HMMs, GRs, and n-grams as lexical substitution techniques – are they portable to other languages?

Judita Preiss
Department of Linguistics
The Ohio State University
judita@ling.ohio-state.edu

Andrew Coonce
The Ohio State University
coonce.3@osu.edu

Brittany Baker
The Ohio State University
baker.1189@osu.edu

Abstract

We introduce a number of novel techniques to lexical substitution, including an application of the Forward-Backward algorithm, a grammatical relation based similarity measure, and a modified form of *n*-gram matching. We test these techniques on the Semeval-2007 lexical substitution data [McCarthy and Navigli, 2007], to demonstrate their competitive performance. We create a similar (small scale) dataset for Czech, and our evaluation demonstrates language independence of the techniques.

Keywords

Lexical substitution, synonyms, Google n-gram corpus, grammatical relations, HMMs, Forward-Backward algorithm, Lucene, Czech, word sense disambiguation

1 Introduction

We present a number of novel approaches to lexical substitution, a task which for a given target word, requires the selection of a suitable alternative word. Our highly modular system not only allows a trivial addition of new modules, but also explores the applicability of the techniques to English and Czech.

Lexical substitution was suggested as a way of evaluating word sense disambiguation [McCarthy, 2002], accounting for the difficulties with selecting a sense inventory in the traditional direct sense evaluations (e.g., [Preiss and Yarowsky, 2002]). In the lexical substitution task, instead of being presented with a set of possible senses to choose from, a system is given a word and is required to find a suitable alternative given the context. For example, the word *bright* in the sentence

> His parents felt that he was a bright boy.

can be replaced with the word *intelligent*. However, the same substitution for the same word in the context of the word *star* (e.g., in the sentence *Our Sun is a bright star.*) is unlikely to reflect the intended meaning. The applications of a system capable of making such substitutions lie in question answering, summarisation, paraphrase acquisition [Dagan et al., 2006], text simplification and lexical acquisition [McCarthy, 2002].

An evaluation task was set up as part of Semeval-2007 evaluation exercises [McCarthy and Navigli, 2007], in which participants were given a target word and its context and were expected to find a suitable substitutable word or phrase. A second task is proposed for Semeval-2010 [Sinha et al., 2009], which expects participants to select a possible substitute from another language, given an input word and context in English.

As with many natural language processing tasks, most work on lexical substitution has been carried out in English. As the lexical substitution task requires an annotated corpus, it is non-trivial to carry out large-scale experiments in other languages. We create a small corpus for Czech, and evaluate our lexical substitution modules[1] not only on the Semeval-2007 lexical substitution data in English, but also on our Czech dataset. Unlike the proposed [Sinha et al., 2009] cross-lingual lexical substitution task in Semeval-2010, in our experiment the target words and contexts as well as substitute are all in Czech.

For English, we demonstrate

1. the importance of refining the set of input candidate substitutes prior to a candidate ranking module being run, and
2. show our modules' suitability to be used in lexical substitution tasks

We create a communal set of candidates, which are used by three independent modules: a grammatical relation [Briscoe et al., 2002] based module investigating the syntactic (and to a certain extent semantic) similarity of contexts, an *n*-gram module exploiting the Google Web 1T 5-gram corpus [Brants and Franz, 2006], and a module discovering the optimal path through the sentence based on the Forward-Backward HMM algorithm (e.g., [Roark and Sproat, 2007]).

[1] Note that such an evaluation is not possible for all of our modules, due to the lack of available tools for the Czech language.

Our paper is organized as follows: Section 2 describes the technique used to create a weighted candidate set, Sections 3, 4, and 5 contain the GR, n-gram and HMM modules respectively. An initial evaluation on English is presented in Section 6. Our experiment on Czech, and the data used to enable this, appears in Section 7, with conclusions drawn in Section 8.

2 Building a candidate set

We create a very modular system, where each of our lexical substitution selection methods is entirely independent of the others. The modules share a common input: the possible set of candidates for each word. In his work, [Yuret, 2007] presents the most successful system in SEMEVAL-2007 and comments that "I spent a considerable amount of effort trying to optimize the substitute sets". We therefore explore performance with two different candidate sets to investigate the hypothesis that the approach used is as important as the candidates selected.

The first approach, which finds the maximum possible performance of the modules (an upper bound), is given all candidates which appeared for the target word in the gold standard data. I.e., all the possible substitutes from the gold standard are gathered together, and given to the modules as possible candidates. (However, as no module is designed to cope with multiword candidates, all the multiword candidates are removed.)

Our second set of candidates is constructed from WordNet [Miller et al., 1990] and the online encyclopedia Encarta (`http://encarta.msn.com`) as follows:

- All WordNet (WN) synonyms of the target word are included (i.e., synonyms of all the possible senses of the correct part of speech)[2].
- The hypernym synset and the hyponym synset are also included for all possible senses.
- All possible Encarta synonyms of the correct part of speech are extracted.

A probability distribution is placed on these candidates based on these (manually selected) weights:

Source	Weight
WN synonym	3
WN hypernym	1
WN hyponym	2
Encarta	3

I.e., if a candidate appears both as a WN synonym and in Encarta, it will get a weight of 6, while if it is only appearing as a hyponym, it's weight will be 2. For example, for the test word *account* (noun):

1. WN synonyms: *history, chronicle, story, bill, invoice, report, story, explanation, . . .*
2. WN hypernyms: *record, importance, profit, gain, statement, . . .*
3. WN hyponyms: *etymology, annals, biography, life, recital, reckoning, tally, . . .*
4. Encarta: *report, description, story, narrative, explanation, version, interpretation, tally, . . .*

PoS	Average
Noun	56
Verb	127
Adjective	37
Adverb	9

Table 1: *Average number of candidates*

the Encarta synonyms add new candidates, while also boosting the weights of, e.g., the synonym *story*, or the hyponym *tally*. Once all the candidates for a target word are generated, the weights are converted into a probability distribution.[3] The average numbers of candidates for each part of speech are presented in Table 1.

While the GR and the n-gram modules only require a set of candidates for the target words, the HMM module requires potential candidates for all words in the sentence in order to find an optimal path through the data. These candidates were generated in the same manner, with PoS tags drawn from the [Elworthy, 1994] tagger (executed as part of RASP [Briscoe et al., 2006]), with the exception that for the non-target words, the original word was also included in the candidate set.

3 Grammatical relations

Given the candidates generated in Section 2, we create several different (hopefully complementary) modules. A combination of these can then utilize the different strengths and weakness of each approach to create a more accurate ranking of proposed candidates overall. The modules can therefore run independently to select the most likely of any given candidates.

For each target word, its context was parsed with RASP [Briscoe et al., 2006] producing grammatical relations (GRs). GRs, mainly binary relations expressing information about predicates and arguments, provide a good means for capturing both syntactic structural information, but also some sense of semantic meaning as well [Briscoe et al., 2002]. GR such as

```
(dobj give dog)
```

where the GR is `dobj` (direct object), it not only tells us that give directly dominates dog (syntax), but there is also a description about a patient relationship.

The main advantage of GRs, as opposed to, for example, n-grams, is the possibility of GRs encoding long distance dependencies. Even with simple sentences, such as:

- *Bob Smith gave the bone to the dog.*
- *Bob Smith gave the big juicy bone to the dog.*

[2] The part of speech of the target word is given in the data.

[3] The low hypernym score is due to relatively rare occurrence of the correct candidate being in the hypernym set.

the GRs will contain the `dobj give bone` relation for both sentences, while a five word n-gram centered on the target word *give* will not even mention the word *bone* in the second case.

The motivation behind this approach is in the assumption that a word which is a valid lexical substitute will appear in the same GRs as the target word. This requires a large corpus annotated with GRs: to this end we employ Gigaword [Graff, 2003], a large collection of English text, which we parsed with the RASP parser and collected information about frequencies of GR occurrences. The GR occurrences are indexed using Lucene, to allow incremental building and searching of the dataset. Each word can be queried, producing a listing of every applicable GR in which said word appeared in the Gigaword corpus, along with a frequency count of occurrence(s) for each GR. A preliminary search was performed on this index to obtain initial probabilities for each GR.

For each target word, all the GRs from its context are extracted and the target word is substituted with a possible candidate. The frequency of this GR is extracted from the parsed corpus, and divided by the probability of that GR, in order to account for unequal GR occurrences throughout the index (the GR `ncmod`, for example, appeared many times more than the GR `iobj`). For each candidate, all its GR frequency weights are summed, and the weights are normalized to produce a probability distribution on candidates.

4 n-grams

Approaches based on n-grams drawn from the Google Web1T corpus [Brants and Franz, 2006] have been shown to constitute a particularly good approach to lexical substitution with the best performing system in SEMEVAL-2007 being n-gram based [Hassan et al., 2007]. The basic algorithm for such an approach is very clear: an n-gram containing the chosen word is extracted from the context, the chosen word is then replaced with a candidate and the frequency of the newly formed n-gram is found. The candidate with the highest frequency wins.

For this work, we use the Google Web1T corpus, a publicly available resource, containing frequency information about 1, 2, 3, 4 and 5-grams drawn from one trillion (10^{12}) words of English Web text, subject to a minimum occurrence threshold. While such a corpus is obviously a very valuable resource, it has been found previously that it is difficult to use due to its sheer size (it is 25Gb in compressed form). In order to provide a reasonable access time (and multiple wildcard searches), we treated each 5-gram as a separate document and indexed the 5-gram corpus with the publicly available tool Lucene (available from `http://lucene.apache.org`).[4]

For a word w with possible candidate substitutions $s_1, s_2, \ldots, s_n$, we exploit a 5 word window W centered around w in the following way for each s_i:

- We search for the frequency ($f_{5-gm}(s_i)$) of the 5-gram W with w replaced with s_i.
- The replaced 5-gram is also searched in a stoplisted form ($f_{stop}(s_i)$). Note that this can result in a much smaller n-gram.
- The frequencies of all consecutive subset 4-grams (with the target word w replaced with s_i) are extracted ($f_{4-gm_j}(s_i)$ for $j = 1, \ldots, 4$).
- The absolute frequency of the unigram s_i is also retrieved ($f_{1-gm}(s_i)$). A more frequent unigram is more likely to be found as part of a 5-gram or 4-gram, purely due to the frequency of occurrence. This factor allows us to remove this bias.

The resulting weight of each s_i is then expressed as shown in Figure 1.[5]

5 Hidden Markov Models

5.1 Introduction

Hidden Markov Models (e.g., [Roark and Sproat, 2007]) and, in particular, Forward-Backward Hidden Markov Models (HMMs), have a strong history of applications in linguistics. The justification for the applicability of a Hidden Markov Model to the problem of lexical substitution lies in both the limited number of possible substitutions and the large training corpus.

When compared to the issue of speech processing, for which a HMM is known to work as a reasonable model, the issue of lexical substitution is highly similar and can be expected to produce results of similar quality.

Meanwhile, the presence of the large training corpus[6] means that the transition probabilities can be calculated with a high degree of certainty for transitions between possible lexical substitutions.

5.2 Motivation

Compared to n-gram and grammatical relation (GR) models, the HMM introduce a few key distinctions which should have significant contributions to the quality of the substitution results. While the n-gram and GR algorithms are capable of comparing the likelihood of a lexical substitution in their respective contexts, they do not allow the non-target words to take on other senses in order to generate a better fit.

That said, the HMM lacks the ability of the GR model to consider the impact of grammar on the sentence. Furthermore, it does not benefit from the relative speed of implementation and execution enjoyed by n-grams.

The forward-backward algorithm allows the model to take into account both the words that preceded and followed the target word that was being disambiguated. In comparison, a Viterbi Algorithm would

[4] Note that subject to a predictably regular repetition, the information contained in the 2, 3, and 4-grams can be extracted from the 5-gram corpus.

[5] As this module is not expected to be acting alone, we are not making any adjustments for data sparsness.

[6] In this case, Google Web1T data is used to generate the transition probabilities.

$$p(s_i) = \frac{f_{5-gm}(s_i) + f_{stop}(s_i) + \sum_{j=1}^{4} f_{4-gm_j}(s_i)}{\sum_{k=1}^{n}(f_{5-gm}(s_k) + f_{stop}(s_k) + \sum_{j=1}^{4} f_{4-gm_j}(s_k)) + f_{1-gm}(s_i)}$$

Fig. 1: *The weight of each candidate s_i*

have limited the effectiveness of the solution to take into consideration words that follow the target word. For example, returning to the previous example

> Brian is a bright boy.

the key word in determining the proper lexical substitution of *bright* is *boy*. In this case, the Viterbi Algorithm would not be able to determine the proper substitution as the determining word *boy* follows the target word *bright*.

5.3 Algorithm

The key inputs to the HMM implementation are:

- S_i is one specific possible lexical substitution within the set of all possible substitutions S
- $B_{w_t i}$, or $P(W_t \rightarrow S_i)$, is the substitution probability of a word W_t by a substitution S_i[7]
- A_{ij}, or $P(S_i \rightarrow S_j)$, is the transition probability between two possible substitutions S_i and S_j[8]
- π_i, or $P(\emptyset \rightarrow S_i)$, is the probability that the model begins simulation in a given state S_i[9]

In implementation, the Forward-Backward Algorithm maximizes the product of the forward-looking matrix, α_{it}, the backward-looking matrix, β_{it}, and the lexical substitution probability, $b_{w_t i}$. The forward-looking matrix α_{it} measures the likelihood that the sentence is at state S_i, at time t, when the word w_t is registered. Likewise, the backward-looking matrix β_{it} measures the likelihood, given that the sentence is at state S_i at time t with probability α_{it}, that there is a valid transition path that reaches the end of the sentence. The lexical substitution likelihood probability $b_{w_t i}$ represents the relative, context-free probability that a given word w_t uses the substitution S_i.

Thus, the product $\alpha_{it} \times \beta_{it} \times b_{w_t i}$ represents the relative probability that the lexical substitution S_i is the intended sense of the word W_t seen in location t of the sentence. By comparing this product for each $S_i \in S$ and dividing the resulting values by the summation of the probabilities for all $S_i \in S$, the relative probabilities represent the likelihood that a specific word is the expected lexical substitution. The candidate with the highest likelihood estimation wins, though any substitution with a probability within two orders of magnitude of the winner is included as a possible solution for evaluation purposes.

5.4 Solution-Space Generalizations

In an ideal model, each sentence would be broken down into its constituent words and every possible substitution of each word would be a possibility interpretation. This idealized model would allow for all possible interpretations of the sentence, providing all possible frames with which to consider a given lexical substitution. Such a model would feature upwards of twenty possible substitutions per word with each requiring processing for all possible preceding and following substitutions.

The complexity of the HMM was found to be proportional to the square of the average number of possible lexical substitutions per word in its input sentences (see Table 2). This idealized model, though loss-less, proved computationally inefficient when scaled to the demands of the application, given the large percentage of time spent looking up transition probabilities in the training corpus. In order to minimize the total number of senses being processed without subjecting the model to unnecessary generalizations, two methods were used to reduce the solution complexity: sliding-window and sense-reduction generalizations.

The sliding-window generalization assumed that terms further from the target word would be less likely to contain useful information to disambiguate the target word sense. As such, a sliding-window representing likely relevant words was formed around each of the target words; any word not within the sliding-window had its possible word senses (expressed by lexical substitutes) reduced to unity while those within the window retained multiple senses.

The sense-reduction generalization assumed that word-senses with low probabilities would not contribute significant information to disambiguating the word-sense of the target word. As such, the senses were reduced by limiting possible sense for words within the sliding-window to only those senses that were common to both Encarta and WordNet.

6 Results

We evaluated various combinations of the above systems on the English lexical substitution data [McCarthy and Navigli, 2007], which contains substitution information about 171 nouns, verbs, adjectives and adverbs manually constructed by 5 native English speaker annotators. Each of our modules is capable of producing a probability distribution which allows us to investigate a number of possible combination techniques. All systems are given identical candidate sets as input, yielding two experiments:

1. Candidate set created from the gold standard (GS)

[7] The candidate sets, S, and their substitution probabilities, $B_{w_t i}$, are shared with the other applications discussed in this paper.

[8] The transition probability, A_{ij}, is generated from the Google 2-gram data set using Lucene.

[9] The initial state probability, π_i, is generated from the Google 1-gram data set using Lucene.

Without Sense-Reduction	545 lexical substitution candidates/sentence
Non-Target Sense-Reduction	88 lexical substitution candidates/sentence
Full Sense-Reduction	45 lexical substitution candidates/sentence
Without Rolling-Window	83 Lucene queries/sentence
With Rolling-Window	24 Lucene queries/sentence

Table 2: *Hidden Markov Model Computational Complexity*

Eval	System	Candidates	Precision	Recall	Mode precision	Mode recall
OOT	GRs	GS	63.49	7.23	71.05	8.78
Best	GRs	GS	5.58	0.64	6.58	0.81
OOT	HMMs	GS	52.74	43.41	63.41	52.28
Best	HMMs	GS	13.64	11.23	18.34	15.12
OOT	n-grams	GS	65.06	65.02	73.80	73.74
Best	n-grams	GS	12.31	12.30	17.33	17.32
OOT	Voting	GS	68.67	68.67	77.80	77.80
Best	Voting	GS	13.90	13.90	19.59	19.59
OOT	GRs	WNE	13.68	0.09	12.50	0.08
Best	GRs	WNE	1.82	0.01	0.00	0.00
OOT	HMMs	WNE	16.52	0.25	20.00	0.33
Best	HMMs	WNE	2.24	0.03	0.00	0.00
OOT	n-grams	WNE	35.79	8.90	48.11	12.44
Best	n-grams	WNE	6.92	1.72	11.01	2.85
OOT	Voting	WNE	36.07	8.98	48.43	12.52
Best	Voting	WNE	7.02	1.75	11.01	2.85

Table 3: *Results of each module on the English lexical substitution task*

2. Candidate set created from WordNet and Encarta as described in Section 2 (WNE).

The results of these evaluation can be found in Table 3. Two evaluations are presented:

1. **best**: Only the top candidate is evaluated against the gold standard (this corresponds to the highest probability candidate).

2. **oot**: The top ten candidates are collected and evaluated against the gold standard.

The results can be compared to the highest performing system in Semeval-2007 which achieved an oot precision / recall of 69.03 / 68.90, and mode precision / recall of 58.54, while the highest performing best system had a precision / recall of 12.90, and mode precision / recall of 20.65. (Note that the results for the WNE experiment are partial, as discussed in Section 6.1 representing only 10% of the data.)

6.1 Discussion

The single largest factor in the effectiveness of an approach to the problem space appears to be the proper determination of the scope of its candidate list. If an under-generated candidate set was used, the lexical substitutions suggested would be technically sound but incorrect insofar as they were only the best from the subset, not from the set of all possible substitutions. Omission of candidates could also reduce the number of valid substitutions to zero, creating a model where no candidate that remained would fit within the constraints imposed by the system evaluating its candidacy.

While under-generation was a concern, the candidate sets more directly suffered from over-generation. In over-generated candidate sets, the inclusion of rarely used substitutions, including hypernyms and hyponyms, served only to dramatically increase solution time without a corresponding increase in solution accuracy. As the complexity of the systems frequently increased proportional to the square of the average number of lexical substitution possibilities, these candidate sets quickly became disproportionately large when compared to the gold standard candidate sets. For such candidate sets that were fully evaluated, no noticable improvement was found in the ability to correctly identify the proper lexical substitution over the gold standard candidates.

These issues served as the motivations for proceding using the gold standard candidates (GS results) instead of the locally generated sets (WNE results). The gold standard candidates avoided the potential shortfall of under-generation as they were guaranteed to contain the anticipated substitution of the target word within their candidate sets; thus, protecting them from failing to produce a candidate selection. At the same time, the candidate list was also small enough to avoid the growth issues experienced in the over-generated candidate lists. Since the gold standard candidates do not overlap within their set, they are significantly more likely to feature a broad selection of possible candidates within the OOT, boosting the accuracy of the results. As we are merely interested in the performance of our modules (to demonstrate their suitability

Czech	PoS	Senses	English
cesta	n	5	way, path
číslo	n	6	number, performance
funkce	n	8	function, event
zůstat	v	6	stay, remain
těžký	a	6	hard, difficult
nechat	v	16	leave
důkaz	n	9	proof
povrch	n	7	surface
partie	n	12	part, partner
věc	n	6	thing
akce	n	7	action, event

Table 4: *Words selected for Czech lexical substitution including (some) English translations*

for the task and to enable their evaluation on the Czech lexical sample task), the use of the gold standard candidate sets is justifiable. Also, a properly generated candidate list would exhibit similar characteristics to this set.

7 Evaluation on Czech data

7.1 Creating the evaluation corpus

Unfortunately a lexical substitution corpus is not available for other languages. In an effort to investigate the applicability of our methods to other languages, we selected an extreme example: Czech, a highly morphologically rich, free order language, which should therefore produce a valuable comparison.

Ten words were selected at random from the online, publicly available, Czech Wiktionary[10] subject to the constraint that they had at least 5 senses listed (note that this step is completely automated, and could be executed with any language). The words chosen, along with the number of senses and their parts of speech in Wiktionary can be found in Table 4. The most frequent English translations are also provided. Ten sample sentences for each of these words (where the target word is to be substituted) were extracted from the Prague Dependency Treebank 2.0 [Hajičová, 1998], which contains markup of lemmatized form and thus allows various instances of use to be extracted. The annotation was done by a single native Czech speaker.

Due to the absence of a freely available parser providing GRs for Czech, it was only possible to run the n-gram and HMM modules in this experiment. Also, after initial experiments with using the Czech Wikipedia as training data, a further inflection problem came to light: should the candidate substitute be of a different gender to the original target word, the sentence stopped being grammatically correct when the candidate was substituted due to agreement. Thus a same animacy / type candidate would always be preferred. Consider the example:

> ...vstoupit do chrámu za účelem policejní <head>akce</head>

PoS	Average
Noun	7
Verb	8
Adjective	7

Table 5: *Average numbers of candidates for Czech*

Evaluation	Precision	Recall
Best	18.86	18.86
OOT	92.11	92.11

Table 6: *Czech lexical substitution*

if the correct substitute for the word *akce*, *čin* is used, the sentence needs to change to:

> ...vstoupit do chrámu za účelem policejního činu

The test data, and the training data, therefore required lemmatization: in the absence of a freely available lemmatizer for Czech, the PDT was used for both training and testing (with the test sentences being withheld from training). Thus n-grams (for $n = 1, 2, 5$) were acquired from this data, and indexed as carried out for English.

The candidates for each word were acquired from the Czech online synonyms resource (`http://www.synonyma-online.cz`), but the candidates for target words were also augmented by semi-automatically extracted synonyms from Wikipedia. The average numbers of candidates are presented in Table 5, and the combined results for the Czech lexical sample are presented in Table 6.

8 Conclusion and future work

We have presented a modular lexical substitution system which incorporates a number of novel approaches to the task. The approaches were shown to have good performance on the English lexical substitution data, while also being highly portable to other, potentially very different, languages (with a very good performance on the Czech data). We highlight the importance of a comprehensive, yet not over-generated candidate set, an issue which we fell has not been addressed enough in the past.

8.1 Future work

The GR module did not deal with issues of sparsness – the motivation being that the other modules will fill in. However, an alternative method for future work could be in grouping GRs together in meaningful ways [Pereira et al., 1993].

The HMM implemented a 1^{st}-Order Forward-Backward Algorithm. This introduces certain limitations to the transition probability matrices. If our running example had been

> Brian is a bright and lively boy.

[10] Available from `http://cs.wiktionary.org/`

instead, the separation of *bright* and *boy* by the intervening words *and lively* would have the effect of neutralizing the impact of the determining word on the target word. In this case, the words that would have the greatest impact on *bright* would be *a* and *and*, neither of which would contribute a significant amount of information that could lead to a proper lexical substitution.

References

[Brants and Franz, 2006] Brants, T. and Franz, A. (2006). Web 1T 5-gram Version 1.

[Briscoe et al., 2002] Briscoe, E. J., Carroll, J., Graham, J., and Copestake, A. (2002). Relational evaluation schemes. In *Proceedings of the beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation*, pages 4–8.

[Briscoe et al., 2006] Briscoe, E. J., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006, Interactive Poster Session*.

[Dagan et al., 2006] Dagan, I., Glickman, O., Gliozzo, A., Marmorshtein, E., and Strapparava, C. (2006). Direct word sense matching for lexical substitution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 449–456.

[Elworthy, 1994] Elworthy, D. (1994). Does Baum-Welch re-estimation help taggers? In *Proceedings of the 4th Conference on Applied NLP*, pages 53–58.

[Graff, 2003] Graff, D. (2003). English gigaword. Technical report, Linguistic Data Consortium.

[Hajičová, 1998] Hajičová, E. (1998). Prague dependency treebank: From analytic to tectogrammatical annotations. In *Proceedings of 2nd TST*, pages 45–50.

[Hassan et al., 2007] Hassan, S., Csomai, A., Banea, C., and Mihalcea, R. (2007). UNT: SubFinder: combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on the Semantic Evaluations*.

[McCarthy, 2002] McCarthy, D. (2002). Lexical substitution as a task for WSD evaluation. In *Proceedings of the Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, pages 109–115.

[McCarthy and Navigli, 2007] McCarthy, D. and Navigli, R. (2007). Task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53.

[Miller et al., 1990] Miller, G., Beckwith, R., Felbaum, C., Gross, D., and Miller, K. (1990). Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.

[Pereira et al., 1993] Pereira, F., Tishby, F., and Lee, L. (1993). Distributional clustering of English words. In *Proceedings of the Association for Computational Linguistics*, pages 183–190.

[Preiss and Yarowsky, 2002] Preiss, J. and Yarowsky, D., editors (2002). *Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguating Systems*.

[Roark and Sproat, 2007] Roark, B. and Sproat, R. W. (2007). *Computational Approaches to Morphology and Syntax*. Oxford University Press.

[Sinha et al., 2009] Sinha, R., McCarthy, D., and Mihalcea, R. (2009). Semeval-2010 task 2: Cross-lingual lexical substitution. In *Proceedings of the NAACL-HLT 2009 Workshop: SEW-2009 - Semantic Evaluations*.

[Yuret, 2007] Yuret, D. (2007). KU: Word sense disambiguation by substitution. In *Workshop of SemEval*.