A Linguistically Grounded Graph Model for Bilingual Lexicon Extraction

Florian Laws, Lukas Michelbacher, Beate Dorow, Christian Scheible,
Ulrich Heid, Hinrich Schütze
Institute for Natural Language Processing
Universität Stuttgart
{lawsfn,michells,dorowbe}@ims.uni-stuttgart.de

Abstract

We present a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora. The graphs we use represent linguistic relations between words such as adjectival modification. We experiment with a number of ways of combining different linguistic relations and present a novel method, multi-edge extraction (MEE), that is both modular and scalable. We evaluate MEE on adjectives, verbs and nouns and show that it is superior to cooccurrence-based extraction (which does not use linguistic analysis). Finally, we publish a reproducible baseline to establish an evaluation benchmark for bilingual lexicon extraction.

1 Introduction

Machine-readable translation dictionaries are an important resource for bilingual tasks like machine translation and cross-language information retrieval. A common approach to obtaining bilingual translation dictionaries is *bilingual lexicon extraction* from corpora. Most work has used *parallel text* for this task. However, parallel corpora are only available for few language pairs and for a small selection of domains (e.g., politics). For other language pairs and domains, monolingual comparable corpora and monolingual language processing tools may be more easily available. This has prompted researchers to investigate bilingual lexicon extraction based on monolingual corpora (see Section 2) .

In this paper, we present a new graph-theoretic method for bilingual lexicon extraction. Two monolingual graphs are constructed based on syntactic analysis, with words as nodes and relations (such as adjectival modification) as edges. Each relation acts as a similarity source for the node types involved. All available similarity sources interact to produce one final similarity value for each pair of nodes. Using a seed lexicon, nodes from the two graphs can be compared to find a translation.

Our main contributions in this paper are: (i) we present a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora; (ii) we show that with this graph-theoretic framework, information obtained by linguistic analysis is superior to cooccurrence data obtained without linguistic analysis; (iii) we experiment with a number of ways of combining different linguistic relations in extraction and present a novel method, multi-edge extraction, which is both modular and scalable; (iv) progress in bilingual lexicon extraction has been hampered by the lack of a common benchmark; we therefore publish a benchmark and the performance of MEE as a baseline for future research.

The paper discusses related work in Section 2. We then describe our translation model (Section 3) and multi-edge extraction (Section 4). The benchmark we publish as part of this paper is described in Section 5. Section 6 presents our experimental results and Section 7 analyzes and discusses them. Section 8 summarizes.

2 Related Work

Rapp (1999) uses word cooccurrence in a vector space model for bilingual lexicon extraction. Details are given in Section 5.

Fung and Yee (1998) also use a vector space approach, but use TF/IDF values in the vector components and experiment with different vector similarity measures for ranking the translation candidates. Koehn and Knight (2002) combine

Coling 2010: Poster Volume, pages 614–622,
Beijing, August 2010

a vector-space approach with other clues such as orthographic similarity and frequency. They report an accuracy of .39 on the 1000 most frequent English-German noun translation pairs.

Garera et al. (2009) use a vector space model with dependency links as dimensions instead of cooccurring words. They report outperforming a cooccurrence vector model by 16 percentage points accuracy on English-Spanish.

Haghighi et al. (2008) use a probabilistic model over word feature vectors containing cooccurrence and orthographic features. They then use canonical correlation analysis to find matchings between words in a common latent space. They evaluate on multiple languages and report high precision even without a seed lexicon.

Most previous work has used vector spaces and (except for Garera et al. (2009)) cooccurrence data. Our approach uses linguistic relations like subcategorization, modification and coordination in a graph-based model. Further, we evaluate our approach on different parts of speech, whereas some previous work only evaluates on nouns.

3 Translation Model

Our model has two components: (i) a graph representing words and the relationships between them and (ii) a measure of similarity between words based on these relationships. Translation is regarded as cross-lingual word similarity. We rank words according to their similarity and choose the top word as the translation.

We employ undirected graphs with typed nodes and edges. Node types represent parts of speech (POS); edge types represent different kinds of relations. We use a modified version of SimRank (Jeh and Widom, 2002) as a similarity measure for our experiments (see Section 4 for details).

SimRank is based on the idea that two nodes are similar if their neighbors are similar. We apply this notion of similarity across two graphs. We think of two words as translations if they appear in the same relations with other words that are translations of each other. Figure 1 illustrates this idea with verbs and nouns in the direct object relation. Double lines indicate *seed* translations, i.e., known translations from a dictionary (see Section 5). The nodes *buy* and *kaufen* have the same objects in the two languages; one of these (*magazine – Zeitschrift*) is a seed translation. This relationship contributes to the similarity of *buy – kaufen*. Furthermore, *book* and *Buch* are similar (because of *read – lesen*) and this similarity will be added to *buy – kaufen* in a later iteration. By repeatedly applying the algorithm, the initial similarity introduced by seeds spreads to all nodes.

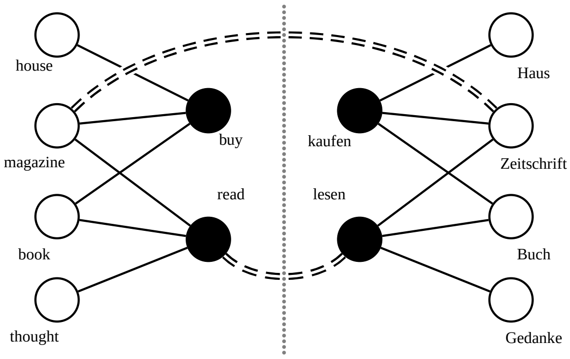

Figure 1: Similarity through seed translations

To incorporate more detailed linguistic information, we introduce typed edges in addition to typed nodes. Each edge type represents a linguistic relation such as verb subcategorization or adjectival modification. By designing a model that combines multiple edge types, we can compute the similarity between two words based on *multiple sources* of similarity. We superimpose different sets of edges on a fixed set of nodes; a node is not necessarily part of every relation.

The graph model can accommodate any kind of nodes and relations. In this paper we use nodes to represent content words (i.e., non-function words): adjectives (a), nouns (n) and verbs (v). We extracted three types of syntactic relations from a corpus: see Table 1.

Nouns participate in two bipartite relations (amod, dobj) and one unipartite relation (ncrd). This means that the computation of noun similarities will benefit from three different sources.

Figure 2 depicts a sample graph with all node and edge types. For the sake of simplicity, a monolingual example is shown. There are four nouns in the sample graph all of which are (i) modified by the adjectives *interesting* and *political* and (ii) direct objects of the verbs *like* and

relation	entities	description	example
used in this paper			
amod	a, n	adjectival modification	a *fast car*
dobj	v, n	object subcategorization	*drive* a *car*
ncrd	n, n	noun coordination	*car*s and *bus*ses
other possible relations			
vsub	v, n	subject subcategorization	a *man sleep*s
poss	n, n	possessive	the *child's toy*
acrd	a, a	adjective coordination	*red* or *blue* car

Table 1: Relations used in this paper (top) and possible extensions (bottom).

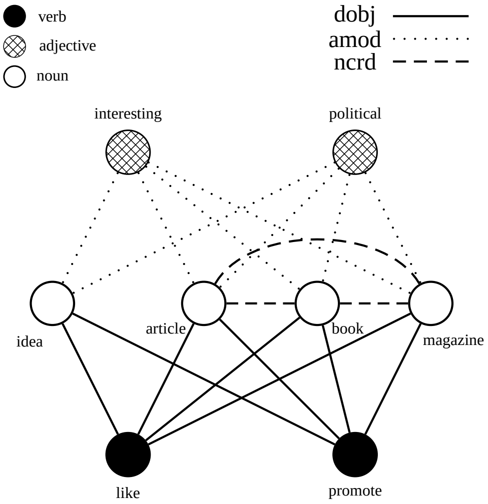

Figure 2: Graph snippet with typed edges

promote. Based on amod and dobj, the four nouns are equally similar to each other. However, the greater similarity of *article*, *book*, and *magazine* to each other can be deduced from the fact that these three nouns also occur in the relation ncrd. We exploit this information in the MEE method.

Data and Preprocessing. Our corpus in this paper is the Wikipedia. We parse all German and English articles with BitPar (Schmid, 2004) to extract verb-argument relations. We extract adjective-noun modification and noun coordinations with part-of-speech patterns based on a version of the corpus tagged with TreeTagger (Schmid, 1994). We use lemmas instead of surface forms. Because we perform the SimRank matrix multiplications in memory, we need to filter out rare words and relations; otherwise, running SimRank to convergence would not be feasible. For adjective-noun pairs, we apply a filter on pair frequency (≥ 3). We process noun pairs by applying a frequency threshold on words (≥ 100) and pairs (≥ 3). Verb-object pairs (the smallest data set) were not frequency-filtered. Based on the resulting frequency counts, we calculate association scores for all relationships using the log-likelihood measure (Dunning, 1993). For noun pairs, we discard all pairs with an association score < 3.84 (significance at $\alpha = .05$). For all three relations, we discard pairs whose observed frequency was smaller than their expected frequency (Evert, 2004, p. 76). As a last step, we further reduce noise by removing nodes of degree 1. Key statistics for the resulting graphs are given in Table 2.

We have found that accuracy of extraction is poor if unweighted edges are used. Using the log-likelihood score directly as edge weight gives too much weight to "semantically weak" high-frequency words like *put* and *take*. We therefore use the logarithms of the log-likelihood score as edge weights in all SimRank computations reported in this paper.

nodes		**n**	**a**	**v**
	de	34,545	10,067	2,828
	en	22,257	12,878	4,866
edges		**ncrd**	**amod**	**dobj**
	de	65,299	417,151	143,906
	en	288,889	686,073	510,351

Table 2: Node and edge statistics

4 SimRank

Our work is based on the SimRank graph similarity algorithm (Jeh and Widom, 2002). In (Dorow et al., 2009), we proposed a formulation of SimRank in terms of matrix operations, which can be applied to (i) weighted graphs and (ii) bilingual problems. We now briefly review SimRank and its bilingual extension. For more details we refer to (Dorow et al., 2009).

The basic idea of SimRank is to consider two nodes as similar if they have similar neighborhoods. Node similarity scores are recursively computed from the scores of neighboring nodes: the similarity S_{ij} of two nodes i and j is computed as the normalized sum of the pairwise similarities of their neighbors:

$$S_{ij} = \frac{c}{|N(i)|\ |N(j)|} \sum_{k \in N(i), l \in N(j)} S_{kl}.$$

where $N(i)$ and $N(j)$ are the sets of i's and j's neighbors. As the basis of the recursion, S_{ij} is set to 1 if i and j are identical (self-similarity). The constant c ($0 < c < 1$) dampens the contribution of nodes further away. Following Jeh and Widom (2002), we use $c = 0.8$. This calculation is repeated until, after a few iterations, the similarity values converge.

For bilingual problems, we adapt SimRank for comparison of nodes across two graphs A and B. In this case, i is a node in A and j is a node in B, and the recursion basis is changed to $S(i, j) = 1$ if i and j are a pair in a predefined set of node-node equivalences (seed translation pairs).

$$S_{ij} = \frac{c}{|N_A(i)|\ |N_B(j)|} \sum_{k \in N_A(i), l \in N_B(j)} S_{kl}.$$

Multi-edge Extraction (MEE) Algorithm To combine different information sources, corresponding to edges of different types, in one SimRank computation, we use multi-edge extraction (MEE), a variant of SimRank (Dorow et al., 2009). It computes an aggregate similarity matrix after each iteration by taking the average similarity value over all edge types $\mathcal{T}$:

$$S_{ij} = \frac{c}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \frac{1}{f(|N_{A,t}(i)|) f(|N_{B,t}(j)|)} \sum_{\substack{k \in N_{A,t}(i), \\ l \in N_{B,t}(j)}} S_{kl}.$$

f is a normalization function (either $f = g$, $g(n) = n$ as before or the normalization discussed in the next section).

While we have only reviewed the case of unweighted graphs, the extended SimRank can also be applied to weighted graphs. (See (Dorow et al., 2009) for details.) In what follows, all graph computations are weighted.

Square Root Normalization Preliminary experiments showed that SimRank gave too much influence to words with few neighbors. We therefore modified the normalization function $g(n) = n$. To favor words with more neighbors, we want f to grow sublinearly with the number of neighbors. On the other hand, it is important that, even for nodes with a large number of neighbors, the normalization term is not much smaller than $|N(i)|$, otherwise the similarity computation does not converge. We use the function $h(n) = \sqrt{n} * \sqrt{\max_k(|N(k)|)}$. h grows quickly for small node degrees, while returning values close to the linear term for large node degrees. This guarantees that nodes with small degrees have less influence on final similarity scores. In all experiments reported in this paper, the matrices $\tilde{A}$, $\tilde{B}$ are normalized with $f = h$ (rather than using the standard normalization $f = g$). In one experiment, accuracy of the top-ranked candidate (acc@1) was .52 for h and .03 for g, demonstrating that the standard normalization does not work in our application.

Threshold Sieving For larger experiments, there is a limit to scalability, as the similarity matrix fills up with many small entries, which take up a large amount of memory. Since these small entries contribute little to the final result, Lizorkin et al. (2008) proposed *threshold sieving*: an approximation of SimRank using less space by deleting all similarity values that are below a threshold. The quality of the approximation is set by a parameter δ that specifies maximum acceptable difference of threshold-sieved similarity and the exact solution. We adapted this to the matrix formulation by integrating the thresholding step into a standard sparse matrix multiplication algorithm.

We verified that this approximation yields useful results by comparing the ranks of exact and approximate solutions. We found that for the high-ranked words that are of interest in our task, sieving with a suitable threshold does not negatively affect results.

5 Benchmark Data Set

Rapp's (1999) original experiment was carried out on newswire corpora and a proprietary Collins dictionary. We use the free German (280M tokens) and English (850M tokens) Wikipedias as source and target corpora. Reinhard Rapp has generously provided us with his 100 word test set

	n	**a**	**v**
training set	.61	.31	.08
TS100	.65	.28	.07
TS1000	.66	.14	.20

Table 3: Percentages of POS in test and training

(TS100) and given us permission to redistribute it. Additionally, we constructed a larger test set (TS1000) consisting of the 1000 most frequent words from the English Wikipedia. Unlike the noun-only test sets used in other studies, (e.g., Koehn and Knight (2002), Haghighi et al. (2008)), TS1000 also contains adjectives and verbs. As seed translations, we use a subset of the dict.cc online dictionary. For the creation of the subset we took raw word frequencies from Wikipedia as a basis. We extracted all verb, noun and adjective translation pairs from the original dictionary and kept the pairs whose components were among the $5{,}000$ most frequent nouns, the $3{,}500$ most frequent adjectives and the 500 most frequent verbs for each language. These numbers are based on percentages of the different node types in the graphs. The resulting dictionary contains $12{,}630$ pairs: $7{,}767$ noun, $3{,}913$ adjective and 950 verb pairs. Table 3 shows the POS composition of the training set and the two test sets. For experiments evaluated on TS100 (resp. TS1000), the set of 100 (resp. 1000) English words it contains and all their German translations are removed from the seed dictionary.

Baseline. Our baseline is a reimplementation of the vector-space method of Rapp (1999). Each word in the source corpus is represented as a word vector, the dimensions of which are words of seed translation pairs. The same is done for corpus words in the target language, using the translated seed words as dimensions. The value of each dimension is determined by association statistics of word cooccurrence. For a test word, a vector is constructed in the same way. The labels on the dimensions are then translated, yielding an input vector in the target language vector space. We then find the closest corpus word vector in the target language vector space using the city block distance measure. This word is taken as the translation of the test word.

We went to great lengths to implement Rapp's method, but omit the details for reasons of space. Using the Wikipedia/dict.cc-based data set, we achieve 50% acc@1 when translating words from English to German. While this is somewhat lower than the performance reported by Rapp, we believe this is due to Wikipedia being more heterogeneous and less comparable than news corpora from identical time periods used by Rapp.

Publication. In conjunction with this paper we publish the benchmark for bilingual lexicon extraction described. It consists of (i) two Wikipedia dumps from October 2008 and the linguistic relations extracted from them, (ii) scripts to recreate the training and test sets from the dict.cc data base, (iii) the TS100 and TS1000 test sets, and (iv) performance numbers of Rapp's system and MEE. These can serve as baselines for future work. Note that (ii)–(iv) can be used independently of (i) – but in that case the effect of the corpus on performance would not be controlled. The data and scripts are available at `http://ifnlp.org/wiki/extern/WordGraph`

6 Results

In addition to the vector space baseline experiment described above, we conducted experiments with the SimRank model. Because TS100 only contains one translation per word, but words can have more than one valid translation, we manually extended the test set with other translations, which we verified using dict.cc and leo.org. We report the results separately for the original test set ("strict") and the extended test set in Table 4. We also experimented with *single*-edge models consisting of three separate runs on each relation.

The accuracy columns report the percentage of test cases where the correct translation was found among the top 1 (acc@1) or top 10 (acc@10) candidate words found by the translation models. Some test words are not present in the data at all; we count these as 0s when computing acc@1 and acc@10. The acc@10 measure is more useful for indicating topical similarity while acc@1 measures translation accuracy.

MRR is Mean Reciprocal Rank of correct translations: $\frac{1}{n}\sum_i^n \frac{1}{\text{rank}_i}$ (Voorhees and Tice, 1999). MRR is a more fine-grained measure than acc@n,

	TS100, strict			TS100, extended			TS1000		
	acc@1	acc@10	MRR	acc@1	acc@10	MRR	acc@1	acc@10	MRR
baseline	.50	.67	.56	.54	.70	.60	.33	.56	.41
single	.44	.67	.52	.49	.68	.56	.40‡	.70‡	.50
MEE	.52	.79†	.62	.58	.82†	.68	**.48**‡	.76‡	.58

Table 4: Results compared to baseline[*]

e.g., it will distinguish ranks 2 and 10. All MRR numbers reported in this paper are consistent with acc@1/acc@10 and support our conclusions.

The results for acc@1, the measure that most directly corresponds to utility in lexicon extraction, show that the SimRank-based models outperform the vector space baseline – only slightly on TS100, but significantly on TS1000. Using the various relations separately (single) already yields a significant improvement compared to the baseline. Using all relations in the integrated MEE model further improves accuracy. With an acc@1 score of 0.48, MEE outperforms the baseline by .15 compared to TS1000. This shows that a combination of several sources of information is very valuable for finding the correct translation.

MEE outperforms the baseline on TS1000 for all parts of speech, but performs especially well compared to the baseline for adjectives and verbs (see Table 5). It has been suggested that vector space models perform best for nouns and poorly for other parts of speech. Our experiments seem to confirm this. In contrast, MEE exhibits good performance for nouns and adjectives and a marked improvement for verbs.

On acc@10, MEE is consistently better than the baseline, on both TS100 and TS1000. All three differences are statistically significant.

6.1 Relation Comparison

Table 5 compares baseline, single-edge and MEE accuracy for the three parts of speech covered. Each single-edge experiment can compute noun similarity; for adjectives and verbs, only amod, dobj and MEE can be used.

Performance for nouns varies greatly depending on the relation used in the model. ncrd performs best, while dobj shows the worst performance. We hypothesize that dobj performs badly because (i) many verbs are semantically non-restrictive with respect to their arguments, (e.g., *use*, *contain* or *include*) and as a result semantically unrelated nouns become similar because they share the same verb as a neighbor; (ii) light verb constructions (e.g., *take a walk* or *give an account*) dilute the extracted relations; and (iii) dobj is the only relation we extracted with a syntactic parser. The parser was trained on newswire text, a genre that is very different from Wikipedia. Hence, parsing is less robust than the relatively straightforward POS patterns used for the other relations.

Similarly, many semantically non-restrictive adjectives such as *first* and *new* can modify virtually any noun, diluting the quality of the amod source. We conjecture that ncrd exhibits the best performance because there are fewer semantically non-restrictive nouns than non-restrictive adjectives and verbs.

MEE performance for nouns (.45) is significantly better than that of the single-edge models. The information about nouns that is contained in the verb-object and adjective-noun data is integrated in the model and helps select better translations. This, however, is only true for the noun

		noun	**adj**	**verb**	**all**
TS100	baseline	.55	.43	.29	.50
	amod	.15	.71	-	.30
	ncrd	.34	-	-	.22
	dobj	.02	-	.43	.04
	MEE	.45	.71	.43	.52
TS1000	baseline	.42	.26	.18	.33
	MEE	.53	.55	.27	.48

Table 5: Relation comparison, acc@1

[*] We indicate statistical significance at the $\alpha = 0.05$ (†) and 0.01 level (‡) when compared to the baseline. We did not calculate significance for MRR.

source	acc@1	acc@10
dobj	.02	.10
amod	.15	.37
amod+dobj	.22	.43
ncrd+dobj	.32	.65
ncrd	.34	.60
ncrd+amod	.49	.74
MEE	.45	.77

Table 6: Accuracy of sources for nouns

node type, the "pivot" node type that takes part in edges of all three types. For adjectives and verbs, the performance of MEE is the same as that of the corresponding single-edge model.

We ran three additional experiments each of which combines only two of the three possible sources for noun similarity, namely ncrd+amod, ncrd+dobj and amod+dobj and performed strict evaluation (see Table 6). We found that in general combination increases performance except for ncrd+dobj vs. ncrd. We attribute this to the lack of robustness of dobj mentioned above.

6.2 Comparison MEE vs. All-in-one

An alternative to MEE is to use untyped edges in one large graph. In this *all-in-one* model (AIO), we connect two nodes with an edge if they are linked by any of the different linguistic relations. While MEE consists of small adjacency matrices for each type, the two adjacency matrices for AIO are much larger. This leads to a much denser similarity matrix taking up considerably more memory. One reason for this is that AIO contains similarity entries between words of different parts of speech that are 0 (and require no memory in a sparse matrix representation) in MEE.

Since AIO requires more memory, we had to filter the data much more strictly than before to be able to run an experiment. We applied the following stricter thresholds on relationships to obtain a small graph: 5 instead of 3 for adjective-noun pairs, and 3 instead of 0 for verb-object pairs, thereby reducing the total number of edges from 2.1M to 1.4M. We also applied threshold sieving (see Section 4) with $\delta = 10^{-10}$ for AIO. The results on TS100 (strict evaluation) are reported in Table 7. For comparison, MEE was also run on the smaller graph. Performance of the two models is very similar, with AIO being slightly better (not significant). The slight improvement does not justify the increased memory requirements. MEE is able to scale to more nodes and edge types, which allows for better coverage and performance.

	$\textbf{MEE}_{small}$	$\textbf{AIO}_{small}$
acc@1	.51	.52
acc@10	.72	.75
MRR	.62	.59

Table 7: MEE vs. AIO

7 Analysis and Discussion

Error analysis. We examined the cases where a reference translation was not at the top of the suggested list of translation candidates. There are a number of elements in the translation process that can cause or contribute to this behavior.

Our method sometimes picks a cohyponym of the correct translation. In many of these cases, the correct translation is in the top 10 (together with other words from the same semantic field). For example, the correct translation of *moon*, *Mond*, is second in a list of words belonging to the semantic field of celestial phenomena: Komet *(comet)*, **Mond** *(moon)*, Planet *(planet)*, Asteroid *(asteroid)*, Stern *(star)*, Galaxis *(galaxy)*, Sonne *(sun)*, ... While this behavior is undesirable for strict lexicon extraction, it can be exploited for other tasks, e.g. cross-lingual semantic relatedness (Michelbacher et al., 2010).

Similarly, the method sometimes puts the antonym of the correct translation in first place. For example, the translation for *swift* (*schnell*) is in second place behind *langsam* (*slow*). Based on the syntactic relations we use, it is difficult to discriminate between antonyms and semantically similar words if their syntactic distributions are similar.

Ambiguous source words also pose a problem for the system. The correct translation of *square* (the geometric shape) is *Quadrat*. However, 8 out of its top 10 translation candidates are related to the *location* sense of *square*. The other two are geometric shapes, *Quadrat* being listed second. This is only a concern for strict evaluation, since correct translations of a different sense were included in the extended test set.

bed is also ambiguous (piece of furniture vs. river bed). This introduces translation candidates from the geographical domain. As an additional source of errors, a number of *bed*'s neighbors from the furniture sense have the German translation *Bank* which is ambiguous between the furniture sense and the financial sense. This ambiguity in the target language German introduces spurious translation candidates from the financial domain.

Discussion. The error analysis demonstrates that most of the erroneous translations are words that are incorrect, but that are related, in some obvious way, to the correct translation, e.g. by cohyponymy or antonymy. This suggests another application for bilingual lexicon extraction. One of the main challenges facing statistical machine translation (SMT) today is that it is difficult to distinguish between minor errors (e.g., incorrect word order) and major errors that are completely implausible and undermine the users' confidence in the machine translation system. For example, at some point Google translated "sarkozy sarkozy sarkozy" into "Blair defends Bush". Since bilingual lexicon extraction, when it makes mistakes, extracts closely related words that a human user can understand, automatically extracted lexicons could be used to discriminate smaller errors from grave errors in SMT.

As we discussed earlier, parallel text is not available in sufficient quantity or for all important genres for many language pairs. The method we have described here can be used in such cases, provided that large monolingual corpora and basic linguistic processing tools (e.g. POS tagging) are available. The availability of parsers is a more stringent constraint, but our results suggest that more basic NLP methods may be sufficient for bilingual lexicon extraction.

In this work, we have used a set of seed translations (unlike e.g., Haghighi et al. (2008)). We believe that in most real-world scenarios, when accuracy and reliability are important, seed lexica will be available. In fact, seed translations can be easily found for many language pairs on the web. Although a purely unsupervised approach is perhaps more interesting from an algorithmic point of view, the semisupervised approach taken in this paper may be more realistic for applications.

In this paper, we have attempted to reimplement Rapp's system as a baseline, but have otherwise refrained from detailed comparison with previous work as far as the accuracy of results is concerned. The reason is that none of the results published so far are easily reproducible. While previous publications have tried to infer from differences in performance numbers that one system is better than another, these comparisons have to be viewed with caution since neither the corpora nor the gold standard translations are the same. For example, the paper by Haghighi et al. (2008) (which demonstrates how orthography and contextual information can be successfully used) reports 61.7% accuracy on the 186 most confident predictions of nouns. But since the evaluation data sets are not publicly available it is difficult to compare other work (including our own) with this baseline. We simply do not know how methods published so far stack up against each other.

For this reason, we believe that a benchmark is necessary to make progress in the area of bilingual lexicon extraction; and that our publication of such a benchmark as part of the research reported here is an important contribution, in addition to the linguistically grounded extraction and the new graph-theoretical method we present.

8 Summary

We have presented a new method, based on graph theory, for bilingual lexicon extraction without relying on resources with limited availability like parallel corpora. We have shown that with this graph-theoretic framework, information obtained by linguistic analysis is superior to cooccurrence data obtained without linguistic analysis. We have presented multi-edge extraction (MEE), a scalable graph algorithm that combines different linguistic relations in a modular way. Finally, progress in bilingual lexicon extraction has been hampered by the lack of a common benchmark. We publish such a benchmark with this paper and the performance of MEE as a baseline for future research.

9 Acknowledgement

This research was funded by the *German Research Foundation* (DFG) within the project *A graph-theoretic approach to lexicon acquisition*.

References

Dorow, Beate, Florian Laws, Lukas Michelbacher, Christian Scheible, and Jason Utt. 2009. A graph-theoretic algorithm for automatic extension of translation lexicons. In *EACL 2009 Workshop on Geometrical Models of Natural Language Semantics*.

Dunning, Ted. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Evert, Stefan. 2004. *The Statistics of Word Cooccurrences - Word Pairs and Collocations*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung (IMS), Universität Stuttgart.

Fung, Pascale and Lo Yuen Yee. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *COLING-ACL*, pages 414–420.

Garera, Nikesh, Chris Callison-Burch, and David Yarowsky. 2009. Improving translation lexicon induction from monolingual corpora via dependency contexts and part-of-speech equivalences. In *CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 129–137, Morristown, NJ, USA. Association for Computational Linguistics.

Haghighi, Aria, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio, June. Association for Computational Linguistics.

Jeh, Glen and Jennifer Widom. 2002. Simrank: A measure of structural-context similarity. In *KDD '02*, pages 538–543.

Koehn, Philipp and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 9–16.

Lizorkin, Dmitry, Pavel Velikhov, Maxim N. Grinev, and Denis Turdakov. 2008. Accuracy estimate and optimization techniques for simrank computation. *PVLDB*, 1(1):422–433.

Michelbacher, Lukas, Florian Laws, Beate Dorow, Ulrich Heid, and Hinrich Schütze. 2010. Building a cross-lingual relatedness thesaurus using a graph similarity measure. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may.

Rapp, Reinhard. 1999. Automatic identification of word translations from unrelated English and German corpora. In *COLING 1999*.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Schmid, Helmut. 2004. Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING '04*, page 162.

Voorhees, Ellen M. and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In *Proceedings of the 8th Text Retrieval Conference*.