English Lexical Substitution Task

Akanksha Bansal
Department of Computer Science
The University of Texas at Austin
bansal@cs.utexas.edu

Gaurav Nanda
Department of Computer Science
The University of Texas at Austin
nanda@cs.utexas.edu

Abstract

In this paper, we present our approach to generate lexical substitutions, i.e replacement of a target word in context with a suitable alternative substitute. We use thesaurus data for generation of set of candidate replacements and propose a distributional similarity based model for ranking the set of candidates, by generating appropriate vector representations for words in context. We evaluated this model on the SemEval 2007 Lexical Substitution task(LexSub) data, which shows promising results. Our model outperforms the current state-of-the-art in one of the evaluation metric proposed in the task. We then propose further enhancements to effectively capture the syntactic and semantic relationship between the words.

1 Introduction

Natural-language inference is highly context-sensitive, the applicability of inference rules depends on word sense and even finer grained contextual distinctions in usage (Szpektor et al., 2007). While closely related to WSD, lexical substitution does not rely on explicitly defined sense inventories: the possible substitutions reflect all conceivable senses of the word, and the correct sense has to be ascertained to provide an accurate substitution.

The motivation of LexSub task (McCarthy and Navigli, 2007) is not only to evaluate the WSD capabilities, but also compare various lexical resources. The task to generate a high quality set of possible substitutions is challenging in itself, for two reasons. First, the available lexical resources are seldom complete in listing synonyms. Second, manually annotated substitutions show that not all synonyms of a word are appropriate in a given context, and many good substitutions have other lexical relation than synonymy to the original word. Earlier experiments, (Sinha and Mihalcea, 2009) suggest that WordNet gives the best set of candidates. Therefore, we are using Big huge Thesaurus (Watson, 2012), which is based on WordNet and Carnegie Mellon Pronouncing Dictionary to generate set of candidate substitutes.

Our work is towards determining the right way to use distributional vectors so that we can quantify the measure between words with similar meaning. Recently, the issue has been taken up by several approaches that include some kind of syntactic information, in part under the heading of distributional compositionality (Mitchell and Lapata, 2008); (Erk and Padó, 2008), in part as a syntax-sensitive contextualization (Thater et al., 2010). In all these approaches the contextual influence of the meaning of a target word is modeled through vector composition. The meaning of *w* in context *c* is represented by a vector obtained by combining the vectors of *w* and *c* using some operation such as component-wise multiplication or addition.

Motivated by these papers, we built our model which constructs context vectors using combination of component-wise multiplication and addition.

2 Problem Definition and Algorithm

2.1 Task Definition

We evaluate our model on predicting suitable replacements for the sentences from the LexSub task. The task involves finding a substitute for a word in the context of a sentence in following ways:

1. finding the *best* candidate given the context of the word token.

2. finding the *set* of *ten* candidate substitutes for the word.

The motivation for this lexical substitution task was to allow a representation of meaning that does not necessitate a pre-defined listing of senses. The sentences have been chosen such that a diverse set of contexts are present in the sentences. This task provides a means of examining the issue of word sense representation by providing an unbiased platform to all, that evaluates the inventories and contextual disambiguation models.

Systems are supposed to identify set of candidate synonyms for a given word, and identify which is the best fit in a given context, thus requiring some discrimination between contexts. The most popular way to accomplish this task is by making use of a predefined inventory such as WordNet and then ranking using a Language based model to rank the substitutions.

2.2 Algorithm Definition

2.2.1 Candidate Generation

The first step is to collect set of potential replacement candidates for a given target word. Here, context of the target word is not considered. Rather, all the possible synonyms are selected and further ranked in the later steps. As discussed earlier, we use Big Huge Thesaurus for candidate selection.

2.2.2 Candidate Ranking

We create a distributional model to represent word meanings. Each lemma is described as a high dimensional vector that records co-occurrence of context features over a large corpus. The pipeline from corpora to semantic space can be roughly split into two major steps: pre-processing the corpus to collect the relevant counts, and processing the extracted counts mathematically.

Dataset and Pre-processing: Intuitively, we need a large dataset to construct co-occurrence matrix used in generating distributional similarity model. We used two different datasets, Wikicorpus by (Reese et al., 2010) and Uwack by (Baroni et al., 2009) to create the distributional space. Wikicorpus contains approximately 750 million words corpus (around 10.GB of data) and Uwack is five times the size of WikiCorpus. Using two different corpora enabled us to analyze the impact of size of corpus on distributional space.

To leverage the syntactic relation of POS tagged words, we suffix all the words with first letter of the Penn Treebank codes for various POS tags. In this way, we are able to capture the polysemic occurrences of single word and categorize them based of the the POS category. We only include Nouns, Verbs, Adverbs and Adjectives in our vector space as we believe that these have highest impact on the words in context. As our data source is based on Wikipedia, we also pre-process our data to exclude all the non-dictionary words.

Semantic Space Generation: After creating co-occurrence files, we use python based DISSECT (Dinu, 2013) toolkit to construct the semantic space. Then, we apply Positive PointWise Mutual Information (PPMI) weighting scheme for feature selection.

$$pmi(r, c) = \log \frac{P(r, c)}{P(r)P(c)} \tag{1}$$

$$ppmi(r, c) = pmi(r, c) if pmi(r, c) \geq 0 else 0 \tag{2}$$

(Bullinaria and Levy, 2007) demonstrated that PPMI performs better than a wide variety of other weighting approaches when measuring semantic similarity with word context matrices.

Context Vector Generation: We make use of prototype-based approach where we compute a (prototype) vector for each word in isolation using the co-occurrence matrix, then modify this vector according to the context in a given sentence (Thater et al., 2009). Earlier work suggested using addition or multiplication of word vectors to generate a context vector. However, in our model, we make distinction between the words to the left and words to the right of the context word.

We hypothesize that if two words exist together, their vectors must be multiplied together to provide meaningful representation of the co-occurring words. Multiplication tends to zero out the undesired features and highlight only the relevant features. However, we believe that vectors of words to the left (L) and right (R) of the target word should not be multiplied, but added. Multiplying L and R vectors may remove many features which could be relevant to the target word. Therefore, we propose a model using combination of addition and multiplication (LRAdd).

Instead of picking up L and R words using a context window, we capture the linguistic relations between the words. We define certain contextual combinations which provides an effective way to capture the effect of the context on the meaning of the target word (T). For a given T, its context vector is constructed depending upon its POS tag.

```
if $T = Adjective$ then
    $Context \leftarrow T * N2R$
else if $T = Noun$ then
    $Context \leftarrow T * A2L + T * V2L + T * V2R$
else if $T = Verb$ then
    a) $Context \leftarrow T * A2L + T * N2L + T * A2R + T * N2R$
    b) $Context \leftarrow T * A2L + T * N2R$
    c) $Context \leftarrow T * A2L + T * N2R + T * D2L + T * D2R$
else if $T = Adverb$ then
    a) $Context \leftarrow T * V2L + T * V2R$
    b) $Context \leftarrow T * V2L + T * N2L + T * V2R + T * N2R$
end if
```

where, X2R represents first item on the right of X type where the types are Nouns(N), Verb(V), Adjective(A) and Adverb(D). For instance, N2L represents first Noun to the left of the context word. For verbs and adverbs, we experimented with different context vector combinations as mentioned above.

For example in the sentence : *She said he was a "bright" and independent boy*. The context vector for the word *bright* can be determined in following ways:

1. If LRAdd method is applied using our semantic rules $V_{Context} = V_{bright} * V_{boy}$.

2. If LRAdd method is applied, not taking into account any semantic relation and picking first L and first R word $V_{Context} = (V_{bright} * V_{independent}) + (V_{bright} * V_{he})$.

3. If Component wise multiplication/addition is performed between the L and R context word vectors, $V_{Context} = V_{bright} \odot V_{independent} \odot V_{he}$.

The only context vector which is able to capture the relationship between the adjective and the noun is when semantic rules are used.

Once we have context vector for the target word is generated, we create similar vectors using all the candidate substitutes and rank them using Cosine Similarity metric.

3 Experimental Evaluation

3.1 Evaluation Methodology

In order to evaluate the performance of the model for the task, we submit for each of the sentence in the test set (1700 sentences) a list of words which can replace the target word. The replacements are assumed to be ranked in the order of their likelihood of occurrence, as the replacement word. The better the set of candidates, the higher the chance that one or more appropriate replacements are found for the given context.

We measure the precision and the recall for four subtasks: *best normal*, which measures the precision and recall obtained when the first replacement provided by the system is selected; *best mode*, which is similar to *best normal*, but it gives credit only if the first replacement returned by the system matches the replacement in the gold standard data set that was most frequently selected by the annotators; *oot normal*, which is similar to best normal, but it measures the precision and recall for the top ten replacements suggested by the system; and *oot mode*, which is similar to best mode, but it again considers the top ten synonyms returned by the system rather than just one. For oot, our system does not report duplicates in the list of best ten candidates. If we do report duplicates in the results, we can artificially inflate the precision values by reporting best substitution ten times. (McCarthy and Navigli, 2009) suggest that, by reporting duplicates, we can theoretically achieve precision of 457% .

Let us assume that H is the set of annotators, namely $\{h_1, h_2, h_3, ...\}$, and T $\{t_1, t_2, t_3, ...\}$ is the set of test items for which the humans provide at least two responses. For each t_i we calculate m_i, which is the most frequent response for that item, if available. We also collect all r_j, which is the set of responses for the i item t_i from the annotator h_j. Let the set of those items where two or more annotators have agreed upon a substitute (i.e. the items with a mode) be denoted by TM, such that $TM \subseteq T$. Also, let $A \subseteq T$ be the set of test items for which the system provides more than one response. Let the corresponding set for the items with modes be denoted by AM, such that $AM \subseteq TM$. Let $a_i \in A$ be the set of system's responses for the item t_i. Thus, for all test items t, we have the set of guesses i from the system, and the set of responses from the human annotators. As the next step, the multi-set union of the human responses is calculated and the frequencies of the unique items is noted. Therefore, for item t_i we calculate R_i which is $\sum r_i^j$ and the individual unique item in R_i, say res, will have a frequency

$$precision = \frac{\sum_{a_i:i \in A} \frac{\frac{\sum_{res \in a_i} freq_{res}}{|a_i|}}{|H_i|}}{|A|}$$

$$recall = \frac{\sum_{a_i:i \in T} \frac{\frac{\sum_{res \in a_i} freq_{res}}{|a_i|}}{|H_i|}}{|T|}$$

$$Mode\ precision = \frac{\sum_{best\ guess_i \in AM} 1\ if\ best\ guess = m_i}{|AM|}$$

$$Mode\ recall = \frac{\sum_{best\ guess_i \in TM} 1\ if\ best\ guess = m_i}{|TM|}$$

Figure 1: Measure of score for *best*

$$precision = \frac{\sum_{a_i:i \in A} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|}}{|A|}$$

$$recall = \frac{\sum_{a_i:i \in T} \frac{\sum_{res \in a_i} freq_{res}}{|H_i|}}{|T|}$$

$$Mode\ precision = \frac{\sum_{a_i:i \in AM} 1\ if\ any\ guess \in a_i = m_i}{|AM|}$$

$$Mode\ recall = \frac{\sum_{a_i:i \in TM} 1\ if\ any\ guess \in a_i = m_i}{|TM|}$$

Figure 2: Measure of score for *oot*

associated with it, namely $freq_{res}$ Given this setting, the precision (P) and recall (R) metrics are defined. The score values are calculated as per the Figure 1 for the *best* task. The score values for the *oot* task are calculated as per Figure 2.

3.2 Results

We experimented with various ways of context vector generation as has been mentioned in the previous section. Table 1 contains the results comparing the performances of the models with LRAdd Approach plus syntactic awareness with addition and multiplication context vector generation approach. The work done by (Thater et al., 2010) suggests that multiplication of context with the target would be the ideal way to capture the essence of the context when composing context word. However our work suggests that multiplication of the target with individual context words, and then addition of the resultant vectors gives the best results. Multiplication of the context word vectors with the target word, removed noise from the resultant vectors, and addition of these resultant composite context vectors ensures that all the features in the context are captured.

As mentioned in previous section, we experimented with various ways of combining the linguistic nature of the context word on results. The best results were achieved when Verb's context was determined by $T * A2L + T * N2R + T * D2L + T * D2R$ and Adverb's context was determined by $T{*}V2L{+}T{*}N2L{+}T{*}V2R{+}T{*}N2R$. However, there was no significant variation in the results with either of the combination. This might be because we do not have significant amount of test data which might show the effect of various these semantic rules.

System	best normal	best mode	oot normal	oot mode
LRAdd	**15.61**	**26.67**	**40.8**	**57.56**
Multiplication	8.23	13.79	24.46	35.96
Addition	12.78	22.28	38.31	53.01

Table 1: Comparison of various context vector generation methods.

Our second set of experiments were focused on determining what type of vector space is able to perform the re-ranking task effectively. We experimented with various weighting schemes like PPMI, PLMI, etc. The best results were obtained when PPMI weighting was applied on the vector space. Table 2 shows the performance values without PPMI in first row and with PPMI in the second row. Then we saw that performing SVD on the vectors space reduced the performance of the model. The third row in Table 2 shows the results.

Table 2: Comparison with various Vectors space features when calculated with LRAdd method

System	best normal	best mode	oot normal	oot mode
Without PPMI	8.34	13.25	33.36	47.15
Only PPMI	**15.61**	**26.67**	**40.8**	**57.56**
PPMI + SVD	9.51	15.2	36.69	51.71

Additionally, in order to get a deeper understanding of the vector space models, we designed the system such that replacement candidates could be suggested partially from thesaurus and partially from vector space. Figure 3 captures the models performance with various such combinations.

Finally, we experimented with two different data sets Wikicorpus and Uwack so as to get an idea about how the performance of our model would get affected by the size of data used to create the vector space. Firstly, we tested the performance of the two data sets on the candidate reranking. Uwack performed about 8% better than Wikicorpus in just the re-ranking across all the

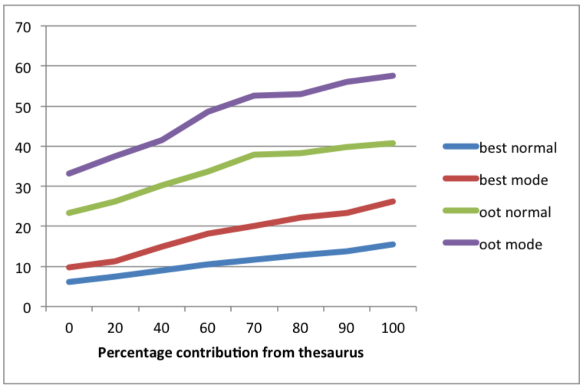

Figure 3: Candidate Suggestion set generated from Vector space and Thesaurus

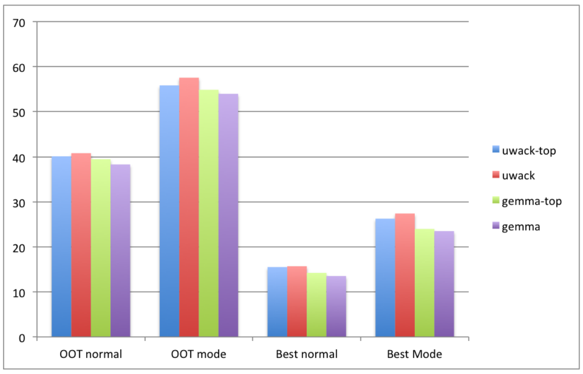

Figure 4: Uwack vs Wikicorpus performance comparison

measures. For the word *pound* Uwack is able to rank *lb* higher while Wikicorpus is not. The gold set for the noun usages of the word *pound* are *sterling; pn; kilo; 11.27 kilograms; per ounces; lb;*.

Secondly, our experiments indicate that selecting top 5000 features improve the performance of the model using Wikicorpus vector space. But degrade the performance for Uwack vector space(refer Figure 4). This can be explained by the fact that selecting top features would be helpful in the case where the vector space is sparse, because top feature selection would reduce the noise from the model. Uwack is a huge data source, therefore the vectors are very closely tied because the co-occurrence counts are significant rather than just being small values in between 1-10, thus by selecting top features we are loosing information.

3.3 Discussion

This section contains discussions on the results of all the experiments which were performed on the model.

Our results validate our hypothesis that our LRAdd approach when used in conjunction with the semantic rules provide us with the best results for ranking the set of candidates.

To generate the set of candidates, our experiments demonstrate that using distributional space in conjunction with thesaurus negatively impacts the performance. Looking at one of the instance, we found that, distributional space suggests replacements like *blue, yellow, dark* for the word *bright*, which are certainly not good replacements. Moreover, not all of the replacements are even present in any of the thesaurus, because they need certain intelligence of the world environment itself. For example, annotators have provided *11.27 kilograms* as replacement for the word *pound*. Then certain usages are not suggested by the thesaurus, but might get suggested by the distributional similarity space. For example the word *pound* when used as a noun could represent the weight metric or the currency or a pound force. Though thesaurus captures the phrase, ”british pound” they are not able to suggest *pn* as a replacement. However, if a distributional similarity space is created from scientific data set it might be able to suggest *pn* as it's replacement. To find replacements like *pn* was our prime motivation to try generating candidates using distributional space.

While generating distributional space, we observe that **SVD** reduces the number of dimensions which can be used to map the given vector space. It has the effect of intensifying the similarity between words which are closely related. Thus SVD can be very helpful in discounting low order co-occurrences. However, in the cases with indirect co-occurrence (i.e. words which appear in similar context), we start to loose the relationship strength and vector space model is not able to rank the words properly. Thus, SVD might help in the task if we made use of vector space for substitute candidate set generation.

The model is not able to perform well when the *figurative use of words* has been made. For example sentences which are looking for figurative meaning of the word: *blue* expect replacements like *sad*. However the vector space suggests replacements like *dark; gentle; dirty; bluish; sexy; low; northern; down; noble; spic;* Though by making use of thesaurus we are able to get words like *grim* in the replacement set. The exact replacement word *sad* is not found.

Figure 5 captures the performance of the model

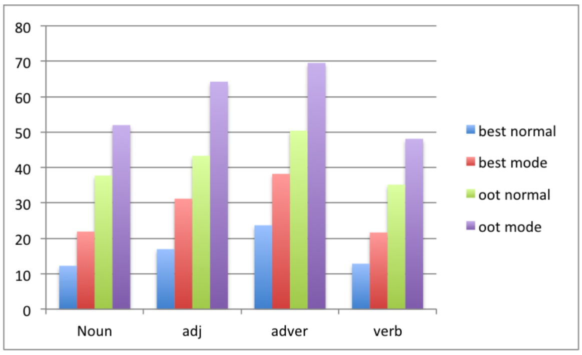

Figure 5: Part of Speech Split of the accuracy across the four measures.

in the various parts of speech. Our system performed worst on Noun and verbs. This would be because of the fact that many of the nouns present as the context words for the target are not even present in the vector space. Therefore, we believe that the performance of the model could be enhanced by making use of much more diverse vector space.

We believe the low performance for the verbs could be explained by two reasons. Firstly, nouns are the object and subject for the verbs and if certain nouns are not present in the vectors space then the context vector for verbs would not be able to capture the complete information. Secondly, because the rules that we have defined to capture the syntactic relation of the words with its context do not capture the subject-object relation which verbs need in order to capture their meanings. The noun which is nearest to the verb does not necessarily imply it will be the subject or object for that verb. Thus, we need to enhance that in order to be able to effectively define the context vectors for verbs.

Our model is very much dependent on the accuracy of the POS tagger used on the target sentence in order to be able to the right context words. There have been numerous instances where the performance of our model has suffered because of inaccuracies in this step itself.

4 Related Work

Polysemy is a fundamental problem for distributional models. Typically, distributional models compute a single vector for a target word, which contains co-occurrence counts for all the occurrences of the target in a large corpus. We make use of polysemy-aware approaches making use of syntactic context similar to what have been described in (Erk and Padó, 2008). In order to capture the multiple sense of the word in multiple Part of speech we define the vectors with regards to the POS tag.

(Mitchell and Lapata, 2008) investigate multiple ways for combination of distributional representations of word meaning along syntactic structure. They proposed to represent the meaning of a complex expression that consists of two syntactically related words *w* and *w'* by a vector obtained by combining the word vectors of *w* and *w'*, and find that component-wise multiplication performs best. Contrary to above, our work however indicates that best results are achieved when multiplication with context words and then addition of the resultant vectors is done.

There are several systems for synonym expansion that participated in the Semeval-2007 lexical substitution task (McCarthy and Navigli, 2009). Most of the systems used only one lexical resource, although two systems also experimented with two different lexical resources. Also, several systems used Web queries or Google N-gram data to obtain counts for contextual fitness. Our model performs better than any of the participating teams.

UNT (Hassan et al., 2007) was the best system for the *best mode* and *oot mode* which participated in the competition. It made use of WordNet and Encarta, along with back-and-forth translations collected from commercial translation engines, to generate the *candidate synonym collection* and N-gram-based models calculated on the Google Web 1T corpus for *context-based synonym fitness scoring*.

KU (Yuret, 2007) was the highest ranking system for the best normal metric. They used a statistical language model based on the Google Web 1T five-grams dataset to calculate the probabilities of all the synonyms to. In the development phase, it made use of WordNet and Roget's Thesaurus. In the test phase, it only uses the Roget resource.

(Szarvas et al., 2013) ML, suggests a supervised lexical substitution system. They train a global model on delexicalized features. They also use WordNet for candidate generation and various lexical, corpus-based and semantic features for training their machine learning model.

Table 4 shows a comparison between the results obtained from our system and other systems who have evaluated their systems on this task. Our

System	best normal	best mode	oot normal	oot mode
KU	12.90	20.65	46.15	61.30
UNT	12.77	20.73	49.19	66.26
ML	**15.94**	26.3	-	-
Our Model	15.61	**26.67**	40.8	57.56

system outperforms both supervised and unsupervised systems for the *best mode* metric and very high results for *best* metric.

5 Future Work

There are two areas where the performance of this model could be improved.

- Experimenting with other ways to combine the vectors.
- Testing with compositional training data.

(Thater et al., 2009) were able to improve the verb recall and precision values by making use of subject object relationship. They have *oot normal* as 66.20. Our model has not been able to perform that well on verbs. Their work indicates that there are better ways to capture the relationship with the context for verbs. Thus in order to make use of the syntactic tree in order to determine the lingustic relationship between the words, we propose making use of parse trees of the sentence to be able to determine the context words for a target. For example: if the parse of a sentence of $S \Rightarrow NV$ and target is the head phrase for the V component, we would define the context vector from the head word of the N phrase.

Another enhancement which we would like to make to this model would be to test with compositional vector space so as to allow the model to suggest multiple words as replacement for the target.

6 Conclusion

In this paper, we have proposed a semantically context sensitive vector-space approach to find substitues for a word in a given context. The experiments provided us with several insights into the various aspects of distributional space.

An evaluation on the SemEval 2007 lexical substitution task data shows promising results. We outperform state-of-the-art unsupervised models in the *Best* evaluation metric. Also, our results are nearly same as the state-of-the-art supervised learning based model. However, our result for *OOT* metric are not as good as the state-of-the-art. This demonstrates the usefulness of capturing semantic relation between words while making use of distributional space for the reranking task.

There are many applications that either require or might benefit from systems to find a replacement word or phrase in context. Capabilities at the lexical substitution task would be useful for systems that recognize paraphrases, for example, question answering and textual entailment. Applications focusing on summarization or text simplification would also benefit from our work.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

Georgiana Dinu. 2013. Nghia the pham, and marco baroni. 2013a. dissect: Distributional semantics composition toolkit. *Proceedings of the System Demonstrations of ACL.*

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906. Association for Computational Linguistics.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 410–413. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 48–53. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. 2009. The english lexical substitution task. *Language resources and evaluation*, 43(2):139–159.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *ACL*, pages 236–244.

Samuel Reese, Gemma Boleda, Montse Cuadros, Lluís Padró, and German Rigau. 2010. Wikicorpus: A word-sense disambiguated multilingual wikipedia corpus.

Ravi Sinha and Rada Mihalcea. 2009. Combining lexical resources for contextual synonym expansion.

György Szarvas, Chris Biemann, and Iryna Gurevych. 2013. Supervised all-words lexical substitution using delexicalized features. In *Proceedings of NAACL-HLT*, pages 1131–1141.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 456.

Stefan Thater, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 44–47. Association for Computational Linguistics.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957. Association for Computational Linguistics.

J Watson. 2012. Big huge thesaurus. *John Watson, LLC*.

Deniz Yuret. 2007. Ku: Word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 207–213. Association for Computational Linguistics.