dictionary. This is therefore different from full word translation, because it is assumed that all possible translations are given in advance, which is not the case in the more general task of full word translation. Full word translation can be regarded as a two-step process: (1) generation of word translation candidates, followed by (2) word translation disambiguation. Full word translation thus requires an extra step in which translation candidates are generated. Solving WTD would nevertheless partly solve full word translation and is therefore worthwhile to pursue.

One of the features which distinguishes the PresesMT approach to MT from mainstream statistical MT is that it tries to avoid relying on large parallel text corpora for training purposes, a resource that is both scarce and expensive. Instead, it aims at learning patterns in the source and target language, and the mapping between them — from large annotated monolingual corpora only. In a similar vein, most empirical approaches to WTD crucially depend on word-aligned parallel text (cf. Chapter 2). In contrast, our goal is to develop data-driven methods for WTD that do not require any parallel text, but rely solely on the combination of bilingual dictionaries and large-scale monolingual corpora. Even though it is unrealistic that such methods would exceed those based on parallel text in terms of performance, we ultimately aim to bridge the gap in performance between the two.

The basic idea underlying the approach described in this chapter is simple. Suppose we have the English sentence *The knight left the castle* and we want to translate the English word *knight*

into Dutch. We have a machine-readable English-Dutch translation dictionary at our disposal which tells us that the corresponding translation is either *ridder* or *paard*.[1] Furthermore, we have access to a corpus of Dutch text from which we retrieve sentences containing either *ridder* of *paard*. Suppose we find *Kasteel Ammersoyen was eigendom van ridder Floris* and *Het witte paard gaat naar veld f4*. Next we look for the Dutch sample sentence which most closely matches our English sentence, or more precisely, the Dutch sample of which the context of *ridder/paard* most closely matches the context of *knight*. Obviously, directly matching English to Dutch contexts is not going to work, so we first translate the input context from English to Dutch. Given the intention in PreseMT to limit resources to monolingual corpora and bilingual dictionaries, we do not use an MT system to translate contexts, but rather carry out a word-for-word translation by dictionary look-up. Literal translation of the Dutch samples above gives us *castle Ammersoyen was owned by knight Floris* and *the white horse goes to square f4* respectively. We can now conclude that the first translated sample is more similar to our English input than the second one, because they share the word *castle*. As the first sentence is a sample for translation candidate *ridder*, we consider this as support for translating *knight* as *ridder* rather than *paard* in the given context.

Evidently this outline of the appraoch is a huge simplification which abstracts away from many important questions. For instance, word-for-word translation of the context is in itself very likely to contain translation ambiguity. At the heart of the matter is how to determine similarity between input context and sample context.Since this is a key issue in many NLP tasks, numerous approaches have been proposed in the literature, ranging from simple measures for word overlap and approximate string matching (e.g., Navarro, 2001), through WordNet-based and corpus-based word similarity measures (e.g., Mihalcea et al., 2006), to elaborate combinations of deep semantic analysis, word nets, domains ontologies, background knowledge and inference (e.g., Androutsopoulos and Malakasiotis, 2010).

The approach to similarity we take here is that of Salton's (1989) Vector Space Models (VSM). These models were orginally developed in the context of Information Retrieval in order to find documents in a document collection which match a given user query. The same idea has been applied to find semantically similar words, commonly known as Distributional Similarity Models or Word Space Models (Dumais et al., 1997; Schütze, 1998). Good introductions to VSM are given by, e.g., Manning and Schütze (1999), and in Stefan Evert's tutorials.[2] These models are based on the assumption that the meaning of a word can be inferred from its usage, i.e., its distribution in text (Harris, 1954). That is, words with similar meaning tend to occur in similar contexts. This idea has a long tradition in Linguistics, as exemplified by Firth's (1957) famous statement "You shall know a word by the company it keeps!"

Vector space models for words are created as high-dimensional vector representations through a statistical analysis of the contexts in which words occur. Similarity between words is then defined as the similarity between their context vectors in terms of some vector similarity measure, usually cosine similarity. A major advantage of this approach to similarity is the balance of reasonably good results with a simple model. What makes it particularly attractive in the context of PreseMT is that it does not require any external knowledge resources besides a large text corpus and that it is fully unsupervised (i.e., no need for human annotation).

[1] In reality there are more translation candidates, but for the sake of exposition we assume there are just two.

[2] `http://wordspace.collocations.de/doku.php/course:start`

The way we apply vector space modelling to disambiguation is as follows. First training and test instances are converted to feature vectors in a common multi-dimensional vector space. Next this vector space is (optionally) reshaped by applying one or more transformations to it. The motivation for a transformation can be, for example, to reduce dimensionality, to reduce data sparseness or to promote generalization. Finally, for each of the vectors in the test corpus, the n most similar vectors are retrieved from the training corpus using cosine similarity, and translation candidates are predicted from the target words associated with these vectors.

Since at the start of this work the PresesMT MT system was not sufficiently developed to serve as a test platform for WTD experiments, we reused the framework from the SemEval 2010 Cross-Lingual Word Sense Disambiguation (CL-WSD) task, as described in Chapter 3. Moreover, this dataset provides multiple translations per source word, which alleviates the general problem in evaluation of MT that there is usually more than one correct way to translate a word. Work reported in this chapter concerns the English-to-German part of the Cross-Lingual Word Sense Disambiguation task.

The remainder of this chapter is structured as follows. Section 4.2 describes general data collection and processing: how training data was sampled from text corpora and annotated with linguistic tools, as well as derivation of test data from the Semeval data. The creation of vectorized train and test corpora is described in Section 4.2.2. Next Section 4.4 explains corpus transformations, translation prediction and scoring. Results from experimental evaluation are presented in Section 4.5.

4.2 Data collection and preprocessing

In order to reuse the data from the Semeval CL-WSD task, some modifications are necessary. As the task provides no training data, this needs to be to collected in some other way. In addition both training and test data have to be linguistically preprocessed. This section describes data collection and preprocessing for the German part of the CL-WSD data set in order to obtain training and test data for VSM-based WTD.

4.2.1 Construction of training data

The construction of training data involves three steps: extracting translation candidates, retrieving translation samples, and tagging and lemmatizing the samples. More in detail, these steps entail the following.

Step 1: Extract translation candidates
The SemEval CL-WSD task is essentially a word translation task which involves two subtasks:

1. finding translations candidates;
2. ranking and filtering translation candidates.

The WTD task equals subtask 2, so this work abstracts away from subtask 1 by assuming that a perfect solution to finding translation candidates already exists. This amounts to assuming that all translation candidates are present in the translation dictionary. In practice this is accomplished by extracting all possible translations from the gold standard. For the English lemma *bank*, for instance, the translation candidates extracted from the trial gold standard for German are[3]

```
bank, bankanleihe, bankanstalt, bankdarlehen, bankengesellschaft,
bankensektor, bankfeiertag, bankgesellschaft, bankinstitut, bankkonto,
bankkredit, banknote, blutbank, daten, datenbank, datenbanksystem,
euro-banknote, feiertag, finanzinstitut, flussufer, geheimkonto,
geldschein, geschäftsbank, handelsbank, konto, kredit, kreditinstitut,
nationalbank, notenbank, sparkasse, sparkassenverband, ufer, weltbank,
weltbankgeber, west-bank, westbank, westjordanien, westjordanland,
westjordanufer, westufer, zentralbank
```

Step 2: Retrieve translation samples
For each of the translation candidates, we collect examples of its use in context. These context samples are retrieved from a large annotated text corpus in the target language. For German, we use the DeWac corpus which contains over 1.6 billion words, as presented by Baroni et al. (2006). This corpus was made available through the SketchEngine within the context of the PRESEMT project (Kilgarriff et al., 2004). We used the Python bindings to the Sketch Engine's backend – Manatee – to find occurrences of a particular translation in the DeWac corpus and to retrieve German sentences containing this word. Some examples for *Bank* (financial institute):

```
Zur Zeit gibt es insgesamt elf Geschäfte sowie zwei Banken und neun
Restaurants in den Terminals.
```

```
Einem Zeitungsbericht zufolge sucht die Deutsche Bank im Auftrag von Stada
bereits nach einem geeigneten Käufer.
```

and for *Ufer* (river bank):

```
Taucht bis ihr einen Felsen am linken Ufer seht.
```

```
Bei etwas über 8 Metern tritt der Rhein in Beuel über die Ufer.
```

Separate sets of sentence contexts were collected, both based on the occurrence of the word form and on the matching lemma. Most of the 243 different gold terms (in total, for the five head words shown in Table 4.5, i.e., Bank, Movement, Occupation, Passage, and Plant)

[3]Note that the possible German translations of the English word *bank* include all translations of English compounds containing *bank*. For instance *datenbank* as translation of *data bank*, *blutbank* as translation of *blood bank*, and so on. This peculiarity is due to design decisions by the creators of the CL-WSD data set.

Table 4.1: Frequency bins for the CL-WSD German gold terms collected from DeWaC, retrieved on the basis of their word form or lemma.

	0-10	11-100	101-100	1000+
Word	39	27	55	122
Lemma	40	21	52	130

are found in the corpus — around 15 have a frequency of 0, but otherwise the frequencies naturally vary substantially. Frequency bins are shown in Table 4.1. Some of the terms have a large frequency in the corpus, often more then 500,000 occurrences. We sampled up to 5000 random sentence contexts to avoid collecting an excessive amount of data. In our subsequent experiments, we used samples retrieved on the basis of the source lemma, because the corpus coverage is slightly better for lemma than for word form.

Step 3: Tag and lemmatize samples
Sample sentences are tokenized, POS tagged and lemmatized using the TreeTagger for German (Schmid, 1994). Example of tagger output:

```
Bei     APPR    bei
etwas   PIS     etwas
über    APPR    über
8       CARD    8
Metern  NN      Meter
tritt   VVFIN   treten
der     ART     d
Rhein   NE      Rhein
in      APPR    in
Beuel   NN      <unknown>
über    APPR    über
die     ART     d
Ufer    NN      Ufer
.       $.      .
<s>
```

By now, the DeWac corpus has been reliably tagged and lemmatized using the TreeTagger as part of the work in WP3.1, so this step is in principle no longer required. Instead lemma and POS tags can be directly obtained from the corpus.

4.2.2 Construction of test data

Construction of test data takes the following steps:

Step 1: Tag and lemmatize
English sentences from the CL-WSD trial or test data are tokenized, POS tagged and lemma-

tized using the TreeTagger for English (Schmid, 1994). The lemmas and POS tags are required to perform the next step.

Step 2: Word-for-word translation of context
The trial/test data consists of English words in context, whereas the training data consists of German words in context. Hence if we want to match a test instance to the most similar training samples, we need to overcome the difference in language. This can be accomplished in two ways: either translate the context of a test instance to the target language (English to German in this case), or translate the context of all training instances to the source language (German to English in this case). The first option involves less work, because the test data set is much smaller than the training data set. The second option would be faster in a real online WTD system, because translation of the training data can be done off-line in advance. In our experiments, we have used the first option and translated the contexts of test instances.

Given the intention in the PresemT project to limit the resources for MT in general – and therefore also for resources used in WTD specifically – to monolingual corpora and bilingual dictionaries, we do not use an MT system to translate contexts, but rather carry out a word-for-word translation by lookup in a bilingual dictionary. For English to German translation, we currently use a reversed version of the GFAI dictionary (an extension of the Chemnitz dictionary with over 900K entries), which is also used in the PresemT online MT system. Translations are looked up for both the word form and the lemma. In case multiple translations for a word are found, simply all alternative translations are included. POS information is currently not exploited for look-up, but may be explored in future research.

Below is a truncated example of a word-for-word translation of a test instance:

```
The          die,der,dat,dem,den,das
Office       Behörde, Offizium, Dienststelle,
             Dienst, Amtsstube, Kontor, Aufgabe,
             Funktion, Posten, Schalter,
             Dienstraum, Ausgabe, Schreibbüro, ...
may          kann, dürfen, kannst, möge, können,
             dürft, mag, darfst, Weiβdornblüte,
             könnt, darf
also         noch dazu, des Weiteren, ebenso,
             ebenfalls, auch, auβerdem, ooch,
             ferner, und auch, des weiteren
make         Marke, Erzeugnis, Herstellung,
             Faktur, Machart, Fabrikat
available    lieferbar, frei, zur Verfügung
             stehend, abkömmlich, zugänglich,
             benutzbar, abrufbar, nutzbar,
             erhältlich, greifbar, vorgelegen,
             disponibel, vorhanden, ...
```

4.2.3 Dictionary coverage

The project has three sets of English-German dictionaries available:

Table 4.2: GFAI dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts.

Source	Annotator coverage	Word coverage
Bank	59/186	4/41
Movement	32/225	2/75
Occupation	145/220	7/28
Passage	77/195	8/43
Plant	99/237	10/61

Table 4.3: CC dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts.

Source	Annotator coverage	Word coverage
Bank	41/186	2/41
Movement	22/225	1/75
Occupation	147/220	8/28
Passage	80/195	13/43
Plant	83/237	7/61

1. the freely available CC dictionary, which is an internet-based German-English and English-German dictionary based on user generated word definitions. It is available at `http://www.dict.cc/`.

2. the Chemnitz dictionary, which is an electronic German-English dictionary containing over 470 000 word translations. It is GPL licensed and available at `http://dict.tu-chemnitz.de/`.

3. the GFAI dictionary, which is a substantially extended version of the Chemnitz dictionary providing over 900 000 entries.

We have performed a study of how these dictionaries cover the SemEval target word clusters, as shown in Tables 4.2, 4.3 and 4.4 on Page 27. The results are generally positive, with the best quality dictionary covering nearly all the terms considered 'modes' in the SemEval trial data, and generally the dictionaries cover the top end of terms when ranked according to annotator agreement.

The GFAI dictionary generally has the highest coverage, with the exception of the CC dictionary covering a larger number of target terms for the source lemma *Plant*.

The coverage of annotator modes and the target terms with highest inter-annotator agreement shows that the SemEval trial and evaluation data may be suitable for judging the quality of the full WTD system.

Table 4.4: Chemnitz dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts.

Source	Annotator coverage	Word coverage
Bank	48/186	2/41
Movement	22/225	1/75
Occupation	90/220	3/28
Passage	54/195	5/43
Plant	72/237	4/61

4.3 Creating corpora

In order to create the corpora, we first need to create the vocabulary, and can then move on to creating both the training and the test corpora, as follows.

Step 1: Create vocabulary
Given the joint set of samples for all possible translations of a particular source word (e.g., *bank*), we create a vocabulary. The vocabulary can be regarded as the features which model the context of a target word. They help to discriminate among translation candidates. There are many ways to create a vocabulary. The one used so far in PresemT is rather pragmatic and straight forward. To begin with, we work with the lower-cased lemmas as provided by the tagger, only backing off to lower-cased token when the tagger fails to provide a lemma. The vocabulary thus initially consists of all lemma types occurring in all the samples. Next, all function words are removed on the basis of the POS tag. Additionally, all words below and above certain frequency cut-offs are removed. As in Text Retrieval, the assumption is that very high-frequent words have little discriminative power, whereas the contribution of very low-frequent words will be insignificant. The exact values of these two thresholds are experimental parameters. Finally, each vocabulary term is mapped to a unique integer id for efficient storage.

Step 2: Create training corpus
Each context sample is converted to a labelled feature vector. The vector's features correspond to the vocabulary terms and their values correspond to the number of times that a particular term occurs in the given sample sentence. The class label is the correct translation. This results in a training corpus of (sparse) labelled feature vectors like this:

```
0, 0, 0, 1, 0, 0, 2, 0, 0, ..., 0, 0, 0, bank
0, 0, 1, 0, 0, 0, 0, 0, 0, ..., 0, 0, 0, bank
0, 0, 1, 0, 0, 0, 0, 1, 0, ..., 0, 0, 0, bank
...
0, 0, 0, 1, 0, 0, 0, 0, 0, ..., 0, 0, 0, ufer
1, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0, 1, 0, ufer
0, 0, 0, 1, 0, 0, 0, 0, 0, ..., 1, 0, 0, ufer
...
```

Step 3: Create test corpus
Finally each word-for-word translated source word context is converted to a feature vector in the same way as for the training samples, using the same vocabulary, resulting in a test corpus of (sparse) feature vectors. The only real difference is that the class label — that is, the German translation of the focus word — is unknown. In addition, test vectors tend to be somewhat denser, because all possible translations of context words are included during the word-for-word translation.

4.4 Prediction

The translation candidates are predicted from the target words associated with the vectors in the training corpus. Prediction with a vector space model takes the following steps:

Step 1: Construct corpus transformation
The training corpus is used to construct a transformation that transforms a corpus from one vector space to another, possibly with a lower dimensionality than the original corpus. We currently use the Gensim toolkit (Řehůřek and Sojka, 2010) also for this step. It supports several types of transformations, whereas others (e.g. PMI and SUM) were newly implemented.

- The TF*IDF ("term frequency times inverse document frequency") transformation (Jones, 1972) is a well-known feature weighting scheme from Information Retrieval which gives more weight to frequent terms within a single document, while at the same time reducing the weight of terms occurring in many other documents. In terms of the PreseMT WTD task, it means that words occurring in many contexts receive less weight than those occurring in only a few contexts. However, as this is completely unrelated to the class label, it may actually reduce the weight of discriminative words.
- Pointwise Mutual Information (PMI) is another weighting method commonly used in vector space models for word similarity Church and Hanks (1990). It measures the association between translations candidates and context terms, and should give higher weight to terms with more discriminative power.
- Latent Semantic Indexing (LSI) reduces the dimensionality of the vector space by applying Singular Value Decompostion Deerwester et al. (1990). It is claimed to model the latent semantic relations between terms and address problems of synonymy and polysemy, hence increasing similarity between conceptually similar context vectors, even if those vectors have few terms in common Dumais et al. (1997).
- Random Projection (RP), also known as Random Indexing (RI), is another way to reduce the dimensionality of the vector space by projecting the original vectors into a space of nearly orthogonal random vectors. RP is claimed to result in substantially smaller matrices and faster retrieval without significant loss in performance Sahlgren and Karlgren (2005).
- Summation (SUM) sums all context vectors for the same translation candidate, resulting in a centroid vector for each translation candidate. In contrast to LSI and RP, it reduces

the number of vectors (rows) rather than the number of dimensions (columns). It is attractive from a computational point of view because the resulting matrix becomes relatively small.

Step 2: Transform corpora
One or more corpus transformations are used to transform both training and test corpora, resulting in feature weighting and/or dimensionality reduction.

Step 3: Index training corpus
The training corpus is indexed to facilitate fast search for similar vectors. This is primarily an optimization step.

Step 4: Translation prediction
For each unlabeled vector in the test corpus (corresponding to an English word), the training corpus is searched for the most similar vectors and the associated labels provide the translations. Cosine similarity is used to calculate vector similarity. For scoring on the Best measure, the single best matching vector in the training corpus is used. For scoring on the Out-Of-Five measure'q, taking the top five does not work, because all five may have the same label. Therefore first the n best matching vectors are retrieved (by default $n = 1000$ in our experiments). Next the cosine similarities of all vectors with the same label are summed and the five labels with the highest summed cosine similarity constitute the predicted translations.

Step 5: Scoring
Scoring compares predictions against the gold standard files and outputs a number of scores. Initially the orginal CL-WSD scoring Perl script was applied, but this has been superseded by our own implementation which in addition to the orginal scoring measures calculates a number of other evaluation measures, including the preferred Best_{JHG}. (cf. Section 3.3.3).

4.5 Experimental results

This section reports experimental results on the CL-WSD trial data for German. Results for different models in terms of the Best_{JHG} score and Out-of-five scores are listed in Table 4.5 and Table 4.6 repectively. Several general observations can be made. To begin with, the scores on *passage* tend to be lower than those on *bank*, *occupation* and *plant*. To a lesser extent, the same holds for scores on *movement*, keeping in mind that max OOF score on movement is also lower. It seems there is no correlation with the number of translation candidates though, as *passage* has 42 whereas *bank* and *plant* have 40 and 60, respectively.

Furthermore, even though most models often outperform both baselines on some words, there is no model that consistently outperforms both baselines on all five words. Although the SumModel comes close, it has a problem with *passage*. In general, all models have their lowest score on *passage*. Looking at the mean scores over all five words, however, the SumModel outperforms both baselines. This is a promising result considering that the SumModel is the

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
RP (300)	15.83	17.50	11.25	5.42	20.00	14.00
LSI (200)	30.42	11.25	21.25	9.17	20.42	18.50
SumModel	**43.75**	17.50	**37.92**	7.92	**43.75**	**30.17**
PMI	32.08	**21.25**	26.67	2.92	38.33	24.25
TF*IDF	20.00	11.67	35.83	3.33	23.33	18.83
BareVSM	28.33	10.00	37.08	9.58	17.08	20.42
MostFreqAlignBaseline	6.25	19.17	35.83	**15.00**	40.00	23.25
MostFreqBaseline	1.25	5.00	2.50	1.67	10.26	4.14

Table 4.5: Best$_{JHG}$ scores for different VSM models on CL-WSD trial data (underlined=above both baselines; bold=highest)

smallest and does not rely on parallel text. This in fact prompted us to choose the SumModel for implementation in the PresemT MT system.

In a similar vein, no model consistently outperforms all others. For instance, even though SumModel yields high OOF scores on four out of five words, PMI scores higher on *plant*. LSI seems to provide no improvements over the BareVSM. RP performed badly, which may be related to implementation issues.

TF*IDF seems to give slightly worse results in comparison to BareVSM. A possible explanation is that its feature weighting is unrelated to vector labels, so it may actually reduce the weight of discriminative context words. PMI, which does take the vector label into account, gives a slight improvement over BareVSM on the Best$_{JHG}$ score. PMI is known to be deficient for rare words, which may explain the lack of a major result.

We also ran experiment combining transformations for feature weighting and dimesionality reduction, such as a combination of TF*IDF and LSI, but this yielded results worse than the basic models presented here.

To sum up, the results on WTD with vector space modelling are fairly good. The first baseline of always choosing the most frequent translation candidate is easily surpassed. Even the second baseline, which required word-aligned parallel text corpora, is often exceeded, although not consistently. The rather large range of VSM scores suggests that the approach holds more potential, but that the factors determining performance are not yet well understood. Evaluation on more data may shed light on these issues.

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
MaxScore	95.60	82.62	93.58	89.57	83.22	88.92
RP (300)	24.80	12.65	22.70	8.82	21.63	18.12
LSI (200)	47.07	12.61	35.40	17.03	35.61	29.54
SumModel	**52.59**	**28.01**	**42.03**	17.72	32.54	**34.58**
PMI	41.00	16.33	38.41	15.47	**38.52**	29.95
TF*IDF	37.76	12.31	27.72	12.16	25.00	22.99
BareVSM	47.88	13.86	40.83	14.60	28.33	29.10
MostFreqAlignBaseline	23.23	20.34	32.78	**27.25**	21.06	24.93
MostFreqBaseline	31.69	14.17	40.02	6.63	20.04	22.51

Table 4.6: Out-of-five (OOF) scores for different VSM models on CL-WSD trial data (underlined=above both baselines; bold=highest)

Chapter 5

Statistical N-gram Modelling

5.1 Introduction

As discussed in Chapter 1, the Corpus Modelling Module is aiding the Translation Equivalent Selection Module in such tasks as translation disambiguation and word ordering. Given that this is ongoing research, the need for language models of various sorts (lemma-based, word-based, POS-based or combinations thereof) may change. The establishment of a framework which allows for the rapid creation of new large language models of high order is therefore a contribution to the project even if the language models built in the development phase might not be used in the final version of the PreseMT system.

The main challenge is the scale of the models, as they are trained on text corpora comprising billions of words. This means that substantial CPU time and memory are required. An efficient solution to this was found in parallelization. The first part of this Chapter describes our approach to fast construction of huge n-gram models. This is followed by an intrinsic evaluation of the models.

The second part of this Chapter presents an application of language models to the WTD task. We present results on WTD using the Semeval CL-WSD data.

5.2 Methodology

The n-gram models are built with the standard tool IRSTLM (Federico and Cettolo, 2007). With large amounts of data this poses challenges in terms of speed and storage, but it lends itself well to data parallelization. It was decided to adapt IRSTLM scripts to the OpenPBS queue handler (a system which distributes jobs to a cluster) and create SRILM scripts to do the same.

The alternative to the adaptation of present tools would be implementing a new language model framework. Even though conceptually simple, it would still involve a reasonably large (and somewhat wasted) effort to create a fully-fledged tool with the state-of-the-art functionality offered by the two aforementioned frameworks.

NTNU has access to a cluster, Kongull, which is a 96 node cluster partitioned in equal parts of nodes with 48G and 24G RAM. The cluster uses a Linux operating system, with the OpenPBS[1] job scheduler.

The IRSTLM software package already had scripts for parallel treatment of data developed for another (closed) version of the PBS system, and this was changed to adhere to the slightly different syntax of OpenPBS. The parallelization step works as follows:

1. A dictionary is compiled for the whole input corpus.
2. The corpus is sectioned into n sections according to word frequency.
3. N-grams are counted for each section.
4. (Sub-)LM scores are computed.
5. Files are merged into one LM .

Steps 3 and 4 are the steps that are carried out in parallel on each node. A bash script submits the jobs to the PBS queue and tells the jobs to delay merging until all jobs have successfully finished.

IRSTLM also uses scripts to section up the building of the LMs because of resource constraints, but doing this serially. It was therefore easy to ensure that the parallel processing gave the same output, and assess the speedup factor (which also would be affected by other uses of the cluster).

5.3 Corpora

In the development of the corpus modelling scripts, three corpora of German (17GB), Italian (13GB), and English (33GB) were used. The corpora were mined from the web in Task T3.1 by Masaryk University. As the process of parallelizing the creation of the n-gram models involves sharding (dividing the corpus into parts) the corpora and counting n-grams for each shard, it was necessary to test with the full versions to ensure data integrity when merging large files. In development some errors related to file locking did not appear unless a big file was input. The corpora are presented in a multi-token format, presenting the wordform, lemma, and part of speech (POS), all of which could be extracted to build n-gram models over.

5.4 Language Models

The IRSTLM framework can output LMs in an internal format, the ARPA LM format, as well as a compiled version for quicker access with IRSTLM tools (the local platform is Linux/amd64, but the compile step can be done on any architecture).

[1] `http://www.mcs.anl.gov/research/projects/openpbs/`

The German data was also filtered through the tokenizer from the TreeTagger (Schmid, 1994), as well as cut at length 50. Lemma-based LMs were also filterted through a set of tokenization steps, rules that filter out apparent noise and words that have been split in two by mistake (typical German prefixes). (The web corpus includes multiples of words and special characters of arbitrary length.) The corpus was not lowercased, and still contained a lot of noise, as words beginning with special characters (i.e., "-Bus", etc). A LM for a 3Bn word corpus was built in half a day with this infrastructure (depending on system load).

N-gram models of various sizes and nature (built over words, lemmas or POS) are unavoidable baselines when building novel models of language. The LMs currently used in the PRESEMT system are lemma-based.

5.5 Intrinsic evaluation

As the performance of the language models will have to be measured relative to the purpose for which they have been created (MT in our case), they are not currently evaluated. However, some form of intrinsic evaluation can be carried out on held-out portions of the corpus. A sample is given by the following statistics for the German corpus, as obtained by the IRSTLM evaluation facility (Federico et al., 2010):

$$\begin{aligned} N_w &= 316,521,965 \\ PP &= 2718.03 \\ PP_{wp} &= 328.11 \\ N_{oov} &= 2,339,456 \\ OOV &= 0.74 \end{aligned}$$

Where N_w is the total number of words in the evaluation corpus, PP is the perplexity, and PP_{wp} reports the contribution of out-of-vocabulary (OOV) words to the perplexity. The out-of-vocabulary word term OOV is defined as $N_{oov}/N_w * 100$, with N_{oov} being the number of OOV words. It is interesting to note that only 0.74% out-of-vocabulary words are obtained on an enormous corpus, even without removing capitalization.

In addition to this, some statistics on the dictionary creation can be retrieved based only on the input corpus, as shown in Table 5.1 on Page 36, where a dictionary of size 898,720 was induced from the in total 29,693,694 words in the original German corpus. The first three columns of the table show the percentage of words in the training corpus whose frequencies are over 0 (all of them, 100%), over 1 (40%), etc.

5.6 Disambiguation with N-gram models

As an alternative approach to WTD using VSM, we tried WTD using n-gram models. Utilising n-gram language models (LMs) to rank target contexts is motivated by their widespread use

Table 5.1: Dictionary growth curve

Freq	Entries	Percent	Freq	OOV on Test
0	898,720	100.00%	<1	3.86%
1	368,359	40.99%	<2	4.88%
2	249,347	27.74%	<3	5.57%
3	194,059	21.59%	<4	6.11%
4	161,028	17.92%	<5	6.56%
5	138,463	15.41%	<6	6.97%
6	122,156	13.59%	<7	7.33%
7	109,814	12.22%	<8	7.65%
8	99,917	11.12%	<9	7.95%
9	92,057	10.24%	<10	8.23%

in NLP and MT. The advantage of n-gram modelling is its conceptual simplicity and practical availability: only one model is needed to process all trial and test words once the model is built and made available.

5.6.1 Method

Adapted to the WTD task, a LM can predict the likelihood of a target context being part of the language. TC sentences are constructed by combining each TC with every possible translation of their context. The shortest TC sentence is the TC itself, and if the LM is queried for all TC candidates, the most frequent would turn out on top. For the English *bank*, the most likely German candidate is *Bank*. The n-gram model should rank TC sentences of the right sense higher, because co-located phrases like *the West Bank* and *Gaza Strip* are reflected in higher n-gram probabilities of their corresponding TC sentences. This applies when the n-gram model finds the TC with the content-bearing word in the right place (when word-to-word translation is correct), unlike for multi-word expressions with different surface forms in German and English.

The LM was built from sentence-separated lemmatised parts of DeWac, a large monolingual web corpus of German containing over 1,627M tokens Baroni and Kilgarriff (2006). For each TL context, a huge number of n-grams to query the model were compiled. With a 5-gram model, a possible 4 words preceding and succeeding the word to be translated could be tested. The results of various context lengths were kept in a 2-dimensional matrix, where each index represents words ahead of, and after the TC word. Results from different context lengths are extracted, until enough TC are found (often 5). If the [-4,1] entry (4 words before, 0 after) is ranked highest, the TC represented by these n-grams would be used exclusively in output, if the limit was reached. If not, the algorithm moves on to the next matrix entry. Because of the naïve word-by-word translation, few n-gram candidates of higher order were found. Ranking by no surrounding context leads to the same answer for all instances of the word, with the most frequent TL sense first.

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
5-gram model	25.00	12.92	27.08	14.17	15.42	18.92
3-gram-model	10.00	16.67	24.17	11.67	6.67	13.84
1-gram-model	**42.50**	5.00	2.50	1.67	3.33	11.00
MostFreqAlignBaseline	6.25	**19.17**	**35.83**	**15.00**	**40.00**	**23.25**
MostFreqBaseline	1.25	5.00	2.50	1.67	10.26	4.14

Table 5.2: Best$_{JHG}$ scores for word translation disambigution on CL-WSD trial data with n-gram langaue models (underlined=above both baselines; bold=highest)

	Bank	*Movement*	*Occupation*	*Passage*	*Plant*	Mean
MaxScore	95.60	82.62	93.58	89.57	83.22	88.92
5-gram model	**31.75**	**23.01**	**37.73**	15.06	**26.55**	**26.82**
3-gram model	27.14	23.01	36.81	17.70	22.16	25.42
1-gram-model	22,92	14.17	24.39	6.63	20.04	17.63
MostFreqAlignBaseline	23.23	20.34	32.78	**27.25**	21.06	24.93
MostFreqBaseline	31.69	14.17	40.02	6.63	20.04	22.51

Table 5.3: Out-of-five (OOF) scores for word translation disambigution on CL-WSD trial data with n-gram language models (underlined=above both baselines; bold=highest)

5.6.2 Results

The simple n-gram model was employed in three different orders, uni-, tri- and pentagram models, but without exploring all possible priorities of context lengths (skewing to before- or after context). Results in terms of the Best$_{JHG}$ are listed in Table 5.2. On average the higher-order models perform better. The 5-gram model beats the most frequent translation baseline, but with the exception of *bank* none of the models surpasses the harder most frequently aligned baseline. Results in terms of the Out-of-five scores are listed in Table 5.3. Again higher order models perform better on average. The 5-gram models outperforms both baselines, except on *passage*.

These results suggest that n-gram language models form a viable approach to word translation disambiguation, outperforming the default strategy of always choosing the most frequent translation.

Chapter 6

Modelling with Self-Organising Maps

6.1 Introduction

In the present chapter, the use of Kohonen's Self-Organising Map (SOM) model in order to model the TL language is discussed. The aim of the work summarized here has been to employ the SOM to determine the semantic relevance of a "candidate" translated term with respect to its context, and thus allow a quantitative comparison among all the available alternatives that are suggested as candidate translations by a bilingual dictionary. SOM (Kohonen, 1997, 1982) is a model of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (usually two-dimensional), discretized representation of the high-dimensional input space of the training samples. SOM differs from other artificial neural networks in the sense that during training it uses a neighbourhood function that is gradually reduced in magnitude, in order to train similarly neighbouring nodes so as preserve the topological properties of the input space. In the chosen approach, the input set consists of multi-dimensional vectors that describe the co-occurrences of encountered lemmas with a well-defined class of representative words. What makes this approach particularly attractive in the context of PreseMT is that in order to model a monolingual corpus it does not require any external knowledge resources besides a large text corpus, the modeling process is fully unsupervised in the creation of the map and that most of the processing is performed off-line. During the actual machine translation process, when this modeling is employed for disambiguation, only the final SOM-generated mapping of words onto the map lattice needs to be accessed. This mapping is small in terms of memory required and can thus be processed very quickly and efficiently.

6.2 Main characteristics of the SOM Model

Self-organizing maps learn to classify input vectors according to their similarity in the pattern space. Thus, self-organizing maps learn both the distribution and topology of the input vectors they are trained on. During training the neuron in the layer that is located closest to an input vectors is selected to adjust its weight vector toward those input vectors. Specifically, the network first identifies the winning neuron (or alternatively Best Matching Unit or BMU) for

each input vector. Then, each weight vector moves to the average position of all of the input vectors for which it is a winner or for which it is in the neighbourhood of a winner. The distance that defines the size of the neighbourhood is altered during training through two phases, the first corresponding to the rough training and the second to the fine-tuning step. During rough training, the input patterns are ordered relative to one another while in the fine-tuning phase the weight vector of each node is fine-tuned to specific patterns.

The neurons in the SOM output layer are arranged in the form of a lattice with either square or hexagonal topologies. In the work summarized here, a hexagonal topology has been used. A new implementation for constructing SOM was developed at ILSP from scratch in Java for the purposes of the PreseMT project. The main reasons for this re-implementation were to employ a common programming language to the rest of the PreseMT architecture and to be able to optimize the performance of the map creation.

6.3 Datasets

The popularity of SOM in terms of numerous, diverse applications is due to its flexibility and efficiency in unsupervised clustering tasks. SOM has been adopted for a variety of experiments involving symbolic datasets. The novelty of the SOM application in the PreseMT project focuses on how it is integrated into a system for machine translation, with emphasis on the task of word translation disambiguation. In this respect, the features used to map the linguistic data to SOM are of particular importance. In the context of language processing, word co-occurrence frequencies at a sentence level have been proposed and the features chosen are frequencies of occurrence of words within the sentences. However, when large document collections are processed in real-world applications, it is virtually impossible to take into account the frequencies of all words in all documents and still process the entire document collection, the limiting factors being both CPU time and memory resources. Consequently, several approaches have been proposed in literature to reduce the number of features, the most common ones being random projection (Kaski, 1998) of the frequency matrix to lower dimension and latent semantic analysis (Deerwester et al., 1990).

6.3.1 Feature extraction

As discussed in Section 4.1, the basic assumption made is that words that occur frequently in similar contexts in natural-language expressions will bear related meanings. More specifically, the contexts considered here are sentences, i.e., text windows from one full stop to the next one. The use of such text windows is based on the hypothesis that the full stop between sentences is the least ambiguous point at which the description of an idea is completed. This basic hypothesis is frequently made in experiments involving word clustering. Ideally, for each lemma, the co-occurrences with all other lemmas would be recorded, although at the cost of a high feature-vector dimensionality. To limit this dimensionality, only a subset of available lemmas was chosen as the feature set, so that every lemma would be described by its co-occurrences with the lemmas from the feature set.

Pareto's principle, also known as the 80-20 rule (Cam and Gilles, 2002) states that 20% of the causes are responsible for 80% of results. Pareto's principle, which is used in the ABC analysis, has mostly been applied to quality control and management tasks. According to the ABC analysis, a portion of the causes is characterized as A, which indicates very important events, with B and C corresponding to less important and to unimportant events, respectively. In the word disambiguation application, category A contains highly frequent lemmas (corresponding to stop-words, such as articles, conjunctions, and auxiliary verbs, as well as other frequent words), B contains relatively frequent lemmas, and C contains rare lemmas. Lemmas from category B are selected for the feature set, since these lemmas do not correspond to very common words (that do not reflect a specialized content) yet are frequent enough to describe all remaining lemmas. Initial limits of the ABC analysis are set to implement an appropriate split of the input data in terms of frequency. For instance, in document organisation applications on the basis of content (Tsimboukakis and Tambouratzis, 2011), the following categories were used.

- Category A contains the most-frequent lemmas that collectively amount to 70% of all occurrences.
- Category B contains lemmas that contribute the next 15% (from 70% to 85%) of all occurrences.
- Category C contains lemmas that correspond to the remaining 15% of occurrences. In addition, to avoid studying exceptionally rare tokens, lemmas that occur less than three times throughout the corpus are omitted.

The co-occurrences of these lemmas with respect to the B lemmas are employed to represent the co-occurrence of words in terms of a numeric vector. More specifically, each lemma from categories A and C is represented by a vector of m elements, each of which indicates the number of times the given lemma co-occurs with the corresponding B lemma.

In order to implement the ABC analysis, initially each lemma's occurrences in the document set are counted. Then, the lemmas are ranked in descending order of frequency. Then category A is created iteratively, by introducing the most-frequent lemma in category A without substitution until the sum of normalized frequencies reaches the threshold of category A. When the sum ranges between thresholds A and B, the corresponding lemmas are assigned to category B and the rest are assigned to category C.

An intuitive representation of this procedure is illustrated in Figure 6.1 where category A is represented by the lemmas that are highly ranked due to their high frequency and thus have no value in computing co-occurrences in terms of these words. These are the lemmas which are ranked higher than the point that corresponds to the intersection of the blue and the green line in Figure 6.1. Next, category B is represented by the middle-frequency words which are useful in terms of both accuracy and coverage. Last, category C contains the rare words which are not useful for the feature vector due to their very infrequent occurrence.

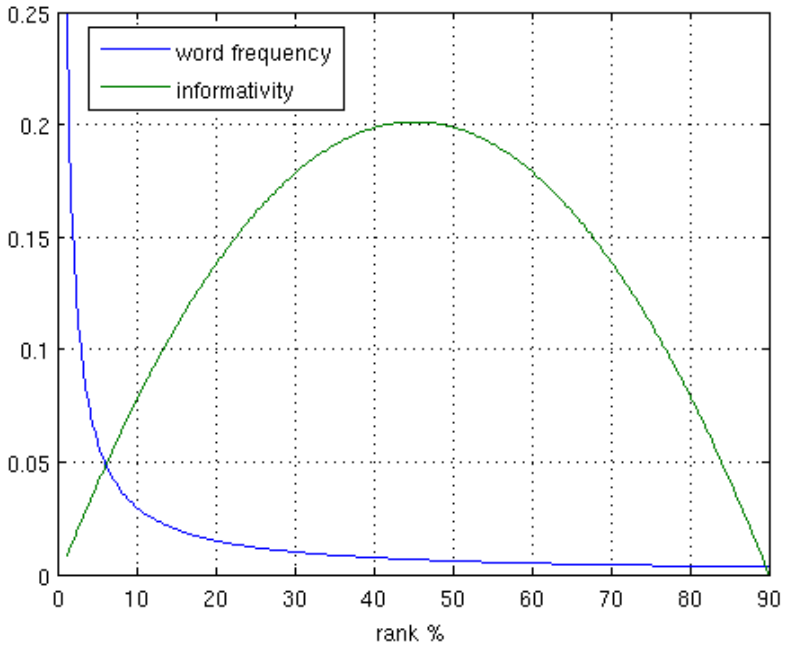

Figure 6.1: Computing the appropriate feature-vector

6.3.2 Corpora

For the purposes of the task of Word Translation Disambiguation, the monolingual corpora which were produced by Masaryk University through web-crawling were used in order to train the SOMs (more information on the corpus creation is provided in PRESEMT Deliverable 3.2.1). In the development of the language model we used subsets of the two large mono-lingual corpora of the PRESEMT target languages, namely German (17GB) and English (33GB).

The SOM models were constructed using the techniques described in the previous sections and parts of the aforementioned data-sets which however never exceeded the size of 1GB. The main reason behind this limitation is the amount of memory that is reserved for computing co-occurrence matrices exceeds commodity hardware specifications by large. Moreover, processing huge corpora has a huge impact on CPU time. For instance, one pass over all input vectors (approximately 0.9GB) during training requires approximately 16 hours for a HP Z800 workstation with 2 quad-core Intel 5585 processors running at 3GHz with 24 GB of memory.

Furthermore, Figures 6.2 and 6.3 indicate how feature-vector and lemma-set cardinality scale over the corpus size. Apparently, they both show a sub-linear behaviour. In order to process the largest possible subset of corpora, it was essential to reduce the size of class B of the ABC analysis. In the present application, to process a total of 900 documents in the German language each of 1 GB, class B was defined by limits of 80% and 82%, i.e., much narrower than those employed by Tsimboukakis and Tambouratzis (2011). In that way, the total number of lemmas used as features was equal to approximately 1,000 lemmas (as an indication, the number of lemmas within class B for limits of 80 and 85% was more than 2,500 lemmas, which was

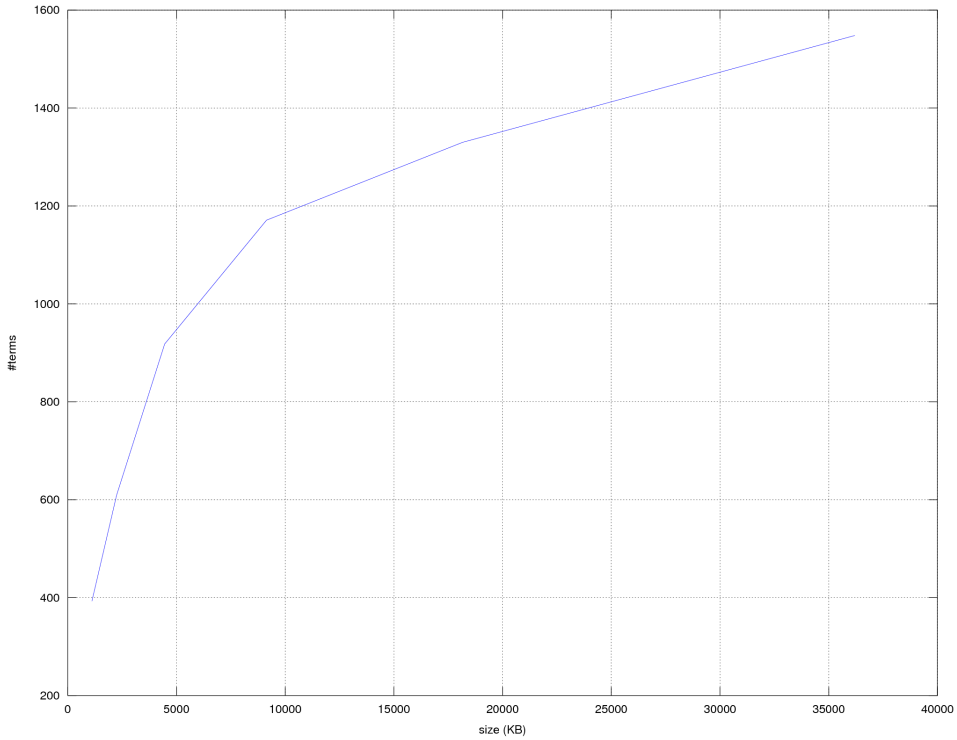

Figure 6.2: Feature-vector over corpus size

beyond the processing capabilities of the aforementioned workstation). Details on countering the implementation limitations are provided in the final section of the present chapter.

6.4 The disambiguation process

Once the aforementioned language model is available, we use our own adaptation of the well-known Viterbi algorithm for the translation equivalent selection module, in order to empower the overall optimal phrase selection efficiently. This task consists in picking one lemma from each set and that way disambiguating multiple translations of single- or multi-words units. Hence, for the purposes of the Translation Equivalent Selection (TES) process, namely the part where disambiguation takes place and the correct translation alternatives are picked, we had to come to terms with the increased perplexity of alternative phrases in the target language, caused by the numerous combinations of equivalent terms, suggested by the Phrase Aligner Module (PAM).

To elaborate, for the i-th alternative term of the phrase in the target language, we compute the transition cost from all the previous possible word forms. We also consider recursively the cost of selecting those forms given previous transitions. Then, from all j different word forms that lead to the i-th term at the k-th position of the phrase, we set $cost(k, i)$ equal to

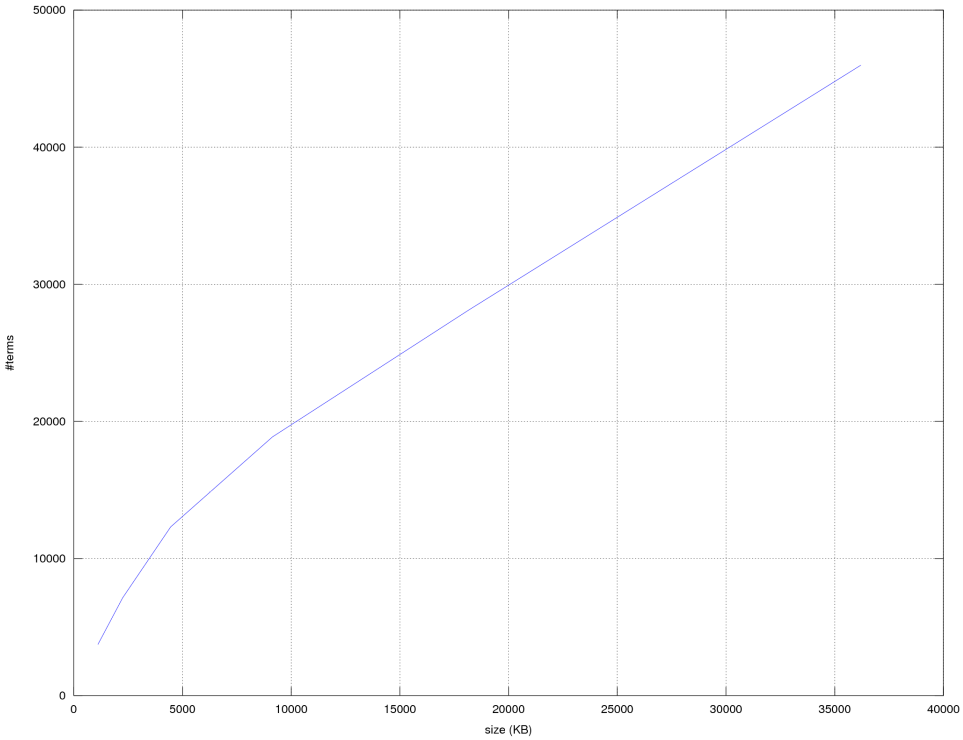

Figure 6.3: Lemmas over Corpus Size

$min_j\{distance(n(k,i), n(k-1,j)) + cost(k-1,j)\}$, where the distance signifies the Euclidean distance of the winner SOM neurons for the corresponding terms. Hence, the optimal path reaching term i at position k contains the optimal sub-path reaching j, and thus, when selecting the next alternative word-form there is no need to expand and compute any redundant suboptimal paths from j as they have been pruned beforehand in order to reduce search costs and overall system response time.

Furthermore, we examined different variations of this approach. The simplest of them tries to resolve all word disjunctions at sentence level with the aforementioned method (this is hereafter denoted as 'sentence'). In addition, a more complex approach that operates in two steps. First, we resolve the disambiguation task among all heads and functional heads of phrases for the entire sentence, and after these have been set all tokens within each phrase are examined, while having the head- and functional head-lemmas fixed form the previous step. This method tries to express the more global role of heads and f-heads as compared to the more local role of other words (and is hereafter denoted as 'FHP').

The disambiguation process has been integrated into the translation engine of the online PRESEMT system. The SOM-based disambiguation module can currently be employed during Greek-to-English and Greek-to-German translation using the generated SOM models for English and German respectively.

Table 6.1: Score table for each disambiguation technique compared to the SemEval baselines

Word	Sentence-based	Phrase-Head based	Baseline
Bank	31.25	31.25	2.49
Movement	30.77	15.00	3.91
Occupation	20.00	20.00	13.45
Passage	15.38	23.08	4.58
Plant	21.43	21.43	11.70

6.5 Experimental evaluation

In order to evaluate our methods we have employed the SemEval platform as described in Chapter 3. Table 6.1 indicates the performance of each of our techniques in terms of the corresponding baseline scores as reported by Lefever and Hoste (2010a). Regarding the metrics used, we use the same evaluation formula as described by McCarthy and Navigli (2007). In essence, high-scores indicate that the system selected a translation that was also picked by more human annotators.

Moreover, by examining Figures 6.4 and 6.5 we obtain an overall image on how our disambiguation process scales up with regard to the iteration cycles of the SOM learning procedure. Specifically, in these pictures we show the average score achieved for our translations according to the SemEval platform. In particular, SemEval examines the translation chosen by the system in twenty sentences for five different words, namely *bank*, *movement*, *occupation*, *passage*, and *plant*. Figures 6.6 and 6.7 show separately the score achieved for each of these terms in isolation. The first 50 iterations constitute the rough training stage, while the remaining 20 iterations correspond to the fine tuning phase of the SOM adaptation process.

The results of Table 6.1 show a substantial improvement of the performance of the implemented disambiguation system with respect to the baselines. More specifically the score for the words *bank* and *movement* is an order of magnitude larger than the baseline and for the rest of the words the baseline is two to four times lower. Notably, for all words the SOM-based system performed better than the SemEval baseline.

Figure 6.4 was created using the SemEval lexicon with words that are not necessarily included in the map of the SOM. In contrast, Figure 6.5 was created using the intersection of PreseMT and SemEval lexica. In other words, according to this scenario, the system selects only translations that exist in both dictionaries, the one used within PreseMT and the one available by SemEval, which can explain the lower variance of the depicted curves compared to Figure 6.4.

On the other hand, Figures 6.4, 6.5, 6.6 and 6.7 imply a non-negative trend of the performance of the SOM-based disambiguator as a function of the number of iterations during training. It is therefore expected that the figures of Table 6.1 will be further improved as the number of iterations increases and SOM reaches an optimal state. It is worth noting that the processing time required to complete one iteration is still high and experimentation with respect to this parameter is consequently time-consuming.

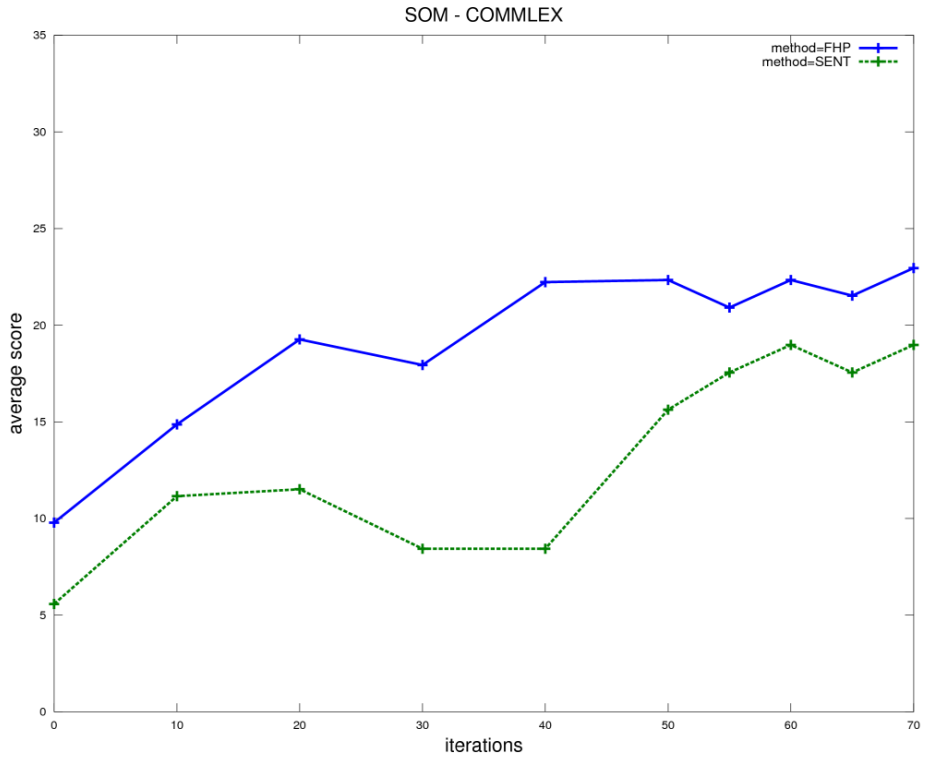

Figure 6.4: SOM results using the intersection of the PRESEMT and SemEval lexicon

Furthermore, our SOM-based approach is extremely efficient in terms of CPU-time and memory requirements with respect to the on-line processing. Specifically, the final map which we constructed from a German corpus of approximately 1GB size requires just 18MB of memory. On the other hand, the corresponding German-English dictionary that is used in order to retrieve all possible candidates of each word occupies 136MB of storage. Tables 6.2 and6.3 provide a thorough look on the performance of the SOM disambiguation method in terms of how much time was spent in the CPU selecting the optimum sequence of translation candidates within a phrase or a sentence, and the time spent to retrieve all these possible candidates from the dictionary. The experiments were run on a standard personal computer. Each row in Table 6.2 corresponds to the total time that was required in order to look up each word from 20 distinct sentences and store them in the appropriate vector elements that represent each structure, either sentences, or phrases, as denoted for each column. Once this information is available we are in a position to efficiently resolve disjunctions for each of our methods. Table 6.3 presents an analytical view of how much time is consumed in this process. Apparently, it is more costly to retrieve all possible translation candidates from the dictionary, rather than to select the optimum sequence of them within a phrase or a sentence. Naturally, the results in Table 6.3 are strongly affected by the number of stored translations for each searched lemma.

As an example, the output of the SOM-based system for an indicative sentence from the SemEval is presented. The sentence in the source language, namely English, is the following:

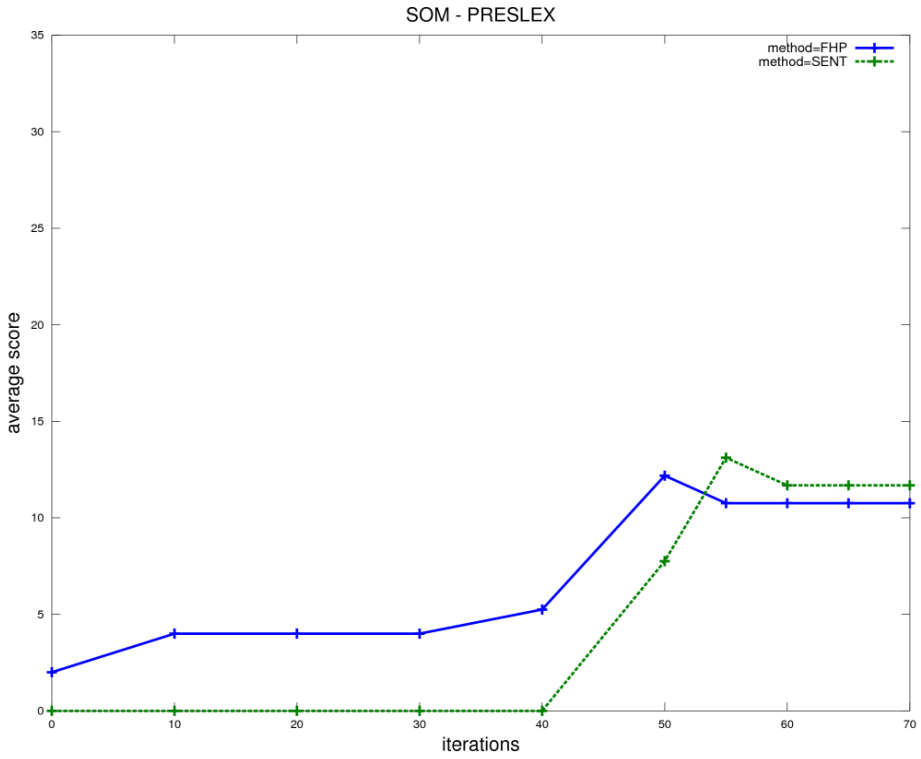

Figure 6.5: SOM results using the PRESEMT lexicon

Table 6.2: CPU time (msec) for resolving disjunctions for the SemEval case study

	Sentences	FHP
Bank	132	116
Movement	38	53
Occupation	44	85
Passage	57	112
Plant	43	36
All (total)	314	402

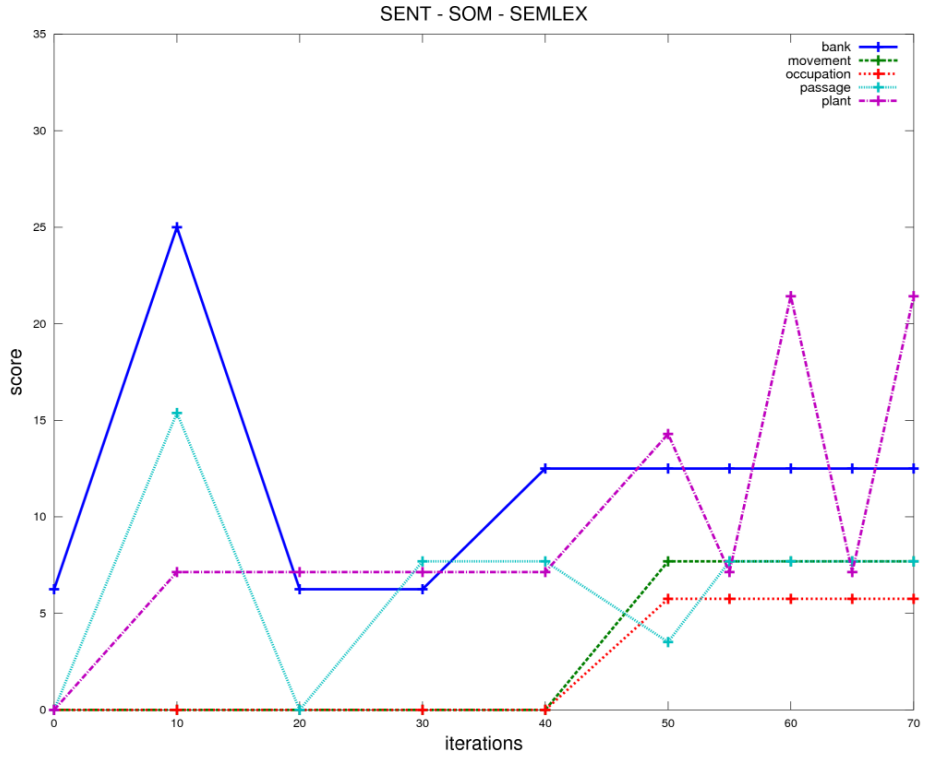

Figure 6.6: SOM results for sentence level disambiguation using the SemEval lexicon

Table 6.3: Time (msec) for retrieving all candidates from the German-English Dictionary for the SemEval case study

	Sentences	FHP
Bank	378	313
Movement	275	261
Occupation	402	422
Passage	232	237
Plant	5078	5013
All (total)	6365	46246

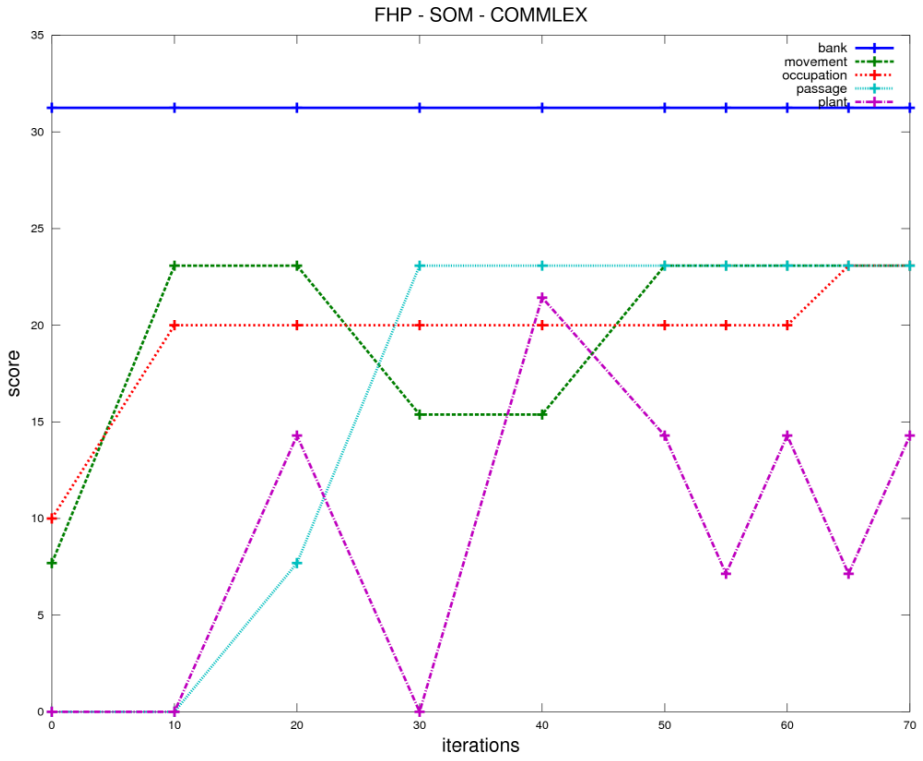

Figure 6.7: SOM results for FHP-based disambiguation using the intersection of the PRESEMT and SemEval lexicon

```
The BIS could conclude stand-by credit agreements with the creditor
countries' central banks if they should so request.
```

Next, the framework can derive the following translations (consisting of lemmas exclusively), depending on the target language and the methods selected. At sentence-level disambiguation, the following output is obtained for the previous sentence:

```
der BIS können ableiten Leistungsbereitschaft Entlastung Abmachung bei der
Gläubiger Staat mittig Bank ob sie sollen also Abrufen.
```

On the other hand, for the FHP-based disambiguation, a different optimal results is obtained, namely:

```
der BIS können zu einem Entschlußkommen Alarmbereitschaft Akkreditiv
übereinstimmung bei der Gläubiger Staat Zentrale Bank sie sollen also
fragen nach.
```

6.6 Future work

There have been two limitations in the effort to process larger monolingual corpora. The first one involves the processing of larger sizes of feature vectors. To achieve that, a possible modification has been identified, to retain for each lemma to be placed in the map only the most important features rather than all features of the feature vector.

The second improvement concerns the ability to train maps more quickly. To that end, and to make use of multi-processor hardware, the existing Java implementation of the SOM training process has been parallelised using Java threads. This new implementation exploits semaphores and mutex locks for synchronization among multiple process threads. It is expected that by porting the SOM training process to all 4 cores of a quad-core CPU such as one single CPU of the aforementioned Z800 workstation, a reduction of the processing time of 50% will be achieved in comparison to the processing time on a single core of the same machine. Figure 6.8 illustrates the gain achieved while launching our multi-threaded implementation for SOM training over a small German corpus of 12MB. There is a substantial gain for just a few additional threads. However, the impact of parallelism diminishes for a higher number of threads due to the synchronization cost between them. In practice, at each iteration, a stage of parallelism is succeeded in turn by a stage of synchronization. Therefore, the more threads there are, the more time the synchronization stage requires compared to the time consumed by the multiple threads. Consequently, beyond some point there is no significant improvement from parallelism. Extensive experiments with this new version of the SOM will be performed in the 3rd year of the project and reported in the relevant deliverables.

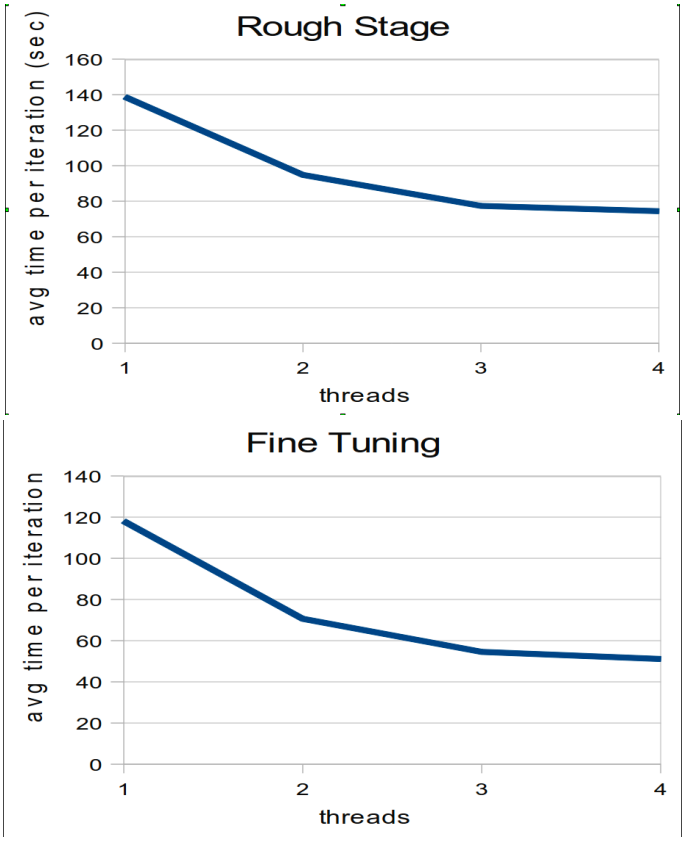

Figure 6.8: Processing time required to perform a complete training iteration for the SOM training process over a small German corpus, as a function of the number of threads used for (a) the rough-training and (b) the fine-tuning stage

Chapter 7

VSM-based disambiguation in the PresesMT MT system

The VSM models for WTD described in Chapter 4 showed potential to improve the translation quality of an MT system. On average, scores were above the simple baseline of choosing the most frequent translation, which is a reasonable baseline in the context of the PresesMT MT system, where no parallel text is available. Some models even outperformed the more challenging baseline of taking the most frequently aligned translation in a parallel text corpus. For this reason, it was was deemed worthwhile to scale up to VSM-based approach to WTD and implement it in the PresesMT system. Whereas the models in in Chapter 4 were limited in that they only targeted nouns present in the Semeval CL-WSD data set for English to German translation, the models described in this chapter cover the most common word translation ambiguities occurring in the bilingual lexica for all translation pairs targeted within the PresesMT projcet.

The corpus modelling module for WTD comprises two parts. The first is an off-line processing step in which translation ambiguities in the lexicon are identified, context samples from a target language corpus are collected and VSM models are constructed. This is described in Section 7.1 below. The second is the WTD module that is part of the on-line PresesMT system. It employs the models from the previous step to perform disambiguation in cooperation with the Translation Equivalent Selection module. This is addressed in Section 7.2. Finally, directions for future work are discussed in Section 7.3.

7.1 Off-line processing

7.1.1 Context sampling

Context sampling is the off-line process of collecting usage samples for all target language words involved in word translation ambiguities in the lexicon. Input to the sampling process consists of a translation lexicon (in the PresesMT XML format) and an indexed monoligual corpora for source and target language. The process proceeds as follows.

Step 1: Counting
Two types of counting are performed on the monolingual corpora for source and target language. The first type is a count of all lemmas occurring in the corpus. The second type is a count of all lempos instances in the corpus, where lempos is a combination of the lemma and a coarse-grained POS tag such as n (noun) or v (verb). Counting is performed using the Manatee corpus management tool provided by LCL/MU, which also forms the backend of the SketchEngine.

Step 2: Finding translation ambiguities
The bilingual lexicon is searched for translation ambiguity, that is, source language entries which have at least two possible translations. Once these ambiguities have been identified, we can retrieve samples for the target words from a target language corpus and use these to build word translation disambiguators. To focus our efforts on translation ambiguities most likely to occur during translation, frequency-based filtering is applied. Disambiguation is not worth the effort for source words that are very infrequent, so those with a count below a certain threshold are disregarded. The exact threshold value depends on the size of the monolingal corpus, but defaults to 10k. In a similar fashion, disambiguation is unlikely to succeed for very infrequent translations, because not enough usage samples can be found in the target language corpus. Target language translations below a certain threshold (default 10k) are therefore disregarded. This filtering requires counts from the source and target corpora, both for bare lemma and lempos, from the Step 1. In addition, there are options to ignore the POS tags of multi-word expressions (MWE) or skip MWEs altogether.

Step 3: Creating a vocabulary
In order to store context samples for a translation in an efficient way, instead of storing the context lemmas themselves, their unique numerical identifiers are stored. This vectorization requires a vocabulary mapping lemmas to IDs. The basis for the vocubulary is a list of target language lemma counts from Step 2. Lemmas below a certain frequency cut-off are remove, with a default of 100. In addition, stopwords are removed and remaining garbage is removed using regular expression filters. All these steps are target language specific. This results in rather large vocabularies, but this is only for the purpose of context sampling; vocabularies are further reduced during model creation.

Step 4: Retrieving samples
For each possible translation identified in Step 2, samples of its usage are retrieved from the target language corpus. Samples consist of full sentences containing the target language lemma. The lemmas in the context samples are encoded using the vocabulary from Step 3. Very short samples (by default less than five tokens) or very long (by default more than 100 tokens) are removed. Care is also take to avoid sampling duplicate sentences. The maximum number of samples per translation candidate defaults to 10k. Encoded samples for each translation are stored as a sparse matrix in the Matrix Market format, a standard file format for exchanging matrices.[1].

[1] `http://math.nist.gov/MatrixMarket/formats.html`

7.1.2 Model construction

After sampling the resulting data is collected and processed into vector space models. Models are generated for each pair of source and target languages since each model induces the relationship between candidate translations in the target language and a particular source language lemma. The model building consists of five steps:

Step 1: Vocabulary pruning
The vocabulary of target language lemmas from the earlier sampling step is pruned in order to reduce the space and processing requirements of the model. The mapping in the vocabulary between context lemmas and matrix column indices is modified to account for the removed vocabulary items.

The vocabulary is currently reduced to correspond with the target language lemmas present in the translation dictionary. The reason for this is that lemmas not present in the translation dictionary can never appear in translations produced by the system. Other more sophisticated pruning approaches have been explored that can reduce the model size substantially, and may be incorporated in the future. The table of translation ambiguity constructed during sampling and used for model construction must also be pruned at this stage in order to exclude unneeded information.

Step 2: Determining the models required
The table of ambiguities as derived from the translation dictionary is used to construct a list of lemma and POS combinations in the source language. Each lemma in this list is followed by its associated translation candidates in the target language which are also lemma/POS combinations. The list of source lemma/POS combinations constitutes the set of models necessary for the language pair, and the content of each model is specified by its set of target language translation candidates.

Step 3: Collecting sparse matrices
The actual vectorized contexts for these target language candidates are retrieved from the samples. The context matrices are read from and processed in a sparse format for efficient processing. Internal processing is done with Compressed Sparse Row (CSR) format matrices, while the resulting model files are stored in Coordinate list (COO) format conforming to the Matrix Market specification.

Step: Summing vectors
The sets of context vectors representing a single target candidate are then subjected to transformation that intends to extract a low number of prototypes representing different semantic meanings of the target language candidate. Currently the centroids of the sample context vectors are used, creating a single prototype vector for each candidate word translation. This corresponds to the SumModel from Chapter 4.

Step 5: Constructing VSMs
The fifth and final step is to combine these prototype vectors into a single model matrix. These matrices, along with various mappings from terms to matrix indices needed during decoding are stored in a directory structure representing the complete model.

The reduction of sample contexts to prototype vector representations is currently simplified to producing the single centroid over all the sampled contexts. It is plausible that using more than one prototype representation would increase the performance of the system, and such approaches have been explored to some extent. Chapter 4 details the challenges in extracting multiple prototypes for the sentence context data.

Along with the model matrices, the model stores mappings from the model prototype contexts to the target lemma/POS tag combination, mappings from source language lemma/POS combination to the context matrices and the target language term vocabulary which is used to build the context vectors for the sentences that are passed to the Translation Equivalent Selection Module. In addition a set of meta data is stored for each model matrix containing the size and density of the matrices.

7.2 Corpus modelling module in the online system

The Corpus Modelling Module is used by the translation system through the Translation Equivalent Selection Module. The Corpus Modelling Module integrates two corpus based models that are generated off-line: Local word order and word translation selection modelled with n-gram based language modelling, and word translation selection modelled using vector space models of sentence level lexical semantics. These are combined in a probabilistic model ranking the various combinations of sentence structure and individual word translations. Viterbi decoding is used to identify the most probable translation given the output of the earlier modules by decoding the graph of possible translation candidates using the n-gram transition probabilities and weighting the translation candidate words with probabilities provided by the vector space model.

Both models are integrated with the PreseMT prototype through the addition of components embedded in the Translation Equivalent Selection Module, which is responsible for the Viterbi decoding strategy and retrieving the necessary probability scores. Separate components are used to fetch or calculate the word transition probabilities and sentence global translation candidate probabilities during the decoding process. The vector space model is implemented in the WSMScorer class which calculates similarity scores between the current sentence context and the vectors in the model corresponding to target candidates. This class constructs the context vector for the current candidate sentence, and then retrieves the models for any word with an ambiguous set of translation candidates. For each model the cosine similarities between the context vector and the prototype vectors are calculated, and the highest similarity for each distinct candidate is used as the probability weight for this candidate. For each source word the probability weights of the candidates are normalized such that the total weight of the candidates sum to one. In this process the individual model matrices are read in as needed and subsequently discarded to minimize memory usage. The component operates directly on the data structures containing the partially translated sentences, decorating them with Word

Translation Disambiguation (WTD) probabilities. They are then directly available during the Viterbi decoding. The n-gram transition probabilities are directly read from stored models and are retrieved through lookup from an in-memory representation.

7.3 Future work

As for all other modules, the contribution of the WTD module to the translation quality of the overall PRESEMT MT system for several language pairs needs to be evaluated. This is on-going work, but at the time of writing, the evaluation results are not yet available.

A major concern for the vector space models is model size, and several approaches have been explored in order to find size-reduced but still models. For reduction of context vocabulary size, we have explored several methods based on the notion of variance in data (i.e., Principal Component Analysis, PCA; Latent Direchlet Allocation, LDA; and TF-IDF) and other based on frequency (ABC filtering with lower and upper frequency thresholds for inclusion). However, until now it has been hard to judge the impact of the dimensionality reduction techniques without the context of the complete translation system. Hence the simpler methods have been included in the prototype at this stage.

The construction of a meaningful set of context prototypes through clustering of sample contexts has proven difficult due to the nature of the sentence level context data. This data is extremely sparse and sentences share very few context features. Clustering methods that are effective in clustering document collections have not been effective in clustering this data. Developing a method of extracting meaningful prototypes for this data have the potential to increase the effectiveness of the model.

Another direction open for exploration is multi-word expressions (MWE). Some of the PRESEMT translation dictionaries contain many MWE entries. These are so far ignored during translation disambiguation because they are hard to handle with the Viterbi decoder. However, the VSM-based approach to disambiguation can easily be extended to MWEs by just collecting samples and building models for target language MWEs.

Chapter 8

Future work

The framework developed for investigating the task of Word Translation Disambiguation (WTD) in the PreseMT Corpus Modelling Module lends itself to fruitful extension in a number of different ways, including to use more data or look at the test data in alternative ways, ways to either extend the vector space approach to semantic similarity modelling or to replace it with alternative approaches, and ways to move on to the task of full word translation. These possible extensions are detailed in the following sections.

8.1 Data and evaluation

A straight-forward way to extend the present work would be to use more SemEval data. The CL-WSD task offers data sets for other language pairs besides German, namely Dutch, French, Spanish, Italian. Evaluating our WTD approach on English-Italian data makes particular sense as Italian is one of the target languages in the project. This work would be carried out during the project's final year, within WP9, as Task 9.3, "Extension to other language pairs".

Furthermore, evaluation could be reconsidered. As discussed, we identified a number of problems with the Best and out-of-five evaluation measures adopted from the CL-WSD and CL-LS tasks. We attempted to address some of these, e.g., the deficiency in mode scoring, by proposing alternatives. However, these measures still remain somewhat hard to grasp intuitively due to their complicated nature, and developing more solidly grounded evaluation measures will certainly contribute to better evaluations.

Even though the CL-WSD and CL-LS have proven to be most useful for studying word translation and disambiguation approaches, they may in the end not be fully representative for the task of word translation in an actual MT environment. For example, the CL-WSD gold standard contains some rather EuroParl-specific translations and the way translation of compounds is handled is questionable. The future will hopefully bring new data sets tailored to evaluating the PreseMT MT system.

8.2 Extending vector space modelling

There are many opportunities to improve on the current vector space models. There is a wide range of alternative similarity/distance measures to cosine similarity, such as Dice coefficient, Jacquard coefficient, City block distance, and Euclidan distance. The context window currently used for co-occurrence counts defaults to a single sentence, but smaller and/or fixed sized windows may work better. Further, there are strong suggestions in the literature that raw co-occurrence counts in the matrix do not work nearly as well as more abstract measures of cohesion such as Pointwise Mutual Information or the T-test statistic. Combinations of TF*IDF with other corpus transformations have so far not been tried, nor the effect of the number of dimensions on the RP and LSI transformations. We have also started working on transformations that implement more class-directed methods of feature weighting such as Information and Gain Ratio.

Yet another direction is to explore combinations of the Vector Space Model approach discussed in Chapter 4 either with the unsupervised Self-Organising Maps of Chapter 6 and/or the more conventional n-gram approach to language modelling discussed in Chapter 5. In particular, as our VSM works with contexts much wider than n-gram models, it can model long distance relations between a translation candidate and a discriminative word in its context. In contrast, n-gram models are good at capturing local relations such as word order and collocations. A combination of both may therefore be beneficial.

8.3 Alternative semantic similarity modelling

Vector space modelling and Self-Organising Maps for determining similarity in contexts are attractive because they do not require any external knowledge resources besides a large monolingual corpus. Still, alternative approaches could be investigated, for example, Memory-Based Learning (MBL). MBL is a supervised machine learning approach which has its roots in nearest neighbour classification (Daelemans, 1999; Daelemans and van den Bosch, 2005). It is based on the idea that direct re-use of examples using analogical reasoning is better for solving NLP problems than the application of (manually) rules extracted from those samples. Memory-based learning has been repeatedly applied to Word Sense Disambiguation (e.g., Veenstra et al., 2000) — a task closely related to Word Translation Disambiguation — and is consistently among the best performing approaches to supervised word sense disambiguation (Navigli, 2009). The key difference to other machine learning approaches is that MBL is a form of lazy learning which refrains from abstraction. This makes it particularly suitable for tasks for which the amount of training data is limited. Moreover, it does not abstract away from low-frequency exceptions typically occurring in natural language.

The MBL system participating in the SemEval CL-WSD task (the UvT system) obtained the highest score for the two languages (Dutch and Spanish) it targeted (van Gompel, 2010). The core of the system consists of so-called *word experts*, one per source word, which are memory-based classifiers trained to predict the correct target word translation on the basis of a range of local and global features such as word, lemma and POS features. Although the training material in the original approach was derived from parallel corpora, essentially the

same approach can be applied to our setting by extracting training instances from a monolingual corpus in combination with a dictionary.

However, initial experiments on applying MBL within PresеMT have so far been discouraging, with the memory requirements being too large and the processing too slow to make it a realistic option.

8.4 Full word translation

So far we have restricted ourselves to disambiguation of a given set of translation candidates. Future work may extend the task to full word translation, that is, including the initial step of collecting a set of translation candidates. Trivially translation candidates can be retrieved from a dictionary, but as no dictionary has complete coverage, inevitable there will be words for which translation candidates have to be constructed in some other way. This may include morphological processing such as inflection and compounding. The vector space modelling approach also offers an interesting alternative: candidate translations can be retrieved from a generic VSM over all tokens encountered in a huge monolingual corpus. This is closely related to the idea of bootstrapping a translation lexicon using a VSM (Rapp, 1999).

Another aspect which has so far been neglected here is that the choice of a particular word translation is likely to depend on other nearby word translations. However, since each word in a sentence may have multiple likely translations, choosing the best word translations becomes a global optimization problem similar to finding the best sequence of words through a word lattice in automatic speech recognition. One interesting direction in this respect is application of Game Theory for finding an optimal solution, as mentioned in the Section on Corpus Modelling in the PresеMT "Description of Work" (Annex I to the PresеMT Grant Agreement).

Bibliography

Androutsopoulos, I. and P. Malakasiotis (2010, May). A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research 38*, 135–187.

Ballesteros, L. and W. Croft (1998). Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 64–71. ACM.

Baroni, M. and A. Kilgarriff (2006). Large linguistically-processed web corpora for multiple languages. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Posters & Demonstrations*, pp. 87–90. Association for Computational Linguistics.

Baroni, M., A. Kilgarriff, J. Pomikálek, and P. Rychlỳ (2006). WebBootCaT: instant domain-specific corpora to support human translators. In *Proceedings of EAMT 2006*, pp. 247–252. Citeseer.

Basile, P. and G. Semeraro (2010, July). UBA: Using automatic translation and wikipedia for cross-lingual lexical substitution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, pp. 242–247. Association for Computational Linguistics.

Cam, R. and L. Gilles (2002). A new model of the pareto effect (80:20 rule) at the brand level. In *Proc. ANZMAC Conf.*, Melbourne, Vic., Australia, pp. 1431–1436.

Chiao, Y., J. Sta, and P. Zweigenbaum (2004). A novel approach to improve word translations extraction from non-parallel, comparable corpora. In *Proceedings of the International Joint Conference on Natural Language Processing*, Hainan, China.

Church, K. and P. Hanks (1990). Word association norms, mutual information, and lexicography. *Computational linguistics 16*(1), 22–29.

Church, K. W. and P. Hanks (1989). Word association norms, mutual information, and lexicography. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 76–83. Association for Computational Linguistics.

Daelemans, W. (1999). Introduction to the special issue on memory-based language processing. *Journal of Experimental & Theoretical Artificial Intelligence 11*(3), 287–296.

Daelemans, W. and A. van den Bosch (2005). *Memory-based language processing*. Cambridge: Cambridge University Press.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American society for information science 41*(6), 391–407.

Dumais, S., T. Letsche, M. Littman, and T. Landauer (1997). Automatic cross-language retrieval using latent semantic indexing. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, pp. 15–21.

Federico, M., N. Bertoldi, and M. Cettolo (2010, November). *IRST Language Modeling Toolkit, Version 5.50.02: User Manual*. Trento, Italy: FBK-irst.

Federico, M. and M. Cettolo (2007). Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 88–95. Association for Computational Linguistics.

Firth, J. (1957). A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis 51*, 1–31.

Fung, P. and K. McKeown (1997). Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora*, pp. 192–202.

Fung, P. and L. Y. Yee (1998). An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th international conference on Computational linguistics*, Morristown, NJ, USA, pp. 414–420. Association for Computational Linguistics.

Gao, J., J. Nie, J. Zhang, E. Xun, Y. Su, M. Zhou, and C. Huang (2000). TREC-9 CLIR Experiments at MSRCN. In *Proceedings of the Nineth Text REtrieval Conference (TREC-9)*, pp. 343–354. NIST.

Gao, J., M. Zhou, J. Nie, H. He, and W. Chen (2002). Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 183–190. ACM.

Harris, Z. (1954). Distributional structure. *Word 10*, 146–162.

Jabbari, S., M. Hepple, and L. Guthrie (2010). Evaluation metrics for the lexical substitution task. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Stroudsburg, PA, USA, pp. 289–292. Association for Computational Linguistics.

Jang, M., S. Myaeng, and S. Park (1999). Using mutual information to resolve query translation ambiguities and query term weighting. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pp. 223–229. Association for Computational Linguistics.

Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28*(1), 11–21.

Kaski, S. (1998). Dimensionality reduction by random mapping: Fast similarity computation for clustering. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, Volume 1, pp. 413–418. IEEE.

Kilgarriff, A., P. Rychly, P. Smrz, and D. Tugwell (2004, July). The Sketch Engine. In *Proceedings of Euralex*, Lorient, France, pp. 105–116.

Kishida, K. (2005). Technical issues of cross-language information retrieval: a review. *Information Processing & Management 41*(3), 433 – 455.

Kishida, K. (2007). Term disambiguation techniques based on target document collection for cross-language information retrieval: An empirical comparison of performance between techniques. *Information Processing & Management 43*(1), 103 – 120.

Koehn, P. (2005). EuroParl: A parallel corpus for statistical machine translation. In *Proceedings of the MT Summit*, Phuket, Thailand.

Koehn, P. and K. Knight (2000). Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 711–715. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Koehn, P. and K. Knight (2001). Knowledge sources for word-level translation models. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp. 27–35.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics 43*(1), 59–69.

Kohonen, T. (1997). *Self-Organizing Maps* (2nd ed.). Springer Series in Information Sciences. Heidelberg, Germany: Springer.

Lefever, E. and V. Hoste (2009). SemEval-2010 task 3: cross-lingual word sense disambiguation. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pp. 82–87. Association for Computational Linguistics.

Lefever, E. and V. Hoste (2010a). Baselines trial data, semeval package. Technical report, University College Ghent, Faculty of Translation Studies,.

Lefever, E. and V. Hoste (2010b, July). Semeval-2010 task 3: Cross-lingual word sense disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, pp. 15–20. Association for Computational Linguistics.

Lin, C., W. Lin, G. Bian, and H. Chen (1999). Description of the NTU Japanese-English cross-lingual information retrieval system used for NTCIR workshop. In *First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pp. 145–148. Citeseer.

Maeda, A., F. Sadat, M. Yoshikawa, and S. Uemura (2000). Query term disambiguation for Web cross-language information retrieval using a search engine. In *Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, pp. 25–32. ACM.

Manning, C. and H. Schütze (1999). *Foundations of statistical natural language processing*. MIT Press.

McCarthy, D. and R. Navigli (2007). SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 48–53. Association for Computational Linguistics.

Mihalcea, R., C. Corley, and C. Strapparava (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 775–780.

Mihalcea, R., R. Sinha, and D. McCarthy (2010). Semeval-2010 Task 2: Cross-Lingual Lexical Substitution. In *Proceedings of the Fifth International Workshop on Semantic Evaluations (SemEval-2010)*, pp. 9–14.

Monz, C. and B. Dorr (2005). Iterative translation disambiguation for cross-language information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 520–527. ACM.

Navarro, G. (2001, March). A guided tour to approximate string matching. *ACM Computing Surveys 33*, 31–88.

Navigli, R. (2009). Word Sense Disambiguation: a survey. *ACM Computing Surveys 41*(2), 1–69.

Qu, Y., G. Grefenstette, and D. Evans (2003). Resolving translation ambiguity using monolingual corpora. In *Advances in Cross-Language Information Retrieval*, pp. 223–241. Springer.

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

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

Rapp, R. and M. Zock (2010). Automatic dictionary expansion using non-parallel corpora. In A. Fink, B. Lausen, W. Seidel, and A. Ultsch (Eds.), *Advances in Data Analysis, Data Handling and Business Intelligence*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 317–325. Springer Berlin Heidelberg.

Řehůřek, R. and P. Sojka (2010, May 22). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, Valletta, Malta, pp. 45–50. ELRA.

Sadat, F., A. Maeda, M. Yoshikawa, and S. Uemura (2002, s). A combined statistical query term disambiguation in cross-language information retrieval. In *Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, pp. 251 – 255.

Sahlgren, M. and J. Karlgren (2005). Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural language engineering 11*(03), 327–341.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Reading, Massachusetts: Addison-Wesley.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*, Volume 12, pp. 44–49. Manchester, UK.

Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics 24*(1), 97–123.

Sinha, R., D. McCarthy, and R. Mihalcea (2009). Semeval-2010 task 2: Cross-lingual lexical substitution. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pp. 76–81. Association for Computational Linguistics.

Tsimboukakis, N. and G. Tambouratzis (2011). Word-map systems for content-based document classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41*(5), 662 – 673.

van Gompel, M. (2010). UvT-WSD1: A cross-lingual word sense disambiguation system. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 238–241. Association for Computational Linguistics.

Veenstra, J., A. Van den Bosch, S. Buchholz, W. Daelemans, and Zavrel (2000). Memory-based word sense disambiguation. *Computers and the Humanities 34*(1), 171–177.

Xu, J. and W. Croft (1998). Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems 16*(1), 61–81.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pp. 189–196. Association for Computational Linguistics.