# Combination of Semantic Relatedness with Supervised Method for Word Sense Disambiguation

Qiaoli Zhou
*School of Computer*
*Shenyang Aerospace University*
*Shenyang, China*
*zhou_qiao_li@hotmail.com*

Yuguang Meng
*School of Computer*
*Shenyang Aerospace Universitye*
*Shenyang, China*
*465611370@qq.com*

***Abstract*—We present a semi-supervised learning method that efficiently exploits semantic relatedness in order to incorporate sense knowledge into a word sense disambiguation model and to leverage system performance. We have presented sense relativeness algorithms which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS-labeled text corpus, with sense-labeled data in the form of example sentences. This paper investigates the way of incorporating semantic relatedness in a word sense disambiguation setting and evaluates the method on some SensEval/SemEval lexical sample tasks. The obtained results show that such representations consistently improve the accuracy of the selective supervised WSD system.**

***Keywords*-Word Sense Disambiguation; Semantic Relatedness; semi-supervised learning; Neural Model;**

## I. Introduction

Word Sense Disambiguation (WSD) has been a hard nut ever since the earliest days of computer-based treatment of language in the 1950s. WSD is the task to identify the intended sense of a word in a computational manner based on the context in which it appears [1]. Many algorithms devote to WSD by exploiting two powerful properties of human language: “one sense per collocation” and “one sense per discourse” [2], [3]. In the “one sense per collocation”, the nearby words provide clues to the sense of the target word. “One sense per discourse” represents the sense that a target word is consistent with a given document. In the WSD research literature, currently, these two assumptions are widely accepted by natural language processing community. There are, however, several difficulties to WSD we need to face.

Firstly, one sense per collocation, the strong tendency for words which exhibits only one sense in a given collocation was observed and quantified in the paper [3]. Collocation refers to a group of practical words that habitually go together, whereas the sense of a word is figured out by accompanying words. In this case, words are to be classified in terms of co-occurrence relation as well as sense. The co-occurrence relation means the constraints shown in the sense combination relation, which is called collocation constraint or selection constraint. When we make use of these assumptions, it is easy to identify the sense of common expressions or idioms containing a target word. For example, the word “place” means general location. But, the meaning of the idiom “take place” is quite different from the meaning of “take her place”. The idiom ”take place” means that something occurs or happens at a particular time or place. Thus, an idiom is a group of words in a fixed order and has a particular meaning which is different from the meanings of the individual word regardless of the context of the word to be disambiguated. Although there are many researches aiming to solve WSD problem using the phrase in WordNet and idiom dictionary, when we take into consideration the overall occurrence in the target corpus, there still remain some cases where a dictionary may not cover some of the idioms that exist in the target corpus. Therefore, the effect of using collocations to resolve lexical ambiguities depends on the type of collocation. It is strongest for immediately adjacent collocations, and weakens with distance. It is much stronger for words in a predicate-argument relationship than for arbitrary associations at equivalent distance. It is very much stronger for collocations with content words than those with function words. Secondly, one sense per discourse, the sense of a target word is highly consistent within any given document. The observation that words strongly tend to exhibit only one sense in a given discourse or document was stated and quantified in Gale, Church and Yarowsky [2]. For example:

A The **stock** would be redeemed in five years, subject to terms of the company’s debt.

B Our soups are cooked with vegan **stock** and seasonal vegetables.

C In addition, they will receive **stock** in the reorganized company, which will be named Ranger Industries Inc.

Because the contexts between sentence A and sentence C are similar, we can identify that the “*stock*” in sentence 1 and in sentence C has the same meaning because they are both used in a “*company*” setting. Yet to date, the full power of this property has not been exploited for sense disambiguation. Many work derived from this assumption into statistical models based on local and topical features surrounding a target word to be disambiguated. However, even when we make use of these assumptions, it is difficult to identify the sense of common expressions or idioms containing the ambiguous term. For example:

(1) In an age when personal grievances is all the **rage**, this tale is uninstructive.
(2) In an age when **rage** powers personal grievances, this tale is uninstructive.

In these above two sentences, there are a word "*rage*" and an idiom "*all the rage*", as the same time, the context of two sentences are almost the same. If we want to identify the meaning of "*rage*" in these two sentences using similarity context, the result of identification must have the same meaning. According to the above mentioned research, we find supervised systems for WSD often rely upon word collocations (i.e., sense-specific keywords) to provide clues on the most likely sense for a word within the given context. Collocation makes these features more obvious, so supervised learning techniques have generally been found to perform more accurately than knowledge-based methods in "*one sense per collocation*". As for "*One sense per discourse*", WSD, a knowledge-based method could obtain more accuracy than other methods. Similarity-based methods determine the sense of a polysemous word (a word with more than one possible meaning) by computing the relatedness between each of its possible senses and the terms in the surrounding context. The correct sense of the ambiguous term is then assumed to be that for which the relatedness is the greatest.

Finally, we find the supervised methods which use an annotated training corpus inducing the appropriate classification models in terms of "*one sense per collocation*". We also find the relatedness-based method enables it to utilize a higher degree of semantic information, and is more consistent with the properties of "*One sense per discourse*"; that is, by considering the greater context in which the word appears. Because relatedness-based disambiguate all words in a text fragment simultaneously by exploiting semantic relatedness across word senses, it usually achieves higher performance than their supervised alternatives which usually do not considering the senses assigned to surrounding words. To overcome above problems and combine advantages, we propose a hybrid approach for WSD, which combines supervised and relatedness-based methods. This is achieved by combining supervised method and semantic relatedness measures. This approach integrates a diverse set of knowledge sources to disambiguate word sense, including Part-Of-Speech(POS), labeled training data, corpora of unlabeled data, salient neighboring words, and glosses of ambiguous words.

The rest of this paper is organized as follows. Section 2 presents a short review of some earlier works. Section3 refer to Neural Language Models we have used for our study and which calculate semantic relatedness. Section 4 explains the approach used in this paper and presents the results and corresponding explanations. A discussion of the experimental results is given in Section5. Finally, section 6 concludes the proposed method.

## II. Previous Work

For supervised based methods, studies have shown that the word, the word n-gram, the traditional orthographic features, and the POS are the bases for WSD, but they are poor at representing semantic background. In order to incorporate semantic knowledge into an ML model, Semi-Supervised Learning (SSL) techniques have been applied to WSD. SSL is an Machine Learning (ML) approach that typically uses corpora of unlabeled data and a small amount of labeled data to build a more accurate classification model than would be built using only labeled data. SSL has received significant attention for two reasons. First, preparing a large amount of data for training requires a lot of time and effort. Second, since SSL exploits unlabeled data, the accuracy of classifiers is generally improved. There have been two different directions in SSL methods: 1) semi-supervised learning approaches, which are randomly select a subset of a large unlabeled dataset and classify these samples using one (self-training) or two (co-training) classifiers, trained on a smaller set of labeled samples. After assigning labels to the new samples, these methods select the samples that were classified with a high confidence (according to a selection criterion) and add them to the set of labeled data, and 2) supervised model induction with un-supervised, possibly unlabeled data, feature learning [4]. Yuan [5] also present an algorithm for semisuper-vised learning, using label propagation to label unlabeled sentences based on their similarity to labeled ones. The approaches in the second research direction induce better feature representation by learning from unlabeled data.

This study extends our previous work in the following ways. First, we propose a hybrid approach for WSD, which combines supervised methods and semantic relatedness. Second, we have presented context2vector which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS labeled text corpus. Third, we explore semantic relatedness algorithms and approach of relatedness feature representation. Finally, we take a small amount of labeled data to build supervised models to avoid being entrapped in the problem of creating annotated corpora. These changes lead to outperforming the results obtained by other systems in the considered competitions.

## III. Neural Language Models-based Semantic Relatedness

Melamud et al. argues that since contexts induce meanings (or senses) for target words, a good context similarity measure should assign high similarity values to contexts that induce similar senses for the same target word [6]. More recently, in an unsupervised setting, word embeddings were used in measuring context-sensitive similarity to learn internal representations of wider sentential contexts [7]. Therefore neural language models were used to measuring context-sensitive similarity.

### A. Context2vec's Neural Model

Like target words, contexts are commonly represented via word embeddings. In an unsupervised setting, such representations were found useful for measuring context-sensitive similarity , word sense disambiguation [7], [8]. The context representations used in such tasks are commonly just a simple collection of the individual embeddings of the neighboring words in a window around the target word, or an (sometimes weighted) average of these embeddings. We note that such approaches do not include any mechanism for optimizing the representation of the entire sentential context as a whole. In this work , we present context2vec-based semantic relatedness of method, an unsupervised model for efficiently learning generic context embedding of wide sentential contexts, using bidirectional LSTM. The main goal of our model is to learn a generic task-independent embedding function for variable-length sentential contexts around target words. To do this, we use a neural network architecture, which is based on word2vec's CBOW architecture, but replaces its naive context modeling of averaged word embeddings in a fixed window, with a much more powerful neural model, using bidirectional LSTM [9]. We use a bidirectional LSTM recurrent neural network to obtain a sentence-level context representation [10]. However, that context representation has a drawback which has not been regarded for POS in input level. Though two words have different POS, they have the same word embeddings. For addressing this problem, in this paper, we proposed an improvement method which adds POS into input level to construct context2vec's neural model. However, we don't need fine-grained POS. We have combined some of the tags. In this paper, we tag five kinds of POS only as shown in table 1 right column.

Our proposed architecture is illustrated in Figure 1. This model learn context and target word representations at the same time, by embedding them into the same low-dimensional space, with the objective of having the context predict the target word via a log linear model. We utilize a much more powerful parametric model to capture the essence of sentential context. Figure 1 illustrates how context2vec represents sentential context. We use a bidirectional LSTM recurrent neural network, feeding one LSTM network with the sentence words from left to right, and another from right to left. The parameters of these two networks are completely separate, including two separate sets of left-to-right and right-to-left context word embeddings. To represent the context of an ambiguous term in a sentence (e.g. for "John/NP [submitted/VV] a paper/NN"), we first concatenate the LSTM output vector representing its left-to-right context ("John/NP") with the one representing its right-to-left context ("a paper/NN").

Let lLS be an LSTM reading the words of a given sentence from left to right, and let rLS be a reverse one reading the words from right to left. Given a sentence $w_{1:n}$ , our 'shallow' bidirectional LSTM context representation for the target $w_i$is defined as the following vector concatenation:

$$biLS(w_{1:n}, i) = lLS(l_{1:i-1}) + rLS(r_{n:i+1}) \quad (1)$$

where $l/r$ represent distinct left-to-right/right-to-left word embeddings of the sentence words.This definition is a bit different than standard bidirectional LSTM, as we do not feed the LSTMs with the target word itself (i.e. the word in position $i$). With this, we aim to capture the relevant information in the sentential context, even when it is remote from the target word. Next, we feed this concatenated vector into a multi-layer perceptron to be capable of representing non-trivial dependencies between the two sides of the context.

$$MLP(x) = L_2 S(ReLU(L_1(x))) \quad (2)$$

where MLP stands for Multi Layer Perceptron, ReLU is the Rectified Linear Unit activation function, and $L_i(x) = W_i x + b_i$ is a fully connected linear operation. Let$c = w_1, \ldots w_{i-1}, -w_{i+1}, \ldots, w_n$be the sentential context of the word in position i. We define context2vec's representation of $c$ as:

$$c = MLP(biLS(w_{1:n}, i)) \quad (3)$$

We consider the output of this layer as the embedding of the entire joint sentential context around the target word. At the same time, the target word itself (right-hand side of Figure 2) is represented with its own embedding, equal in dimensionality to that of the sentential context. We note that the only (yet crucial) difference between our model and word2vec's CBOW (Figure 1) is that CBOW represents the context around a target word as a simple average of the embeddings of the context words in a window around it, while context2vec utilizes a full-sentence neural representation of context.

### *B. Context2vec-Based Semantic Relatedness*

After obtaining context2vec, we need to obtain sematic similarity. We compute semantic similarity metrics in that space of context-to-context. All these are measured by the vector cosine value between the respective embedding representations. To classify a test word instance in context, we consider all of the tagged instances of the same word lemma in the training set, and find the instance whose context embedding is the most similar to the context embedding of the test instance using the context-to-context similarity metric. Then, we use the tagged senses of that instance. We note that this is essentially the simplest form of a k-nearest-neighbor algorithm, with $k = 4$.

$$\arg\max_{1 \le i \le k} \mathrm{sim}(c_0, c_i) \quad (4)$$

### *C. Similarity-Based Feature Representation*

Feature representation is very important to supervised methods. Therefore, semantic similarity was transformed into a special feature in this paper. To get context vector, some conditions are to be fulfilled in each of the experiments. These conditions are summarized as follows:

(1) POS of the disambiguation sentence and instances.
(2) Containing the POS of notional words and deleting the POS of function words.
(3) POS of notional words mapping based on the table 1.

Disambiguation sentence: *Sodalities have an important role in **activating** laity for what are judged to be religious goals both personally and socially* . Above sentence after the POS tagging: *Sodalities_NN have an important_JJ role_NN in **activating_VV** laity_NN for what are*

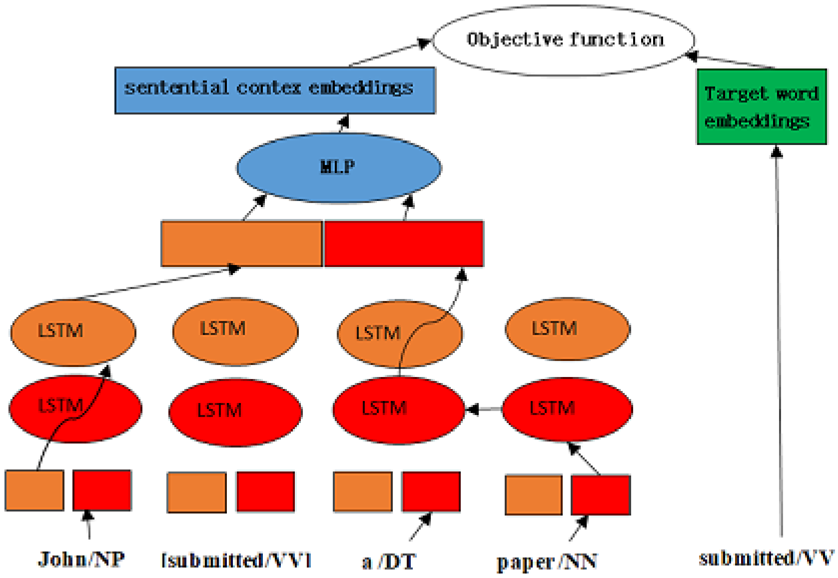


Figure 1. context2vec with POS

Table I
POS MAPPING LIST

| | |
|---|---|
| RB,RBR,RBS | RB |
| NNS,NN | NN |
| NP,NPS | NN |
| JJS,JJR,JJ | JJ |
| VV,VVD,VVG,VVN,VVP,VVZ | VV |

*judged_VV to be religious_JJ goals_NN both personally_RB and socially_RB* . We implemented the semantic disambiguation as a centroid-based classifier, which, given a disambiguation sentence, ranks all the instances by their relevance to the sentence. Given a disambiguation sentence, we first represent it as a vector using contex2vec. To compute semantic relatedness of a pair of sentences we compare their vectors using the formulation 4. In table 2 the second column are instances tagged sense, the third column are senses id and the forth column are weight of every sense. If some word correspond to two senses, the weight would signed 0.5. We measure senses of disambiguation word in all instances based on formulation 5.

$$Score(S_i) = T * weight \tag{5}$$

Where $S_i$is one of sense of the disambiguation word, $T$ is the number of times which sense appeared in the top 4 instances, and weight is the same as weight in the table 2. For example,“ *active*” totally has five senses. Based on the formulation 5 the score of every sense as follow:

- $Score(S_1) = 1.5$
- $Score(S_2) = 2.0$
- $Score(S_3) = 0.5$
- $Score(S_4) = 0.0$
- $Score(S_5) = 0.0$

We take the score as sense feature in the supervised learning. As our supervised WSD dataset, we used the Senseval-3 lexical sample dataset, denoted SE-3, which includes 7,860 train and 3,944 test instances. We used the training set for instance corpus and report accuracy results on the test set [11].

## IV. METHODS

The machine learning tool that the supervised system used for word sense disambiguation is Conditional Random Fields (CRFs). We extract types of features and then use CRFs as the classifier. The features and templates implemented in our system are explained below section 4.2. After extracting these features, the classifier (CRFs) is used to train a model for the same number sense of target word. In the test phase, the model is used to classify test samples and to assign a sense tag to each sample.

### A. Preprocessing

We used the two billion word ukWaC as our learning corpus [12]. We use TreeTagger to tag POS [13] . To speed-up the training of context2vec, we discarded all sentences that are longer than 64 words,reducing the size of the corpus by 10%.We lower-cased all text and considered any token with fewer than 100 occurrences as an unknown word. This yielded a vocabulary of a little over 180K words for the full corpus, and 160K words for the trimmed version.

### B. Feature and Template

The word representation feature is essential to classifier, but it is poor because it only carries some morphological and shallow-syntax information of words. However, the sense representation features can be extracted by calculating semantic relatedness and may be capable of introducing sense knowledge background to the WSD model. We take the score of sense as sense feature in section 3.2, and then an ML algorithm is employed to build a model for WSD. We applied features and templates in ML algorithm as follows:In table3,$t$ represents words in

Table II
FIRST 4 INSTANCES IN INSTANCE CORPUS

| # | Input: disambiguation sentence | Sense id | weight |
|---|---|---|---|
| 1 | You step_VV on to **activate_VV** it . | 38201 | 1.0 |
| 2 | Which parts_NN of the sensory_JJ system_NN are **activated_VV**. | 38202 | 1.0 |
| 3 | Different_JJ genes_NN are **activated_VV** in different_JJ cells_NN . | 38203 | 1.0 |
| 4 | This clause_NN has never_RB yet_RB been **activated_VV** . | 38201 38203 | 0.5 |

Table III
FEATURES AND TEMPLES OF CRFS TRAINING

| feature | temple |
|---|---|
| t | $t_0$ |
| $score$ | $score(s_j)(1 \le j \le m)$ |
| $t + score$ | $t_0 score(s_j)$ |
| POS | $p_0$ |
| others | $t_{-4}t_{-3}t_{-2}t_{-1}t_0t_1t_2t_3t_4$ , $p_{-4}p_{-3}p_{-2}p_{-1}p_0p_1p_2p_3p_4$ |

Table IV
TAGGING FORMAT OF CRFS TRAINING SET

| term | POS | F1 | F2 | F3 | F4 | Fi | tag |
|---|---|---|---|---|---|---|---|
| You | PP | 0 | 0 | 0 | 0 | 0 | O |
| step | VV | 0 | 0 | 0 | 0 | 0 | O |
| on | IN | 0 | 0 | 0 | 0 | 0 | O |
| to | RB | 0 | 0 | 0 | 0 | 0 | O |
| activate | VV | 1.5 | 2.0 | 0.5 | 0.0 | 0.0 | F1 |
| it | PP | 0 | 0 | 0 | 0 | 0 | O |
| . | . | 0 | 0 | 0 | 0 | 0 | O |

a given sentence $W$, $score$ represents semantic similarity score(details see section 3.3), $t{+}score$ represents the combination feature, $t_0$represents the disambiguation word, $t_{-1}$ is the word preceding $t_0$, $t_1$ is the word following $t_0$, $score(s_j)$represents one sense of $t_0$, $p$represents POS of word.We compute semantic similarity metrics in the space of context-to-context. All these are measured by the vector cosine value between the respective embedding representations.

### C. Semi-Supervised Learning

We apply the CRFs tools to build WSD model. In this paper, tagging format of CRFs training is shown in table4. In table4, F1, F2, F3 and F4 denote the feature of every sense based on the score . Fi denotes $i^{th}$ score of sense.

### D. Experiments Results

In this section, we study the performance of our classifiers on SE3 lexical sample task, SemEval-2013 Task 12 and SemEval-2015 task 13 [14], [15]. For experiment results, we report F1 score (Navigli, 2009). In table 5, we refer to Context2vec based sense relatedness as 'CBSR'. In Table 5, we compared our overall F1 scores with different systems which include An-do [16], Rothe [17], Melamud [10], Grozea [18]. The best performance is achieved when we combine supervised learning (CRFs) and CBSR. Ando achieved high performance using Alternating Structure Optimization that is a very complicated method compared to ours. Rothe has a system to learn embeddings for synsets and lexemes. Mel-maud learn context embeddings of a word and classify a test word instance with the sense of the training set word whose context embedding is the most similar to the context embedding of the test instance. The method of Mel-maud is similar to CBSR, but our method adds the POS feature into the context embedding. Grozea is the best system in Senseval-3. As can be observed from Table 6, we did see significant improvements using CBSR with the CRFs model. In Table 6, the improvements obtained by using Semcor with OMSTI as training data over Semcor only are significant.

Table V
COMPARISON OF OUR SYSTEM WITH OTHER SYSTEMS ON SENSEVAL-3 LEXICAL

| System | Micro-average |
|---|---|
| CRFs (baseline) | 62.0 |
| CBSR | 75.3 |
| CRFs+CBSR | **75.9** |
| Ando | 74.1 |
| Rothe | 73.6 |
| Mel-amud | 72.8 |
| Grozea | 72.9 |

## V. DISCUSSION

Supervised word sense disambiguation systems usually treat words as discrete entities and consequently ignore the concept of relatedness between words. However, by adding sense relatedness, some of the samples that cannot be discriminated basing on the original features (surrounding words, long distance dependency) have more chances to be classified correctly. Moreover, sense relatedness contain valuable linguistic information too. Hence, adding representations of sense can provide valuable information to the classifier and the classifier can learn better discriminative criteria based on such information. Our approach to WSD does not rely on large labeled data sets. Instead, it leans on supervised models learned from small labelled data sets, on representations of sense relatedness learned from structured semantic resources and on medium unlabeled corpora. This enables us to exploit how to integrate the semantic knowledge of word in the framework of WSD.

## VI. CONCLUSION

We have presented sense relativeness algorithms which combine neural model learned from a generic embedding function for variable length contexts of target words on a POS labeled text corpus, with sense-labeled data in the form of example sentences. Meanwhile, corpora resource provides collocation occurrences of a target word that

Table VI
EVALUATION RESULTS OF DIFFERENT RUNS WITH VARIED APPROACH ON ALL-WORDS DATASETS OF SEMEVAL-2013 AND SEMEVAL-2015

| Method | SemEval-13 Semcor | SemEval-13 Semcor+OMSTI | SemEval-15 Semcor | SemEval-15 Semcor+OMSTI |
|---|---|---|---|---|
| CRFs (baseline) | 46.7 | 52.3 | 50.1 | 58.7 |
| CBSR | 64.4 | 65.8 | 69.5 | 70.3 |
| CRFs+CBSR | 64.5 | 66.0 | 69.9 | 70.6 |

cannot be gained from sense inventory resources and context embedding. Semi-supervised WSD systems generate high quality WSD result using CBSR, as it may carry more useful information learned from large corpora than other methods. Our system that solves WSD would enable corpora resource to take full advantage of their enlightened decision to incorporate sense knowledge into supervised learning method.

## ACKNOWLEDGMENT

The authors would like to thank the Humanities and Social Sciences of Ministry of Education Planning Fund (18YJA870020) for their support in this research.

## REFERENCES

[1] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–69, Feb. 2009.

[2] W. A. Gale, K. W. Church, and D. Yarowsky, "One sense per discourse," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. Association for Computational Linguis-tics, 1992.

[3] D. Yarowsky, "One sense per collocation," in *Proceedings of the Workshop on Human Language Technology*, ser. HLT '93. Association for Computational Linguistics, 1993.

[4] K. Taghipour and H. T. Ng, "Semi-supervised word sense disambiguation using word embeddings in general and specific domains," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2010.

[5] D. Yuan, R. Doherty, J. Richardson, C. Evans, and E. Altendorf. (2016) Word sense disambiguation with neural language models.

[6] O. Melamud, I. Dagan, and J. Goldberger, "Modeling word meaning in context with substitute vectors," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015.

[7] E. Huang, R. Socher, C. Manning, and A. Ng, "Improving word representations via global context and multiple word prototypes," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012.

[8] X. Chen, Z. Liu, and M. Sun, "A unified model for word sense representation and disambiguation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013) Efficient estimation of word representations in vector space.

[10] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning generic context embedding with bidirectional lstm," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2016.

[11] R. Mihalcea, T. Chklovski, and A. Kilgarriff, "The senseval-3 english lexical sample task," in *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, 2004.

[12] A. Ferraresi, E. Zanchetta, M. Baroni, and S. Bernardini, "Introducing and evaluating ukwac, a very large web-derived corpus of english," in *In Proceedings of the 4th Web as Corpus Workshop (WAC-4)*, 2008.

[13] TreeTagger, "Treetagger - a part-of-speech tagger for many languages."

[14] R. Navigli, D. Jurgens, and D. Vannella, "Semeval-2013 task 12: Multilingual word sense disambiguation," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, 2013.

[15] A. Moro and R. Navigli, "Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 2015.

[16] R. K. Ando, "Applying alternating structure optimization to word sense disambiguation," in *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*. Association for Computational Linguistics, 2006.

[17] S. Rothe and H. Schütze, "Autoextend: Extending word embeddings to embeddings for synsets and lexemes," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, 2015.

[18] C. Grozea, "Finding optimal parameter settings for high performance word sense disambiguation," in *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*. Association for Computational Linguistics, 2004.