D3.3.1: CORPUS MODELLING MODULE (VER.1)

Grant Agreement number	ICT-248307
Project acronym	**PRESEMT**
Project title	**P**attern **RE**cognition-based **S**tatistically **E**nhanced **MT**
Funding Scheme	Small or medium-scale focused research project – STREP – CP-FP-INFSO
Deliverable title	**D3.3.1: Corpus modelling module (ver.1)**
Version	**V4**
Responsible partner	NTNU
Dissemination level	Public
Due delivery date	31.12.2010 (+60 days)
Actual delivery date	3.2.2011

Project coordinator name & title	**Dr. George Tambouratzis**
Project coordinator organisation	**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**

Contents

Figures

Tables

1. Executive summary

The PRESEMT Corpus modelling module is an offline 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.

The work on language modelling in PRESEMT proceeds along two lines: a short term and a long term approach. The first approach aims at a short term 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. This addresses 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 models are an established technology, constructing such models on the basis of text corpora containing billions of words poses interesting challenges in the area of parallelisation and high-performance computing.

The second approach targets development of new language models in order to measure semantic similarity between word translations. However, evaluation of such language models is unfeasible without a full implementation of the PRESEMT translation system and accompanying translation evaluation procedures. Instead, this deliverable focuses on one particular aspect of the translation process for which it is easier to evaluate the contribution of different language models. This is the goal of Word Translation Disambiguation, the task of 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 in SemEval-2010, which provides proper trial and test data for several languages and tasks. The work described in the present deliverable, however, only relates to the English-to-German part of the SemEval Cross-Lingual Word Sense Disambiguation task, and some initial experimental results on this task are reported.

One of the features distinguishing 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. At the heart of the matter is how to determine similarity between input context and sample context. A number of different approaches to this will be investigated during the course of the project, two of which are reported in the present deliverable. In addition to the n-gram models, the approach to similarity taken here is that of Vector Space Models, models that are 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).

List of abbreviations

CLIR	Cross-Lingual Information Retrieval
CL-LS	Cross-Lingual Lexical Substitution
CL-WSD	Cross-Lingual Word Sense Disambiguation
EM	Expectation Maximisation
HMM	Hidden Markov model
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LM	Language Model
LSI	Latent Semantic Indexing
MBL	Memory-Based Learning
MT	Machine Translation
OOF	out-of-five
OOV	out-of-vocabulary
PoS	Part of Speech
RP	Random Projection
SL	Source Language
SOM	Self-Organising Maps
TF*IDF	Term Frequency times Inverse Document Frequency
TL	Target Language
VSM	Vector Space Models
WTD	Word Translation Disambiguation

2. Introduction

The on-line PRESEMT translation system comprises three major steps. First source language text is linguistically annotated, which includes the usual steps of tokenisation, lemmatisation, PoS-tagging and lexicon lookup. 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 word translation and word ordering. It is the task of the Corpus modelling module 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. In the remainder of this text we will use the terms "*corpus model*"/"*corpus modelling module*" and "*language model*"/"*language modelling module*" interchangeably.

2.1 Motivation of approach

The challenge in translating a word 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. We can define this as a subtask in the translation process.

> **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).

It can thus 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 different, however, 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. Since in reality it is unlikely that any given bilingual dictionary has full coverage, fall-back strategies are needed. This involves, for example, morphological generation such as inflection and compounding of words. Solving word translation disambiguation would nevertheless partly solve full word translation and is therefore worthwhile to pursue.

One of the features distinguishing 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 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 PRESEMT approach 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 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 PRESEMT approach is a huge simplification which abstracts away from many important questions. At the heart of the matter is how to determine similarity between input context and sample context. A number of different approaches to this will be investigated during the course of the project, two of which are reported in the present deliverable.

2.2 The Corpus modelling module

As noted at the beginning of this introduction, the PRESEMT Corpus modelling module should deliver language models that can be used by the Translation equivalent selection module for tasks such as word translation and word ordering. This means that the design and implementation of the Corpus modelling module depends on the requirements of the Translation equivalent selection module. However, as work on both modules started at the same time, this posed some practical problems. As work on the Translation equivalent selection is in progress, it is at this time impossible to explicitly specify its requirements regarding the language models. At the same time, work on language modelling in isolation is difficult without an application context like the PRESEMT translation system to serve as a framework for evaluating models. As a solution to these problems, our work on language modelling proceeds along two lines: a short term and a long term approach.

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

The first approach aims 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) MT. They 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. Statistical n-gram models address 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 parallelisation and high performance computing. In addition, they serve as the state-of-the-art baseline on which we hope to improve in the second line of work.

The second approach targets development of new language models according to Annex I to the PRESEMT Grant Agreement, in order to measure semantic similarity between word translations. Over time three different strategies for this will be investigated. The first is to use Vector Space Models (VSM) to measure the similarity between context of source word and the contexts of translations candidates in a target language corpus. The second one is to use Memory-Based Learning (MBL) to build word experts for disambiguating individual words. The third approach is similar to the first, but uses Self-Organising Maps (SOM) to measure context similarity. Of these, only initial work with vector space models for WTD will be reported in this deliverable; the complete work using all three strategies will be reported in the later version of the deliverable (D3.3.2: Corpus modelling module, ver.2).

However, 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. Still, awaiting the availability of a fully-functioning Translation equivalent selection module would be equally unfeasible. The present version of the deliverable thus focuses on one particular aspect of the translation process for which the contribution of different language models can be evaluated more easily; and this is the goal of Word Translation Disambiguation. One of the advantages of this choice of task is that we can 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, 2010). This provides proper trial and 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 described in the present deliverable, however, only relates to the English-to-German part of the CL-WSD task, and some initial experimental results on this task are reported. Extending this to evaluating the WTD approach on English-Italian data would obviously be reasonable, as Italian is one of the target languages in the PRESEMT project, albeit one which will be targeted under the project's final year (in WP9, as Task *T9.3: Extension to other language pairs*).

2.3 Deliverable outline

The rest of the deliverable is structured as follows: the first two sections give some background and establish the framework in which the experiments will be run. Hence Section 3 starts out by discussing some related work – in particular different approaches to word translation – and Section 4 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, which provide training and test data.

One of the advantages of reusing the SemEval word translation task framework is that it includes an evaluation method. However, it is not completely the evaluation method needed in the PRESEMT project, and the evaluation criteria used in SemEval have some important shortcomings. These evaluation criteria form the topic of Section 5, which proposes some modifications to the SemEval criteria that are needed in the PRESEMT context.

The next sections go into detail on the current two 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. Section 6 is devoted to the short-term approach, and details how n-gram models have been created from corpora mined from the web; while Section 7 targets the long-term approach, discussing the use of Vector Space Models in Word Translation Disambiguation. This section reports some initial experimental results on applying this approach to the English-to-German part of the SemEval Cross-Lingual Word Sense Disambiguation task.

Finally, Section 8 gives a concluding discussion of the present version of the Corpus modelling module and points to the future directions of research that we aim to follow in producing the second version of the module, as will be reported in Deliverable D3.3.2: Corpus modelling module (ver.2) by Month 24 of the project.

3. 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 Maximisation 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**

   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 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.

Table 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	identical	11.9
Monolingual corpus	spelling + context	38.6

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 section will be restricted to the fourth approach above, that is, to translation using monolingual corpora in combination with bilingual dictionaries.

3.1 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 7.2).

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 7.2). 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 rescore 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.

3.2 Estimating word translation probabilities using Expectation Maximisation

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 Section 6), 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 Maximisation algorithm.

Monz and Dorr (2005) also employ an iterative procedure based on Expectation Maximisation 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.

3.3 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 t1 and t2, they retrieve all available translations from a dictionary. Next they generate all possible pairs of translations (*a, b*) such that *a* is a translation of *t1* and *b* is a definition of *t2*.

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 favour infrequent co-occurrences. It essentially measures the percentage of the co-occurrences of and 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 with respect to 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 generalise 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 with respect to IR between these measures. Slight variations on this approach can be found in Sadat et al. (2002).

Gao et al. (2001, 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.

4. The SemEval word translation tasks

The initial work on Word Translation Disambiguation (WTD) in PRESEMT 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 proper trial and test data, an evaluation method, baseline scores, and scores of competitive systems (relying on parallel data). This section reviews relevant parts of both these shared tasks, while the next section (Section 5) then at length discusses the evaluation criteria used in SemEval, and some of their shortcomings.

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

The Cross-Lingual Lexical Substitution (CL-LS) task[2] (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.

4.1.1 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 comprises 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).

4.1.2 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. In the best score, 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 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 (i.e., the most frequently picked translation only).

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

The details of the evaluation measures are fairly complicated and will be discussed in depth in Section 5.

4.1.3 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.

4.2 SemEval-2010 Task 3: Cross-Lingual Word Sense Disambiguation

The Cross-Lingual Word Sense Disambiguation (CL-WSD) task[3] (Lefever and Hoste, 2010, 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.

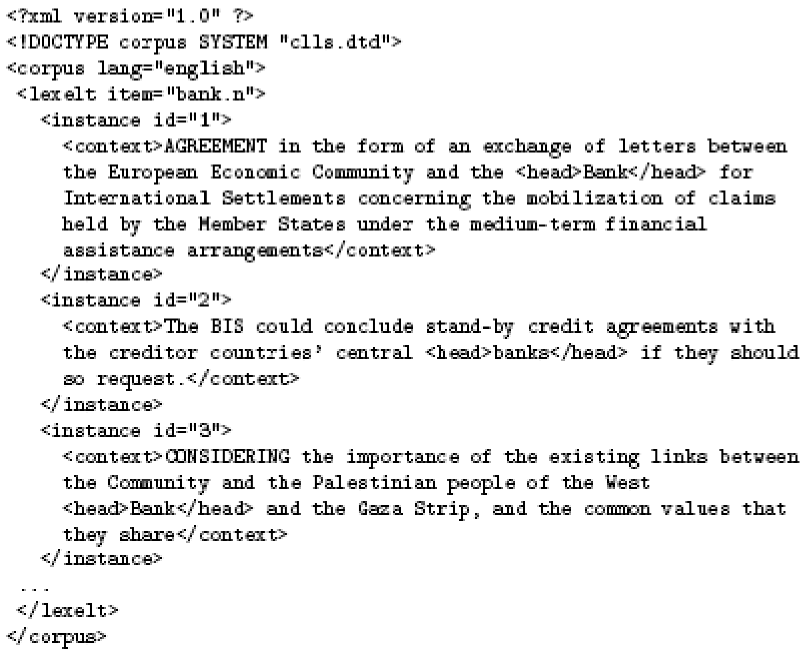
4.2.1 Data

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 lemmatised 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 lemmatised.

Second, trial and test data is extracted from two independent corpora (JRC-ACQUIS[4] and BNC[5]). 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 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.

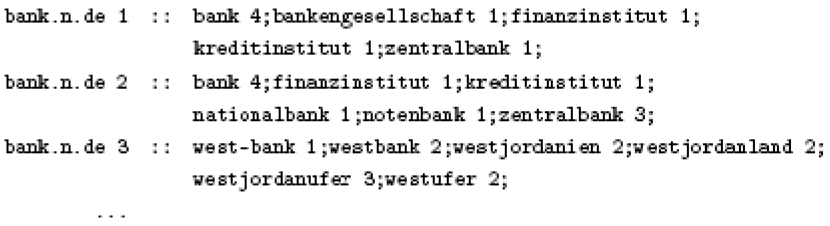
Below is a sample from the trial data in XML format, where each context element provides an English sentence which contains a surface form of the lemma ‘*bank*’.

[3] http://semeval2.fbk.eu/semeval2.php?location=tasks#T8
[4] http://langtech.jrc.it/JRC-Acquis.html
[5] http://www.natcorp.ox.ac.uk/



```
<?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 following sample 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.

4.2.2 Evaluation

Evaluation is almost identical to that in the CL-LS task, except that the out-of-ten score is replaced by an out-of-five score. Again, the evaluation criteria will be further discussed in Section 5.

4.2.3 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.

5. 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 setup includes an evaluation method. However, it is not completely the evaluation method needed in PRESEMT and – in particular – the evaluation criteria used in SemEval have some important shortcomings. Thus the decision to use the SemEval Cross-Lingual Word Sense Disambiguation (CL-WSD) material in the development of the PRESEMT Corpus modelling module made it quite necessary to do a thorough appraisal of the evaluation criteria specified for the task. This section discusses at length the SemEval evaluation criteria and proposes some modifications to the criteria needed in the PRESEMT context.

Specifically, it is crucial for the development of the Corpus modelling module that the criteria help to accurately assess the performance of the system and to identify progress – or lack thereof – in the improvement of the module. Here we discuss the evaluation criteria in two contexts relevant to the development of the module: The performance of the module on the CL-WSD task itself (which does not include candidate selection), and the performance of the module on the full generation and ranking of word translation candidates, including candidate selection based on the dictionaries available to the project.

5.1 SemEval scoring criteria

The SemEval CL-WSD task presents two different scoring criteria, ‘best’ (**Best**) and ‘out-of-five’ (**OOF**), where the ‘best’ score penalises for additional guesses, while ‘OOF’ does not penalise additional guesses but limits them to five. There is also a ‘Mode’ variant of the criteria, which measures the ratio of annotator “modes” found among the submitted target terms. 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 penalised 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 criteria are stated by Lefever and Hoste (2010), but the authors do not explicitly describe the motivation for combining their various factors in the way they do. Since we have some reservations about the criteria we would have liked to see this discussed, but instead the paper refers to similar evaluation criteria used in the Lexical Substitution (LS) task described by McCarthy and Navigli (2007). Here the Best criterion is described as emphasising the systems with the tightest agreement to annotators selections, and the OOF criterion (which is an out-of-ten score in the LS task) as allowing the systems to more freely include a variety of terms while not being explicitly penalised for it.

The Best criteria are defined by Lefever and Hoste (2010) using the following formulae.

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

(2)
$$\text{Recall} = \frac{1}{|T|} \sum_{a_i; i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|a_i| \cdot |H_i|}$$

The OOF criteria are defined as follows:

$$\text{Precision} = \frac{1}{|A|} \sum_{a_i : i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|} \tag{3}$$

$$\text{Recall} = \frac{1}{|T|} \sum_{a_i : i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|} \tag{4}$$

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.

5.2 Scoring normalised by sum of all counts

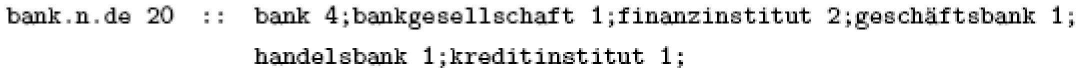
The scoring is implemented by a Perl script supplied alongside the SemEval trial data. It must be noted that the Perl script appears to be a modified version of one used for other purposes and the amount of superfluous code makes the script somewhat hard to follow. In order to have a simple and easy to understand evaluation system we have re-implemented the scoring script in our Python experimental environment. This enables us to modify the evaluation criteria when we need to and easily integrate evaluation into the PRESEMT Corpus modelling module.

We have verified that the Best and OOF scores are identical to those produced by the original script, but we have improved the ‘mode’ scores described below to also consider multiple ‘modes’, i.e., where two gold terms have the same annotator agreement count. We will refer to this method of selecting the mode as the “**Extended Mode**” (EM) of the test items, and using the EM our implementation will report different “**Mode scores**” than the CL-WSD supplied script. In general the ‘Mode’ scores will be higher and better reflect the target systems ability to find ‘Mode’ terms since test items where the ‘Mode’ happens to be shared by two gold terms are not disregarded.

Both criteria sum the normalised scores for the target terms submitted. The normalisation consists of dividing the annotator agreement count for the term by the total count (*$freq_{res}$* and $|H_i|$ in the equations). This is done separately for each context which means that the weight of the same inter-annotator agreement count will vary between test items. The two criteria then differ in that the Best score is divided by the number of submitted target terms ($|a_i|$ in the equations) while the OOF is not. The OOF criterion is described as more lenient given that it will uniformly give higher-numbered scores.

5.3 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 SemEval CL-WSD criteria will as described 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.

Table 2: 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

A similar issue is the case for the OOF score; for those the PRESEMT version of the scoring does not divide by the number of submitted terms, but 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 normalisation 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 normalised across words. One way to visualise this is to consider that the score is penalised 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.

5.4 'Perfect system' scoring

In order to illustrate the maximum score attainable with the SemEval evaluation method, a perfect system for both the Best and OOF criteria was simulated. As can be seen in Table 2, 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, minimising 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' 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.

5.5 Scoring normalised by annotator frequency count

Jabbari et al. (2010) discuss some of the problems described above in the context of the CL-LS task, which as mentioned has similar scoring criteria, and propose a set of modifications that are similar to those we have implemented in our own alternative scoring function. This consists of normalising the score for each term to the highest annotator frequency count instead of the sum of all the counts. This ensures that the target terms considered best by the annotators contribute 1 to the score, while the less suitable terms contribute correspondingly less, resulting in a score of 1.0 for a system that submits only the terms considered the best and penalising the submission of multiple candidates. This makes the weighting of the scores per term stable across contexts and heads, making it easier to compare scores and evaluate improvements to the systems' performance.

The equation is then the following:

$$\text{(5)} \qquad \mathrm{Best}_i = \frac{\sum_{a \in A_i} freq_i(a)}{maxfreq_i \cdot |A_i|}$$

where the *maxfreq$_i$* term is the highest inter-annotator agreement count for this test item. The mean over the scores for all the test items in the set is then considered the full score.

Jabbari et al. (2010) also suggest replacing the 'Mode' scoring by a Best1 score which is the same as the Best score except that the system is only allowed to submit one target term. They suggest that the out-of-ten (corresponding to the out-of-five score for the CL-WSD task) be replaced by traditional precision and recall over annotator counts.

One might also ask if the SemEval CL-WSD data is biased toward methods using parallel corpora, and possibly even specifically towards EuroParl from which the gold items are drawn. The terms available to and selected by annotators could imply this, since they cover quite a variety of terms.

For example the gold terms for Bank include terms like *bank*, *bankanstalt*, *bankinstitut*, *finanzinstitut* and *kreditinstitut* for the same context. It also includes proper names specific to the context and gold terms that are in fact translations of multiple words – not just the target head. For example, *west-bank*, *westbank*, *westjordanien*, *westjordanland*, *westjordanufer*, *westufer* are translations of the compound *West Bank* rather than of the word *Bank* in isolation.

It is not reasonable for a translator to consider most of these gold standard translations, some of them particularly marginal in usage and context specific, but for the purposes of the WTD system the generation of candidates may be left out and the disambiguation system alone may be scored. Still, some of these candidates may be extremely rare in a specific corpus and thus unfairly penalise a system.

6. Statistical N-gram Language Modelling

As a first instalment of the Corpus modelling module, n-gram models have been created from the corpora mined from the web for use in the PRESEMT project. The n-gram models are built with the standard tools **IRSTLM** (Federico and Cettolo, 2007) and **SRILM** (Stolcke, 2002). With large amounts of data this poses challenges in terms of speed and storage, and lends it self well to data parallelisation. 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.

As discussed in Section 2, the Corpus modelling module is aiding the Translation equivalent selection module in the selection of the most likely translation candidate, hence 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.

6.1 Methodology

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[6] 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 parallelisation 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).

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

6.2 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 parallelising 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.

6.3 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).

The current LMs are not so interesting in their own right as questions regarding tokenisation have been deferred. The production versions will inevitably be different from these development versions.

The German data was also filtered through the tokeniser from the TreeTagger (Schmid, 1994), as well as cut at length 50. (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). Furthermore, models that take hours to load into memory are impractical for development purposes so smaller sample LMs were built for this purpose. Having infrastructure in place, an LM for a 3Bn word corpus can be built and rebuilt in half a day.

N-gram models of various sizes and nature (built over words, lemmas or PoS) are unavoidable baselines when building novel models of language.

6.4 Evaluation

As the performance of the language models will have to be measured relative to the purpose for which they have been created, they are not currently evaluated. However, some statistics on held-out portions of the corpus can be assembled. 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 capitalisation.

In addition to this, some statistics on the dictionary creation can be retrieved based only on the input corpus, as shown in Table 3, 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.

Table 3: 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%

7. Word Translation Disambiguation

As discussed in the Introduction, Word Translation Disambiguation (WTD) is the task of 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). Since the PRESEMT MT system is currently not sufficiently developed to serve as a test platform for WTD experiments, we intend to the reuse the framework from the two word translation tasks from SemEval-2010, namely, the Cross-Lingual Lexical Substitution task and the Cross-Lingual Word Sense Disambiguation task, as described in Section 4. Work reported in this section, however, only concerns the English-to-German part of the Cross-Lingual Word Sense Disambiguation task.

7.1 General data collection and pre-processing

In order to reuse the data from the SemEval tasks described above, some modifications are necessary. Moreover, neither task provides training data, so this needs to be collected in some other way. Finally both training and test data have to be linguistically pre-processed. In this section, we focus on describing data collection and pre-processing for the German part of the CL-WSD data set in order to obtain training and test data for the WTD work in PRESEMT.

7.1.1 Construction of training data

The construction of training data involves three steps: extracting translation candidates, retrieving translation samples, and tagging and lemmatising the samples. 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 in this work we abstract away from subtask 1 by assuming that a perfect solution to finding translation candidates already exists. 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:

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

Step 2: Retrieve translation samples

For each of the translation candidates, we need examples of its use in context. These context samples are extracted 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). We use the Sketch Engine (Kilgarriff et al., 2004) web service API to search for occurrences of a particular translation in the DeWac corpus and to retrieve German sentences containing the word. Some examples for *Bank* (financial institute) are given below:

> *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.*

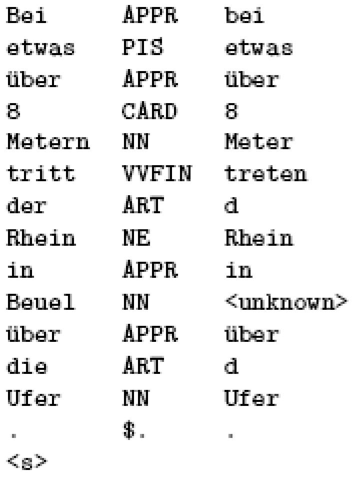
Separate sets of sentence contexts were collected, both based on the occurrence of the word form and on the matching lemma. Most of the 243 different gold terms (in total, for the five head words shown in Table 2, i.e., *Bank*, *Movement*, *Occupation*, *Passage*, and *Plant*) are found in the corpus – around 15 have a frequency of 0, but otherwise the frequencies naturally vary substantially. Frequency bins are shown in Table 4. Some of the terms have a large frequency in the corpus, often more then 500,000 occurrences. We sampled 5000 random sentence contexts for terms with a frequency higher than 5000 to avoid collecting an excessive amount of data.

Table 4: Frequency bins for the CL-WSD German gold terms collected from DeWaC

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

Step 3: Tag and lemmatise samples

Sample sentences are tokenised, PoS-tagged and lemmatised 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>
```

7.1.2 Construction of test data

Construction of test data takes the following steps:

Step 1: Tag and lemmatise

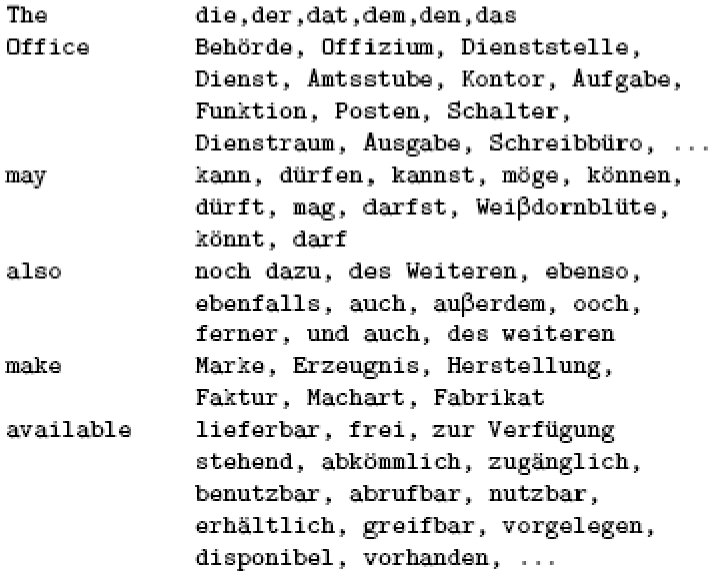
English sentences from the CL-WSD trial or test data are tokenised, PoS-tagged and lemmatised using the TreeTagger for English (Schmid, 1994).

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 bridge 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 offline 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 used in WTD 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. 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.

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, Weißdornblüte,
             könnt, darf
also         noch dazu, des Weiteren, ebenso,
             ebenfalls, auch, außerdem, 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, ...
```

7.2 Disambiguation with a Vector Space Model

Recall that the core idea of the WTD approach in PRESEMT is to search among sample contexts of the translation candidates for those which are most similar to the context of a source word. A major question is therefore how to measure similarity of textual contexts. 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) for words, also 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[7]. 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 is 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, e.g., 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).

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 multidimensional vector space. Next this vector space is 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 generalisation. 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. A more detailed description of these steps follows below. For implementation we use Gensim (Řehůřek and Sojka, 2010), a framework for Vector Space Modelling in Python[8].

7.2.1 *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 may be considered as the features which model the context of 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.

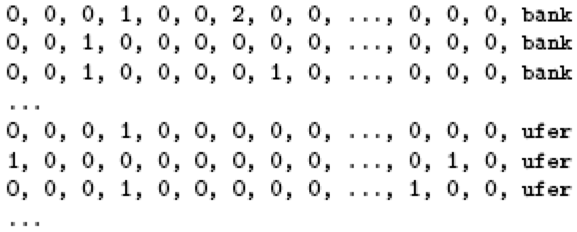
[7] http://wordspace.collocations.de/doku.php/course:start

[8] http://nlp.fi.muni.cz/projekty/gensim

To begin with, we work with the lowercased lemmas as provided by the tagger, only backing off to lowercased token when the tagger fails to provide a lemma. The vocabulary thus initially consists of all lemma types occurring in 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.

7.2.2 Prediction

The translation candidates are predicted from the target words associated with the vectors in the vector space. 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 to Gensim toolkit (Řehůřek and Sojka, 2010) also for this step. It supports several types of transformations:

- The TF*IDF ("*term frequency times inverse document frequency*") transformation (Spärck 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. But this is completely unrelated to the class label, so it may actually reduce the weight of discriminative words!

- Random Projection (RP) transformation—also known as Random Indexing—is a way to reduce the dimensionality of the vector space by statistical approximation (Sahlgren and Karlgren, 2005). It is claimed to result in much smaller matrices and quicker retrieval without significant loss in performance.
- Other transformations supported by Gensim are Latent Semantic Indexing, LSI (Dumais et al., 1997) and Latent Dirichlet Allocation, LDA (Blei et al., 2003).

Step 2: Transform corpora

One or more corpus transformations are used to transform both training and test corpora.

Step 3: Index training corpus

The training corpus is indexed to facilitate fast search for similar vectors.

Step 4: Translation prediction

For each vector in the test corpus, we search the training corpus for the most similar vector(s). So far simple cosine similarity has been used for this purpose. The predicted translation is the class label (i.e., German word) associated with the most similar vector(s). Alternatively, we take the majority class over the *N* most similar vectors.

Step 5: Scoring

The CL-WSD scoring script is applied which compares predictions against the gold standard files and outputs a number of scores. Alternatively, our own modified evaluation measures are applied.

7.3 Experimental results

This section reports experimental results on the CL-WSD trial data for German; given the preliminary nature of the experiments, we have so far refrained from evaluation on the test data for methodological reasons. We first report a number of baselines scores, followed by results for a number of different vector space model settings.

7.3.1 Baselines and upper bounds

A reasonable baseline for this task has been constructed by using word and lemma frequency statistics from the DeWaC corpus, which consists of over 1.6 billion tokens and is described by Baroni et al. (2006). We consider this the simplest possible reasonable approach to the WTD task beyond drawing an arbitrary candidate from the lexicon, utilising no information beyond the target word itself. The baseline system uses the frequency of each candidate in DeWaC. The candidates were collected from the CL-WSD gold and the frequency of the tokens or tokens with the same lemma were extracted using the Sketch Engine query API (Kilgarriff et al., 2004). The frequency counts for both word and lemma forms were collected, with the intention of using the best scoring of the two as the baseline.

As can be seen in Tables 5 and 6, the scores that result from this baseline vary by a fair amount, which is to be expected, since as discussed in Section 5.3 there is considerable variation between the contexts in the SemEval trial data, and the SemEval scores are very biased toward selecting exactly the translations that carry a disproportionately large share of the inter-annotator agreement counts. The baseline scores are in general fairly low since they select the same words regardless of context and as such can only supply relevant terms for a fraction of the test items. It is worth noting that the word frequency scores are for the most part slightly better than their lemma counterparts except for the head word 'Bank' where both scores are quite a bit higher than for all the other heads. In addition, the baseline never selects the "Mode" or "Extended mode" (Section 5.1) among the top ten except for the word '*Bank*', and this might make the baseline too pessimistic. The 'Mode' scores are as result all near zero and of little use as baselines.

Table 5: Baselines obtained by using target side **word** frequencies alongside the published baselines. In addition to the OOF score for the top-5 frequency candidates, the table shows the Best scores for the single top frequency candidate and for the top-10 frequency candidates.

	Top-1	**Top-10**		**Published**	
Head word	**Best**	**Best**	**OOF**	**Best**	**OOF**
Bank	0.50	0.06	8.91	2.49	23.23
Movement	0.42	0.05	3.37	3.91	20.34
Occupation	3.83	0.43	7.28	13.45	32.78
Passage	1.50	0.17	9.11	4.58	27.35
Plant	0.42	0.05	2.54	11.70	21.06

Table 6: Baselines obtained by using target side **lemma** frequencies alongside the published baselines. In addition to the OOF score for the top-5 frequency candidates, the table shows the Best scores for the single top frequency candidate and for the top-10 frequency candidates.

	Top-1	**Top-10**		**Published**	
Head word	**Best**	**Best**	**OOF**	**Best**	**OOF**
Bank	0.50	0.06	16.57	2.49	23.23
Movement	0.87	0.10	2.99	3.91	20.34
Occupation	3.83	0.43	7.69	13.45	32.78
Passage	1.50	0.17	8.66	4.58	27.35
Plant	0.42	0.05	2.54	11.70	21.06

While the baseline sets a lower bound on performance, the difference in focus between the SemEval task and the PRESEMT WTD task will require that we consider evaluation scores in the context of the differences in baseline score for each head word and the methodology under evaluation in general, for example if we are doing candidate selection or not. It can not be expected that the baseline scores and SemEval evaluation measurements would consistently and monotonously make judgments of the WTD system quality by themselves. In the light of this we will prefer to use the alternative normalisation of the scores as discussed in Section 5.5, which alleviates these problems to some extent.

With the exception of the OOF score for the head work '*Bank*' our baselines are considerable lower than the published ones using frequencies based on parallel alignments. This probably reflects the strengths of using parallel corpora for this task in addition to the difficulty of this task when no parallel resources are available. 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 which is discussed by Koehn and Knight (2001) (cf. Section 3).

7.3.2 Vector Space Model

The CL-WSD trial data concerns 5 German words and contains 20 instances of each word in the context of a sentence. Training and test corpora were constructed according to the procedure described in Sections 7.1.1 and 7.1.2. Table 7 shows the training corpus size, listing per word the number of samples (corresponding to the number of rows in the matrix), the number of features (corresponding to the number of columns in the matrix, or alternatively the size of the vocabulary), and the density (percentage of filled cells in the matrix).

Table 7: Training corpus size per word

Word	Samples	Features	Density (%)
Bank	83,507	11,525	0.155
Movement	161,936	19,176	0.097
Occupation	81,202	11,804	0.166
Passage	128,150	15,550	0.122
Plant	103,863	13,893	0.130

The general vector space model offers a wide range of modelling options. In the initial experiments carried out so far, the following models and settings have been explored:

1. **VSM:** bare vector space model without any transformations
2. **VSM + TF*IDF:** vector space model transformed by TF*IDF weighting
3. **VSM + RP:** vector space model transformed by Random Projection down to 300 dimensions
4. **VSM + LSI:** vector space model transformed by Latent Semantic Indexing down to 200 dimensions

Table 8 presents results in terms of the best measure, showing scores for baselines, VSM variants and maximum score obtainable with perfect results. Clearly all scores are far above the first baseline of picking the most frequent translation in the TL corpus. However, the baseline based on access to a word-aligned parallel corpus poses more of a challenge. Only the scores on 'Bank' can be seen to outperform this second baseline. In two other cases, i.e., for '*Movement*' and for '*Passage*', the gap between the baseline and the best performing VSM is relatively small. Furthermore, there is no VSM variant that consistently gives the highest scores, although transformed models appear to be better than the bare VSM model.

Table 9 presents results in terms of the out-of-five measure. Again VSM scores surpass the first baseline, but do not consistently outperform the more challenging second baseline.

There appears to be no strong correlation between the best performing VSM variants in Table 8 and Table 9, nor in the scores per word. For example, the best score of VSM variants on 'Plant' is far below the second baselines, whereas the out-of-five score is consistently above it.

Table 8: Best scores per word for different baselines and vector space models

	Bank	Movement	Occupation	Passage	Plant
Most frequent translation	0.50	0.87	3.83	1.50	0.42
Most frequently aligned	2.49	3.91	**13.45**	**4.58**	**11.70**
VSM	8.57	3.37	6.92	3.75	5.42
VSM+TF*IDF	6.71	1.36	10.92	4.17	5.42
VSM+RP	**13.51**	3.52	5.14	4.35	6.70
VSM+LSI	11.86	**3.94**	4.72	2.92	6.74
Perfect	42.71	28.78	30.40	37.29	29.58

Table 9: Out-of-five scores per word for different baselines and vector space models

	Bank	Movement	Occupation	Passage	Plant
Most frequent translation	16.57	2.99	7.69	8.66	2.54
Most frequently aligned	23.23	**20.34**	32.78	**27.35**	21.06
VSM	40.70	15.90	34.75	15.77	**32.39**
VSM+TF*IDF	43.68	18.40	26.30	23.85	24.28
VSM+RP	36.34	20.06	**44.07**	20.45	26.44
VSM+LSI	**51.49**	14.54	34.25	22.81	26.78
Perfect	95.06	82.62	93.58	89.57	81.97

To sum up, the results on WTD with vector space modelling are encouraging. The first baseline of always choosing the most frequent translation candidate is easily surpassed. Even the second baseline derived from parallel text corpora is often exceeded, although not consistently. The rather large range of VSM scores suggests that the approach holds potential, but that the factors determining performance are not well understood at this preliminary stage.

7.3.3 Dictionary coverage

The project has three sets of English-German dictionaries available: the freely available CC dictionary[9], the Chemnitz dictionary[10], and the GFAI dictionary.

[9] dict-cc is an internet based German-English and English-German dictionary based on user generated word definitions. It is available at http://www.dict.cc/.

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

We have performed a study of how these dictionaries cover the SemEval target word clusters, as shown in Tables 10, 11 and 12. 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 head '*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 10: GFAI dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts

Head	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 11: CC dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts

Head	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

Table 12: Chemnitz dictionary coverage of the CL-WSD candidates in terms of number of words and annotator agreement counts

Head	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

8. Discussion and future work

This final section of the deliverable points to some future directions of research to follow in producing the second version of the PRESEMT Corpus modelling module, in addition to the obvious task, namely that the so far developed approaches to Word Translation Disambiguation (WTD), and ultimately to word translation in general, will be integrated in the PRESEMT MT system. Here the aim is for an optimal model in terms of trade-off between best performance and practical constraints on processing time and memory usage. Efficient implementations in Java will have to created and interfaced with the main PRESEMT MT software architecture. Most notably, the precise division of labour between the Corpus modelling module and the Translation equivalent selection module deserves some future attention.

In addition, the framework developed for investigating the task of WTD 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 subsections.

8.1 Data and evaluation

A straightforward 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.

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 Euclidean 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 more conventional n-gram approach to language modelling discussed in Section 6 with the Vector Space Model approach discussed in Section 7. 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 for determining similarity in contexts is attractive because it does not require any external knowledge resources besides a large monolingual corpus. Still, it is intended to investigate at least two alternative approaches to this. Firstly, Memory-Based Learning (MBL) is a form of supervised learning which has been consistently among the best performing approaches to supervised WSD, a task closely related to WTD. Secondly, Kohonen's (1995) Self-Organising Map (SOM) is another unsupervised method for clustering similar vectors.

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.

Self-Organising Maps (SOM)

The SOM model has been applied to textual data (Kohonen et al., 2000), and has thereafter been further developed to operate on very large text collections, via the WEBSOM method for document mining (Honkela et al., 1997; Lagus et al., 2004).

The incorporation of the SOM approach in PRESEMT will be based on recent work by consortium members on unsupervised organisation of document collections based on the documents' content (Tsimboukakis and Tambouratzis, 2011). Two approaches to this have been tried, both based on a two-stage process comprising

1. a word map created via unsupervised learning which functions as document representation, and
2. a supervised Multi Layer Perceptron-based (MLP) classifier.

In the first stage, words are grouped depending on the concept they express, using an unsupervised learning algorithm. This creates a word map, groups of related words, taking into account contextual features. In the second stage, a supervised classifier (an MLP neural network) assigns documents to categories, making use of the word map-generated groups as features.

This method is directly relevant to the PRESEMT Corpus modelling module as the content at a sentence level is directly exploitable for Word Translation Disambiguation. Here it is the first stage above which is of relevance, while the second stage will probably be modified to suit the specific task at hand. The first stage creates a word map that groups together semantically-related words into the same state or states at a small distance from each other. This word map represents documents by more compact feature vectors, achieving a dimensionality reduction compared to other possible feature sets. Two distinct variants have been evaluated for the first stage, the first one based on SOM and the second on Hidden Markov Models (HMM).

In the case of the SOM model, a systematic method has been developed for selecting the features to be retained. This technique is based on Pareto's principle (known as the 80-20 rule), according to which 20% of the features are responsible for 80% of the result (Rungie et al., 2002). Using Pareto's principle, a portion of the causes is characterised as *A*, which indicates very frequent events, with *B* and *C* corresponding to less frequent and to unimportant events, respectively (Tsimboukakis and Tambouratzis, 2011). Within the processing of the text data, only a subset of the words (assigned to classes *A*, *B* and *C* based on their relative frequencies) is used to represent the pattern space, resulting in a reduction of the data vectors used. At the same time, since these characteristics are systematically derived, they are expected to be more accurate than randomly-chosen features.

Extensive experimentation has been performed on several datasets with up to hundreds of thousands of textual documents in Greek and English, using both the SOM-based and the HMM-based variants. Both variants have been shown to possess similar classification accuracies, while each has different advantages, in terms of the ability to map relationships between words, the ability to map multiple meanings, and the capacity for processing large collections of texts. Tsimboukakis and Tambouratzis (2011) have illustrated the effectiveness of the proposed approach in comparison to state-of-the-art methods.

This approach appears promising for the Corpus Modelling and will be investigated in detail during the second year of the project. One issue to address is how to design the second stage to tailor it to the task at hand. Another important issue concerns speeding up the processing to handle very large textual datasets (even though this is an off-line process, it needs to be performed in an efficient manner). Additional aspects include investigation of the optimal limits of *ABC* analysis and study of new variants on the basis of the first experiments performed.

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, namely 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 optimisation 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 Annex I to the PRESEMT Grant Agreement.

9. References

Androutsopoulos, I. and P. Malakasiotis (2010). 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., A. Kilgarriff, J. Pomikálek, and P. Rychlý (2006). WebBootCaT: instant domain-specific corpora to support human translators. In Proceedings of EAMT 2006, pp. 247–252.

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.

Blei, D., A. Ng, and M. Jordan (2003). Latent Dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022.

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. 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.

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. 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.

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

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 nonparallel 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., E. Xun, M. Zhou, C. Huang, J. Nie, J. Zhang, and Y. Su (2001). TREC-9 CLIR experiments at MSRCN. In The Ninth Text Retrieval Conference (TREC 9), pp. 343–354.

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. Reprinted in Z. Harris, Papers in Structural and Transformational Linguistics, Reidel, Dordrecht, Holland 1970.

Honkela, T., S. Kaski, K. Lagus, and T. Kohonen (1997, June). WEBSOM – Self-Organizing Maps of document collections. In Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, pp. 310–315.

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.

Kilgarriff, A., P. Rychlý, P. Smrž, and D. Tugwell (2004). 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. (1995). Self-Organizing Maps. Berlin, Germany: Springer.

Kohonen, T., S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela (2000, May). Self organization of a massive document collection. IEEE Transactions on Neural Networks 11(3), 574–585.

Lagus, K., S. Kaski, and T. Kohonen (2004, June). Mining massive document collections by the WEBSOM method. Information Sciences, 163, 135–156.

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 (2010). 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.

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 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 AAAI'06.

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 Advances in Data Analysis, Data Handling and Business Intelligence, pp. 317–325. Springer.

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.

Rungie, C., G. Laurent, and C. Habel (2002). A new model of the Pareto effect (80:20 rule) at the brand level. In Proceedings of ANZMAC 2002, Melbourne, Australia, pp. 1431–1436.

Sadat, F., A. Maeda, M. Yoshikawa, and S. Uemura (2002). 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, Vol. 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.

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

Stolcke, A. (2002). SRILM - an extensible language modelling toolkit. In Seventh International Conference on Spoken Language Processing, 3, pp. 901–904.

Tsimboukakis, N. and G. Tambouratzis (2011). Word map systems for content-based document classification. Systems Man and Cybernetics. In print.

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 co-occurrence 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.

10. Appendix I: Work flow charts

The following pages show the work flow of the development framework for the Word Translation Disambiguation task in the PRESEMT Corpus modelling module, detailing three parts of the process:

1. **Construction of training corpus**

   As described in Section 6.1.1 this process involves three steps: extracting translation candidates, retrieving translation samples, and tagging and lemmatising the samples.

2. **Construction of test corpus**

   As described in Section 7.1.2 the construction of test data takes the following steps: tokenisation, PoS-tagging, lemmatisation, and word-for-word translation of context.

3. **Prediction**

   As described in Section 7.2.2, prediction of translation candidates from the target words associated with the vectors in a vector space model takes the following steps: construct corpus transformation (e.g., Random Projection), apply corpus transformations to both training and test corpora, index training corpus to facilitate fast search for similar vectors, translation prediction, and scoring.

Figure 1: Construction of training corpus

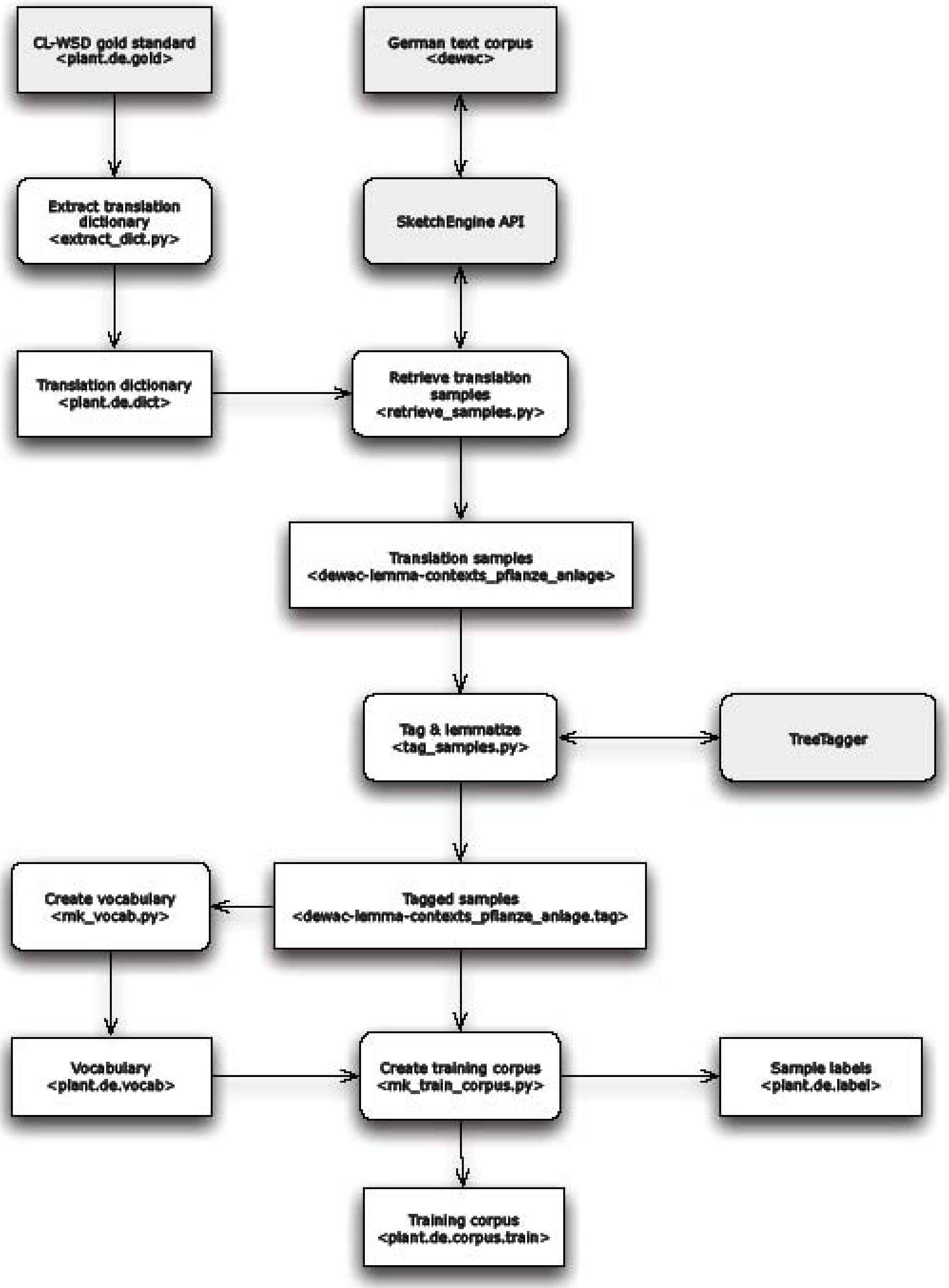


Figure 2: Construction of test corpus

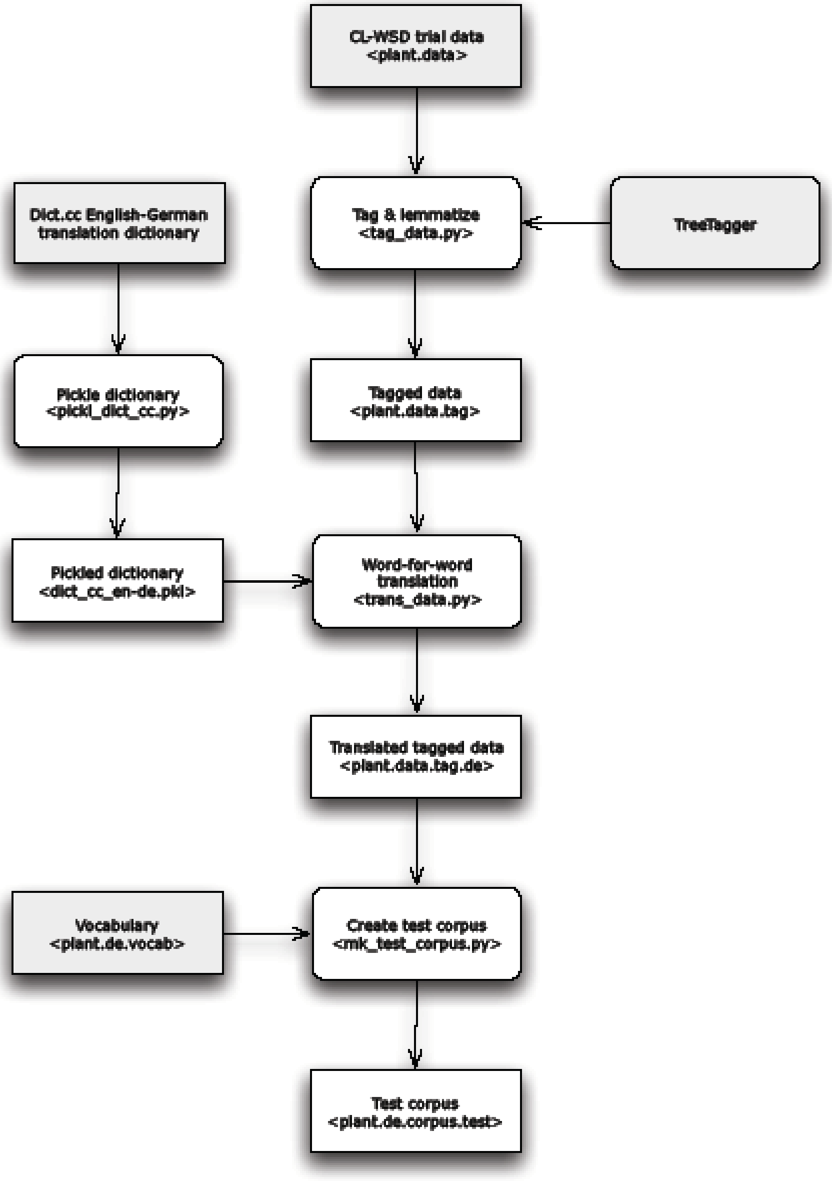
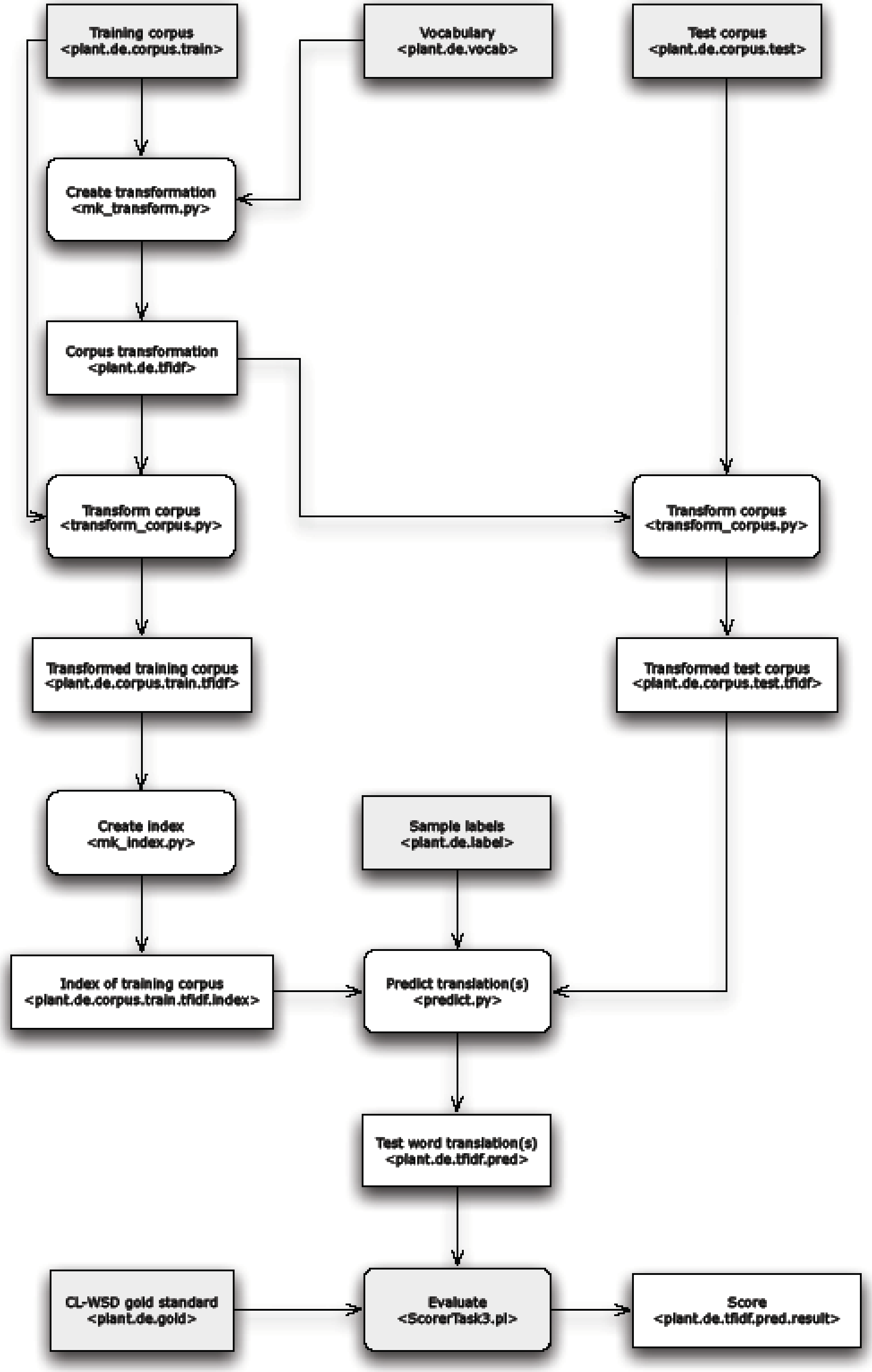


Figure 3: Prediction with vector space model


Training corpus
<plant.de.corpus.train>
Vocabulary
<plant.de.vocab>
Test corpus
<plant.de.corpus.test>
Create transformation
<mk_transform.py>
Corpus transformation
<plant.de.tfidf>
Transform corpus
<transform_corpus.py>
Transform corpus
<transform_corpus.py>
Transformed training corpus
<plant.de.corpus.train.tfidf>
Transformed test corpus
<plant.de.corpus.test.tfidf>
Create index
<mk_index.py>
Sample labels
<plant.de.label>
Index of training corpus
<plant.de.corpus.train.tfidf.index>
Predict translation(s)
<predict.py>
Test word translation(s)
<plant.de.tfidf.pred>
CL-WSD gold standard
<plant.de.gold>
Evaluate
<ScorerTask3.pl>
Score
<plant.de.tfidf.pred.result>