Utilizing Semantic Composition in Distributional Semantic Models for Word Sense Discrimination and Word Sense Disambiguation

Cem Akkaya
University of Pittsburgh
Email: cem@cs.pitt.edu

Janyce Wiebe
University of Pittsburgh
Email: wiebe@cs.pitt.edu

Rada Mihalcea
University of North Texas
Email: rada@cs.unt.edu

***Abstract*—Semantic composition in distributional semantic models (DSMs) offers a powerful tool to represent word meaning in context. In this paper, we investigate methods to utilize compositional DSMs to improve *word sense discrimination* and *word sense disambiguation*. In this work, we rely on a previously proposed multiplicative model of composition. We explore methods to extend this model to exploit richer contexts. For word sense discrimination, we build context vectors, which are clustered, from the word representations based on the extended compositional model. For word sense disambiguation, we augment lexical features with their word representations based on the same extended compositional model. For both tasks, we achieve substantial improvement.**

I. Introduction

Distributional semantic models (DSMs) [1], [2], [3] provide a means for representing word meaning. They are based on the assumption that the meaning of a word can be inferred from its distribution in text and that words appearing in similar contexts tend to have similar meanings [4]. As a tool to represent word meaning, DSMs have been successfully applied to many NLP tasks. Some examples are word sense discrimination [5], paraphrases [6], thesaurus compilation [7] and language tests [8].

A DSM is a co-occurrence matrix such that each row represents the distribution of a target word across contexts in a large corpus. The context can be defined in many ways. It can be a document, a sentence or a word window around the target word. A DSM allows a type-based representation. This means that the rows of the co-occurrence matrix represent word types rather than word tokens. Thus, all contexts and senses of a target word are accumulated into one vector. There have been recent approaches addressing this issue. [9], [10], [11] develop specialized models for semantic composition that allow them to derive representations of word meaning in context. These compositional models are built on top of conventional DSM. The meaning of a target word in a specific context (i.e. word token) is computed through vector operations applied to the co-occurrence vector of the target word and the co-occurrence vectors of the words in the specific context. The result is a new co-occurrence vector that holds the meaning of the target word token. These models demonstrate so far promise for paraphrase ranking and phrase similarity rating tasks.

In this paper, we propose to utilize these compositional models for *word sense discrimination* and *word sense disambiguation (WSD)*. To be specific, we investigate using element-wise multiplication model introduced in [10], since this model has been shown to perform consistently well without the need for parameter tuning. This compositional model represents word meaning in context in a reliable way. Our hypothesis is that we can make use of this token-based meaning representation to improve word sense discrimination and word sense disambiguation. Unfortunately, [10] and others define their model to compute the meaning of a target word in specific dependency relations (e.g. verb-object). This treatment is restricting if we want to use this model for representing the meaning of a target word token in arbitrarily long context where the target word token can be related to various context words via various dependency paths. Thus, we investigate methods to extend the underlying model and apply it to arbitrary target words in arbitrary contexts. We apply the extended model to word sense discrimination by clustering target word instances according the semantic vectors derived from the extended compositional model. We compare this approach to two previous approaches based on work in [5], [12]. Then, we apply the extended model to supervised word sense disambiguation by feature expansion, making it semi-supervised. That is, we expand the lexical features of a supervised word sense disambiguation system using their co-occurrence vectors derived from the extended compositional model. We compare this approach to several other word representations including brown clusters [13], which is an effective word representation for feature expansion [14]. Our experiments show that the extended model can be effectively utilized for word sense discrimination and word sense disambiguation (WSD) outperforming previous approaches.

The rest of the paper is structured as follows. In section II, we give a short introduction to DSM. In section III, we explain element-wise multiplication model introduced in [10]. We illustrate the obstacles we face when we want to apply this compositional model in complex contexts. In section IV, we introduce our efforts to extend the model. In section V, we introduce the target tasks and we explain how we apply our extended compositional model to them. In Section VI, we conduct experiments to evaluate our approach and compare

	computer	cheese	button	cat
$\overrightarrow{mouse}$	22	8	17	13
$\overrightarrow{click}$	23	0	18	0
$\overrightarrow{catch}$	0	2	0	11

TABLE I
A HYPOTHETICAL WORD-WORD CO-OCCURRENCE MATRIX

	computer	cheese	button	cat
$click + mouse$	45	8	35	13
$catch + mouse$	22	10	17	13
$click \cdot mouse$	506	0	306	0
$catch \cdot mouse$	0	16	0	143

TABLE II
ADDITIVE AND MULTIPLICATIVE COMPOSITION OF CO-OCCURRENCE VECTORS

to previous work. In Section VII, we review related work. In Section VIII, we draw conclusions.

II. DISTRIBUTIONAL SEMANTIC MODELS

Distributional semantic models (DSM) are based on the assumption that the meaning of a word can be inferred from its distribution in text. A DSM is basically a co-occurrence matrix – also called *semantic space* – such that each row vector represents the distribution of a target word across contexts. The context can be a document, a sentence, or a word window around the target word. In this paper, we focus on the latter one. In that setting, a DSM is a word-word co-occurrence matrix. The dimensions of the vector represent co-occurring context words and hold some score based on the occurrence frequency of the context word near the target word in the specified window. This co-occurrence vector builds the semantic signature of the target word. Basically, each target word is described in terms of co-occurring words in its textual proximity. Table I represents a hypothetical word-word co-occurrence matrix for the words "mouse", "click" and "catch". The dimensions of the semantic space are "computer", "cheese", "button" and "cat". In this example, the matrix holds simple co-occurrence frequencies but it can be defined to have an association score between the target and context words such as point-wise mutual information. Note that DSMs model meanings of words out of context (i.e., of word types).

III. COMPOSITIONAL DSMS

Compositional DSMs [9], [10], [11] offer a powerful tool to represent words in context. They build on top of the conventional DSMs introduced in section II. The meaning of a word in context (i.e., word token) is computed through composition operations applied to the target word and its context. [10] evaluate a good amount of composition operations. Vector summation and element-wise vector multiplication are two sample composition operations from [10]. To illustrate, Table II gives compositional vectors for the word "mouse" in the context of "click" and "catch" for both operations. *click+mouse* is computed by summing co-occurrence vectors of "click" and "mouse" from the semantic space in Table I. *click·mouse* is computed by element-wise multiplication of the co-occurrence vectors of "click" and "mouse" from the same semantic space. The same is true for *catch+mouse* and *catch·mouse*. Note that the vectors in Table I and in Table II have the same dimensions. The difference is that the semantic space in TableI is type-based, where the semantic space in Table II is token-based. In the token-based semantic space, "mouse" will have a different semantic vector depending on its context. From now on, we will refer to the co-occurrence vectors in the type-based semantic space as *type vectors* and the co-occurrence vectors in the token-based semantic space as *token vectors*.

It is appealing that the multiplicative model allows one vector to pick out the relevant content of the other. Indeed, [10] show that element-wise multiplication performs overall better than vector addition and other composition operations on a phrase similarity task without the need of parameter tuning. Thus, in our work, we rely on element-wise multiplication to derive contextual meaning of words.

[10] consider composition only between specific word pairs, namely word pairs related by a verb-object relation. In a later paper, [15] also consider noun-noun and noun-adjective relations. This treatment is restricting if we want to use this model for representing the meaning of a target word token in arbitrarily long contexts where the target word token can be related to various context words via various dependency paths. Consider the following example in Figure 1. Considering only the verb-object relation and taking *begin·strike* as the compositional meaning of "strike" is not sufficient in this context. The words that are most informative for disambiguating "strike" are "workers" and "mines". "workers" is related to strike over a "nsubj←|dobj→" dependency path and "mines" is connected to "strike" over a "prep–at←| nsubj←|dobj→" dependency path. That simple example shows that we need to utilize longer dependency paths to reach informative and discriminative context words. Thus, we propose to extend the model proposed in [10] to include longer arbitrary dependency paths.

A simple strategy would be to utilize all possible context words connected to the target word through a dependency path to compute the compositional representation of the target word. But, that might introduce too much noise, since we can reach every word in the sentence if we fully traverse the dependency tree. Thus, we propose to investigate methods to filter out uninformative dependency paths (i.e. context words).

IV. EXPLOITING RICHER CONTEXTS

In this section, we introduce the methods we use to choose informative context words of a target word to incorporate into the compositional representation of that target word.

From now on, we will refer to context words that are related to the target word over a dependency path as *context clues*. The most important question is how to filter out the uninformative context clues. We try four different methods

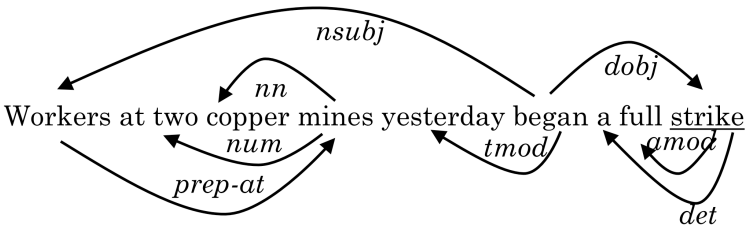

Fig. 1. Example for compositional representation

for this purpose. The first two of them are simple in nature. They define constraints on the type of the context clue :

- *content* : the context clue should be a content word (i.e., noun, verb, adjective, or adverb).
- *nostop* : the content clue cannot be a stop word.

The next two are more elaborate filtering mechanisms. We define two scoring functions to assign an importance score to each context clue. Our intuition is that context clues which carry more information to disambiguate the sense of the target word token should get a higher score and be chosen to contribute to the compositional representation of the target word (e.g. "workers" and "mines" in the figure 1 rather than "began").

The first scoring function keeps track of the change of the type vector of the target word after applying the type vector of the context clue to it. The hypothesis is that a context clue which selects out a specific sense of the target word will zero out a substantial amount of dimensions of the type vector of the target word. I.e., the more dimensions the context clue zeros out, the better the disambiguation should be. We count the dimensions of the type vector of the target word which become zero after applying the context clue. In order to avoid very infrequent context words getting high scores (since they will have lots of zero dimensions), we put a normalizing factor, the number of zero dimensions of the context clue. The scoring function, $maxzero$, is as follows: ($zero$ is a function which returns the number of zero dimensions in a vector).

$$maxzero = \frac{zero(target) - zero(target \cdot clue)}{zero(clue)}$$

The second scoring function takes into account distributional substitutes of the target word based on the dependency path and the context clue. By distributional substitutes, we mean the set of words which are connected to the context clue via the same dependency relation in our corpus (described in section VI). Our hypothesis is that the substitute sets can provide us useful information about the discriminative power of the corresponding context clues. If one of the substitutes related to a context clue is similar to one of the senses of the target word more strongly than other senses, we can conclude that the context clue is discriminative and should be scored high. For this purpose, we make use of WordNet similarity measure introduced by [16]. First we assume that each sense of a target word is equally probable and find the similarity of a substitute to different senses of the target word. Then, we normalize the similarity score over the senses and obtain a probability distribution over the senses of the target word. We apply the Kullback Leibler (KL) divergence to determine how much the new distribution differs from the uniform distribution. The context clues with substitution sets which have high maximum KL divergence scores are also scored high. Below is the formula for the $discsubs$ scoring function. $subs$ is a function which returns the set of distributional substitutes of a target word in context of a dependency path and context word. $disc$ is a function which computes the KL divergence value as described above.

$$discsubs = \max_{s \in subs(target,clue,path)} disc(s, target)$$

To illustrate, in Figure 2, we see examples of distributional substitutes of strike for two context clues "workers" and "begin". One of the substitutes derived from the context clue "workers" is "protest", which is strongly related to the "work stoppage" meaning of "strike" in WordNet. On the other hand, substitutes derived from the context clue "began" (e.g. "eating") are more general in nature and do not favour a specific sense of "strike" in WordNet. As a consequence, "workers" will get a higher score than "began", which is exactly what we want.

These two scoring functions give us two filtering mechanisms where we accept the best scoring context clues. We can choose more than one context clue to apply to the target word. By "applying" we mean element-wise multiplication of the type vector of the context clue with the type vector of the target word. We apply each chosen context clue separately to the target word resulting in multiple token vectors for the target word. We average these token vectors to obtain an ultimate single token vector of the target word. Our intuition is that each context clue chooses out some relevant dimensions of the target word and by averaging them, we smooth the contribution of the various context clues to create the final representation.

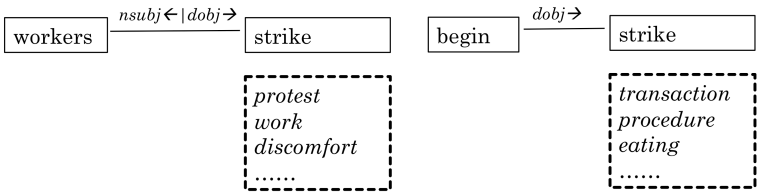

Fig. 2. Example for distributional substitutes

V. Applications

A. Word Sense Discrimination

Word sense discrimination is the automatic identification of the senses of a target word. The goal of word sense discrimination is to cluster target word instances, so that the induced clusters contain instances used with the same sense. Word sense discrimination accepts as input a set of target word instances represented as feature vectors – also called *context vectors* – which are then clustered based on the similarity of their feature vectors. The input is similar to the input of supervised word sense disambiguation (WSD) systems. The main difference is that the sense labels are missing.

[5] and [12] are two prominent works in this field. The biggest difference between them is the representation of the feature vectors. [5] relies on a DSM based on a word-word co-occurrence matrix to create feature vectors. Recall that a distributional semantic model represents word types, not word tokens. For a token-based treatment, [5] utilizes a second-order representation. That is, [5] represents each target word token by averaging type vectors of the neighbouring words that occur in its context. In contrast, (Purandare and Pedersen, 2004) creates feature vectors from local feature representations similar to the feature vectors common to supervised WSD.

[5] is basically using an additive model for compositional representation. Our approach is similar to [5], except that, instead of averaging type vectors, we average the token vectors of the neighbouring words, which we compute with our extended compositional model. Below in Section VI, we compare our approach to using second-order representation relying on type vectors as in [5].

Additionally, we compare our approach to one based on the local feature representation as described in [12]. We use the following features from [17] to build the local feature representation:

CW : the target word itself
CP : POS of the target word
CF : surrounding context of 3 words and their POS
HNP : the head of the noun phrase to which the target word belongs
NB : the first noun before the target word
NA : the first noun after the target word
VB : the first verb before the target word
VA : the first verb after the target word
SK : at most 10 context words occurring at least 5 times; determined for each sense.

B. Feature Expansion for Semi-supervised WSD

Second, we apply our compositional representation to supervised WSD. Supervised WSD, as many other supervised NLP applications, suffers from data sparsity. Consider following WSD example, where the learner has seen two instances of the target word fine and needs to disambiguate a third instance. For simplicity, we assume that the classifier utilizes only two lexical features: one word before and one word after the target word we want to disambiguate.

.. a $\text{fine}_{\text{sense\_a}}$ distinction .. $\rightarrow$ a, distinction, *sense_a*
.. a $\text{fine}_{\text{sense\_b}}$ film of oil .. $\rightarrow$ a, film, *sense_b*
.. a $\text{fine}_{?}$ layer of chocolate .. $\rightarrow$ a, layer, ?

According to WordNet, the first instance has the meaning of minutely precise especially in differences in meaning and the second instance has the meaning of thin in thickness or diameter. Unfortunately there is not enough evidence for the learner in the seen instances to classify the third example as sense *b*. It is easy to classify for a human the third example as sense *b*, since we know that "film" and "layer" are synonyms and can substitute one another in the given context. The information is implicitly there but the learner has no means to make use of it.

Researchers previously investigated methods to induce word representations from unlabeled large corpora and augment lexical features with their corresponding word representations. The aim is to incorporate shared semantic content between word features without changing the underlying learning algorithm or the training method as in other semi-supervised methods. This can be thought as a form of feature smoothing. These word representations are created without any labeled data in a general way. They can be used as plug-and-play modules in any system relying on lexical features to handle data sparsity.

There are many ways to build these word representations. Researchers used word clustering [18], [19], [20], [21], word embeddings via neural language models [14] and distributional semantic models [2] to augment features in supervised NLP systems, thereby making them semi-supervised. Most of these approaches except [21] use type-based word representations. This means these representations are compiled as static lists or vectors. Thus, they do not take into account that the same

	nofilter	content	nostop	maxzero				discsubs			
				1	2	3	weighted	1	2	3	weighted
L1	45.86	44.5	48.72	45.87	45.03	45.21	45.64	49.48	48.39	48.98	49.01
L2	44.78	45.05	48.23	49.18	46.34	45.72	45.57	49.48	48.73	48.54	48.76
L3	45.91	46.29	48.4	49.86	47.9	46.58	45.66	49.52	48.09	47.36	48.65
L4	45.87	44.9	48.1	**50.34**	49.22	46.88	46.82	48.54	48.5	48.24	48.80

TABLE III
EFFECT OF THE VARIOUS DEPENDENCY PATH LENGTHS AND FILTERING TECHNIQUES USED TO COMPUTE THE CONTEXTUAL REPRESENTATION ON THE CLUSTERING PERFORMANCE

word can have different meanings depending on its context and should have different representations depending on it.

We use our compositional representation to augment a vanilla WSD system. This is a token-based feature augmentation approach, where each word can have different representations depending on its context, as opposed to a type-based augmentation. The vanilla WSD system we augment uses the same set of features described in Section V-A. We augment all lexical features (i.e. CW and CF features) with their compositional representations.

Below in Section VI, we make various comparisons between representation obtained from our extended compositional model and other word representations for feature expansion. To be specific, we compare token vectors obtained from our extended compositional model to brown clusters, which has been shown to work well in NLP [14], to the additive context representation we see in [5] and to simple type vectors from our semantic space.

VI. EXPERIMENTS

In this section, we evaluate the application of our extended compositional model to both word sense discrimination and word sense disambiguation. For both tasks, we use the union of the SENSEVAL II [22], and the SENSEVAL III [23] datasets as our evaluation data.

A. Semantic Space

In this paper, all approaches making use of a DSM – including our extended compositional model – use the same semantic space. The semantic space we use in our experiments is built from a text corpus consisting of 120 billion tokens. We compile the corpus from various resources in order to have a balanced corpus. The corpus consists of news articles from GIGAWORD and editorials from NewYorker, NewYork Times, Slate, Townhall, BBC and Guardian. It also consists Open American National Corpus.

The rows of our semantic space correspond to word forms and the columns of the semantic space correspond to word lemmas present in the corpus. We adopt a window size of 10. We keep 2000 dimensions (i.e., 2000 most frequent lemmas) and do not filter out stop words, since they have been shown to be useful for various semantic similarity tasks in [3]. We use positive point-wise mutual information to compute values of the vector components, which has been shown to be favourable in [3].

B. Word Sense Discrimination

As described in Section V, our approach is very similar to [5]. [5] represents each target word token by averaging type vectors of the neighbouring words that occur in its context. In our approach, instead of averaging type vectors, we average the token vectors of the neighbouring words, which we compute with our extended compositional model. Our first goal is to measure the effect of longer dependency paths and different filtering strategies. For this purpose, we cluster all the evaluation data with hierarchical clustering – average linkage clustering. For clustering, we require 7 clusters as it is done in [12]. We choose cluster purity as our evaluation metric. To compute cluster purity, we assign each cluster to a sense label, which is the most frequent one in the cluster. The number of the correctly assigned instances divided by the number of all the clustered instances gives us cluster purity. Following [12], we assign each sense label to at most one cluster so that the assignment leads to a maximally accurate mapping of senses to clusters.

In our experiments we consider dependency paths of length up to four. For the *maxzero* and the *discsubs* filtering strategies, we need to specify how many context clues we want to choose. We try following variants : choosing highest ranking context clue, choosing two highest ranking context clues and choosing three highest ranking context clues. We also try a variant where we let all context clues contribute to the compositional representation of the target word, but we weight them by the inverse of their rank. The results are reported in Table III. The rows are the dependency path lengths (e.g. L2 means we are using dependency paths of length at most 2) and the columns are the filtering strategies.

The results show that using longer dependency paths can improve cluster purity. The best result is obtained when we consider dependency paths up to length 4 and utilize *maxzero* filtering strategy choosing only the highest scoring context clue. The results illustrate the benefit of using longer dependency paths. Among the filtering strategies, *maxzero*, *discsubs* are consistently better than using no filtering, especially when we only use the highest scoring context clue. *nostop* also achieves good performance. On the other hand, *content* is not better than using no filtering.

Our second goal is to compare our approach to previous approaches [5] and [12] as described in Section V. We split our evaluation dataset randomly into a development and a test set. The development set contains around 25% of all target words in our evaluation data. In the development set, we find the

	Cluster Purity
token_averaging	**51.19**
type_averaging	45.46
local_features	47.08

TABLE IV
WORD SENSE DISCRIMINATION : COMPARISON TO PREVIOUS APPROACHES

	Accuracy
token_vector	**65.52**
brown clusters	61.63
type_averaging	62.25
type_vector	62.76
noexpansion	59.96

TABLE V
FEATURE EXPANSION : COMPARISON TO PREVIOUS APPROACHES

best parameter setting for our extended compositional model – *maxzero* filtering with a dependency path length up to four gives best results. In the test set, we compare the performance of our approach with this setting to previous approaches. We use the same clustering parameters and evaluation metric as before. Table IV holds the comparison. *type_averaging* is the system based on [5], *local_features* represents the system utilizing local feature representation as described in [12] and *token_averaging* in bold is our system relying on the extended compositional model. The result show that our model improves over previous approaches. The improvements are statistically significant at the $p < .05$ level based on a paired t-test.

C. Feature Expansion

In this section, we use our compositional representation to augment a vanilla WSD system. We expand lexical features (i.e. CW and CF features) with token vectors derived from our extended compositional model. For comparison, we also use other word representations to augment the vanilla WSD system. One of them is brown clusters [13], which has been effectively used for feature expansion[14], [19] before. We induce brown clusters from the same corpus as we build the semantic space. We use in our experiments 400 brown clusters. We additionally utilize the additive context representation described in [5] and simple type vectors from our semantic space for further comparison.

We use the same development and test set from Section VI-B. In the development set, we find the best parameter setting for our extended compositional model via 10-fold cross-validation. For feature expansion, we have the best setting if we do not have any filtering and use dependency paths up to length four. In the test set, we compare the performance of our approach with this setting to other word representations via 10-fold cross-validation. Table V holds the comparison. *token_vector* in bold is the word representation relying on our extended compositional model. *type_averaging* is the word representation based on context vectors described in [5]. *type_vector* is the word representation based on simple type vectors from the semantic space. *brown clusters* is the word representation based on brown clusters and *noexpansion* is the plain WSD system. The results show that our model achieves a better performance than other word representations for feature expansion. The improvements are statistically significant at the $p < .05$ level based on a paired t-test.

VII. RELATED WORK

Recently, several researchers have investigated composition in distributional semantic models [9], [24], [15], [25], [26]. These models offer a powerful tool to represent meaning in context. Composition is achieved through algebraic operations on word vectors or word matrices. Our work relies on the methods introduced in [10]. [10] defines the composition between specific word pairs related over a grammatical dependency relation. Our work extends this model to longer arbitrary dependency relations and investigates methods to filter out uninformative dependency paths and context clues.

Our work on word sense discrimination is similar to [5], except that, instead of averaging the type vectors, we average the token vectors of the neighbouring words, which we compute with our extended compositional model. Note that, [5] uses an additive model by averaging type vectors of all words in a context. In contrast, our model relies on a multiplicative model as introduced in [10].

In this paper, we use token-based vectors from our extended compositional model for feature expansion. Several researchers have investigated methods to induce word representations from unlabeled large corpora and augment lexical features with their corresponding word representations such as word clustering [18], [19], [20], [21], word embeddings via neural language models [14] and distributional semantic models [2]. Most of these approaches except [21] use type-based word representations. Our method is one of the few token-based approaches for feature expansion.

VIII. CONCLUSIONS

In this paper, we utilize an extended compositional DSM for word sense discrimination and word sense disambiguation. Building on previous work, we extend element-wise multiplication model introduced in [10] to effectively incorporate richer contexts. We use arbitrary longer dependency paths and investigate methods to filter out uninformative context clues. Our experiments show that longer dependency paths introduce useful information improving performance and that filtering mechanisms are essential.

For word sense discrimination, we build context vectors from the word representations based on the extended compositional model. For word sense disambiguation, we augment lexical features with their word representations based on the same extended compositional model. We show that our approach can improve both word sense discrimination and word sense disambiguation.

To our knowledge, we are the first ones to apply a compositional DSM to feature expansion. It is one of the few instances of token-based feature expansion with [21].

IX. Acknowledgments

This material is based in part upon work supported by National Science Foundation awards #0917170 and #0916046. The authors are grateful to the three paper reviewers and to Alexander Conrad for their helpful suggestions.

References

[1] P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," Mar. 2010. [Online]. Available: http://dx.doi.org/10.1613/jair.2934

[2] M. Sahlgren, "The word-space model: using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces." Ph.D. dissertation, Stockholm University, 2006.

[3] J. Bullinaria and J. Levy, "Extracting semantic representations from word co-occurrence statistics: A computational study," *Behavior Research Methods*, vol. 39, pp. 510–526, 2007, 10.3758/BF03193020. [Online]. Available: http://dx.doi.org/10.3758/BF03193020

[4] Z. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[5] H. Schutze, "Automatic word sense discrimination," *Computational Linguistics*, vol. 24, no. 1, pp. 97–124, 1998.

[6] D. Lin and P. Pantel, "Discovery of inference rules for question answering," *Journal of Natural Language Engineering*, vol. 7, no. 3, 2001.

[7] R. Rapp, *A freely available automatically generated thesaurus of related words*, 2004, pp. 395–398.

[8] T. K. Landauer and S. T. Dutnais, "A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychological review*, pp. 211–240, 1997.

[9] K. Erk and S. Padó, "A structured vector space model for word meaning in context," in *EMNLP*, 2008, pp. 897–906.

[10] J. Mitchell and M. Lapata, "Vector-based models of semantic composition," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 236–244. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1028

[11] S. Thater, G. Dinu, and M. Pinkal, "Ranking paraphrases in context," in *Proceedings of the 2009 Workshop on Applied Textual Inference*, ser. TextInfer '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 44–47. [Online]. Available: http://dl.acm.org/citation.cfm?id=1708141.1708149

[12] A. Purandare and T. Pedersen, "Word sense discrimination by clustering contexts in vector and similarity spaces," in *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2004)*, Boston, 2004.

[13] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Comput. Linguist.*, vol. 18, no. 4, pp. 467–479, Dec. 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=176313.176316

[14] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ser. ACL '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 384–394. [Online]. Available: http://dl.acm.org/citation.cfm?id=1858681.1858721

[15] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive Science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[16] C. Leacock and M. Chodorow, "Combining local context and WordNet sense similarity for word sense identification," in *WordNet, An Electronic Lexical Database*. The MIT Press, 1998.

[17] R. Mihalcea, "Instance based learning with automatic feature selection applied to Word Sense Disambiguation," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, August 2002.

[18] S. Miller, J. Guinness, and A. Zamanian, "Name tagging with word clusters and discriminative training," in *HLT-NAACL 2004: Main Proceedings*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 337–342.

[19] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, ser. CoNLL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 147–155. [Online]. Available: http://dl.acm.org/citation.cfm?id=1596374.1596399

[20] T. Koo, X. Carreras, and M. Collins, "Simple Semi-supervised Dependency Parsing," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, Jun. 2008, pp. 595–603. [Online]. Available: http://www.aclweb.org/anthology-new/P/P08/P08-1068.bib

[21] F. Huang and A. Yates, "Distributional representations for handling sparsity in supervised sequence-labeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ser. ACL '09. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 495–503. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687878.1687948

[22] J. Preiss and D. Yarowsky, Eds., *Proceedings of SENSEVAL-2, Association for Computational Linguistics Workshop*, Toulouse, France, 2001.

[23] R. Mihalcea and P. Edmonds, Eds., *Proceedings of SENSEVAL-3, Association for Computational Linguistics Workshop*, Barcelona, Spain, 2004.

[24] J. Reisinger and R. J. Mooney, "Multi-prototype vector-space models of word meaning," in *HLT-NAACL*, 2010, pp. 109–117.

[25] S. Rudolph and E. Giesbrecht, "Compositional matrix-space models of language," in *ACL*, 2010, pp. 907–916.

[26] E. Grefenstette and M. Sadrzadeh, "Experimental support for a categorical compositional distributional model of meaning," *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011. [Online]. Available: http://www.aclweb.org/anthology-new/D/D11/D11-1129.pdf