Word Meaning in Context: A Simple and Effective Vector Model

Stefan Thater and **Hagen Fürstenau** and **Manfred Pinkal**
Department of Computational Linguistics
Saarland University
{stth, hagenf, pinkal}@coli.uni-saarland.de

Abstract

We present a model that represents word meaning in context by vectors which are modified according to the words in the target's syntactic context. Contextualization of a vector is realized by reweighting its components, based on distributional information about the context words. Evaluation on a paraphrase ranking task derived from the SemEval 2007 Lexical Substitution Task shows that our model outperforms all previous models on this task. We show that our model supports a wider range of applications by evaluating it on a word sense disambiguation task. Results show that our model achieves state-of-the-art performance.

1 Introduction

Distributional vector-space models of word meaning have proven helpful for a number of basic natural language processing tasks, such as word sense discrimination (Schütze, 1998) and disambiguation (McCarthy et al., 2004), or modeling of selectional preferences (Erk, 2007), and have been successfully used in a variety of applications like information retrieval (Manning et al., 2008) or question answering (Tellex et al., 2003). Standard distributional models of meaning are attractive because they are simple, have wide coverage, and, in particular, can be acquired using unsupervised methods at virtually no cost. Vector-space models of meaning lend themselves as a basis for determining a soft and gradual concept of semantic similarity (e.g., through the *cosine* measure), which does not rely on a fixed set of dictionary senses with their well-known problems (Kilgarriff, 1997).

The sensitivity of word meaning to the context of use, however, poses a major challenge for distributional semantics. Meaning vectors are based on co-occurrence counts for *words* across all word senses and usages. This means that, for instance, any occurrence of the verb *charge*, such as in the expressions *charge a fee* or *charge a battery*, is assigned the same vector representation, ignoring the difference of word sense. On the other hand, the fact that *charge* and *impose* are near-synonyms in *charge/impose a fee* will not be properly reflected in their respective meaning vectors, since the former, but not the latter, includes (context words reflecting) the "supply electricity" sense of *charge*.

The problem of modeling context-sensitivity in a distributional framework has first been addressed in the seminal paper of Schütze (1998), who uses second-order bag-of-words vectors for the task of word sense discrimination. 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 "syntax-sensitive contextualization" (Thater et al., 2010). These approaches have in common that the contextual influence on the meaning of a target word w 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.

The results published during the last couple of years show a considerable increase of performance, but at the price of an increasing complexity and lack of intuitive transparency of the models. In this paper, we will demonstrate that one can keep the model simple and at the same time outperform the state of the art. We achieve this as follows: First, we take a different, more general view on the basic operation of contextualization. Like the aforementioned approaches, we model contextualization as modification of the target vector, but we do not restrict this operation to variants of vector composition, but consider a broader range of

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 1134–1143,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

operations, which *re-weight* individual vector components. Second, we identify the distributional similarity score between the words defining the vector components on the one hand, and the actual context words in a given syntactic position on the other hand as the most effective basis for this re-weighting.

We evaluate our method on two different tasks: *paraphrase ranking* and *word sense disambiguation*. The paraphrase ranking task has been used in several approaches and provides benchmarks for our system, and the controlled conditions of the experiment make it easy to assess the influence of different design decisions on the performance. In practical terms, we will use a paraphrase ranking task derived from the SemEval 2007 Lexical Substitution Task (McCarthy and Navigli, 2007). We exceed the state of the art by almost 6% in terms of generalized average precision.

The application to word sense disambiguation (WSD) demonstrates that our model is more generally applicable. We phrase the WSD task as a paraphrase ranking task: Roughly speaking, finding the contextually appropriate word sense amounts to identifying the WordNet synset containing the best paraphrase candidate for the target. We evaluate our system on the SemEval 2007 coarse-grained unsupervised WSD task (Navigli et al., 2007). Our results are competitive to the results reported in the literature.

Plan of the paper. We will first review related work in Section 2, before we present our model in Section 3. We evaluate our model's performance on a paraphrase ranking task in Section 4 and on the task of word sense disambiguation in Section 5. Section 6 concludes.

2 Related work

Inspired by earlier work of Kintsch (2001), who proposes a network algorithm to extract context-specific vector representations for words in context, Mitchell and Lapata (2008) investigate the systematic combination of distributional representations of word meaning along syntactic structure. They propose 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 for the task under consideration. They consider their proposal primarily under the aspect of compositionality, but it can also be taken to be a method to contextualize a target word through its dependents.

Erk and Padó (2008) propose structured vector representations, where each word is characterized by a standard co-occurrence vector, plus separate vector representations for the (inverse) selectional preferences for subject, object, and other syntactic relations. Contextualization is modeled by combining, e.g., the basic vector of the target verb with the selectional preferences of subject and object.

Thater et al. (2010) propose a similar approach, where word meaning is modeled as a second-order vector obtained by summing over first-order vectors representing the inverse selectional preferences of a word's syntactic arguments. Contextualization is modeled as above in terms of vector composition. Among the aforementioned approaches, their proposal performs best, but at the cost of a rather complex and unintuitive concept of second-order co-occurrence vectors.

Other approaches achieve good results without using vector composition. Dinu and Lapata (2010) represent word meaning in context by using a latent variable model, where context-dependence is modeled by conditioning the latent variable on the context in which a word occurs. Similar proposals have been made by Reisinger and Mooney (2010a) and Li et al. (2010).

A different approach has been taken by Erk and Padó (2010) and Reisinger and Mooney (2010b). Instead of "refining" vector representations ranging over all words in a corpus by means of vector composition, they start out from "token" vectors for individual instances of words in context, and then group these token vectors into different sense-specific clusters.

3 The model

We propose a model of word meaning that allows the computation of vector representations for *individual uses* of words, characterizing the specific meaning of a word in its sentential context. For instance, the vector of the verb *charge* in the expression *charge a tax* should reflect its monetary sense, while its vector in the expression *charge a battery* should be representative of its "supply electricity" sense.

We derive a *contextualized* vector from the basic meaning vector of a target word by *reweighting* its components on the basis of the context of the occurrence, where we take the context to be made

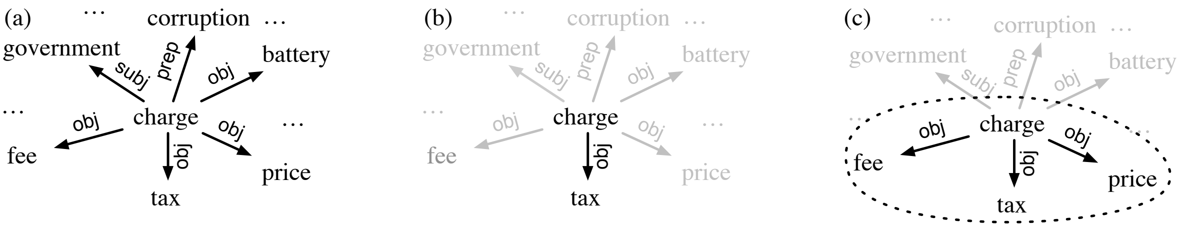

Figure 1: Graphical representation for a basic vector for *charge* (a), and two contextualized vectors for *charge* in context *charge a tax*, obtained by (b) a strict and (c) our more sophisticated contextualization method based on semantic similarity.

up of the direct syntactic dependents of the target (and its direct inverse dependents). The dimensions of both basic and contextualized vectors represent co-occurring words in specific syntactic relations. Fig. 1a shows the basic vector for *charge* as an example, where we use arrows to indicate the internal structure of the vector; the weights of the vector components are omitted for simplicity.

The operation of contextualization reinforces those dimensions of the basic vector that are licensed by the context of the specific instance under consideration. The easiest way of adapting the vector of a word to its context of use is to retain only those dimensions corresponding to its syntactic neighbors, which results in an extremely sparse vector with zero values for most of its dimensions. For instance, contextualizing the vector for *charge* in *charge a tax* (Fig. 1b) would zero out all (r,w) components with $r \neq$ OBJ or $w \neq$ *tax*, retaining only one non-zero dimension (the one for *tax*).

As we will see in Section 4, this simple approach is surprisingly successful. However, we achieve substantially better results by leveraging *semantic similarity information* about the context words. Instead of considering only the dimensions of the context words themselves, we retain dimensions of those words that are distributionally similar to the context words, weighted by their similarity score. The vector for *charge* in *charge a tax* will then contain additional non-zero dimensions for all words similar to *tax* (Fig. 1c). In a way, similarity-based contextualization is a formalization of the intuitive concept of “the meaning of w in the context of a word *like* w'.”

Formal description. We assume a set W of words and a set R of syntactic relations. The latter includes dependency relation labels such as SUBJ or OBJ for *subject* and *object*, as well as the corresponding inverse relations such as SUBJ^{-1}. We represent the meaning of any word $w \in W$ by a vector in the vector space V spanned by the set of basis vectors $\{\mathbf{e}_{(r,w')} \mid r \in R, w' \in W\}$. Such a vector records the association strength between w and any context word w' occurring in relation r. Specifically, we associate a word $w \in W$ with a vector $\mathbf{v}(w) \in V$ by setting

$$\mathbf{v}(w) := \sum_{r \in R, w' \in W} f(w,r,w') \cdot \mathbf{e}_{(r,w')}$$

where f is a function that assigns a weight to the dependency triple (w,r,w'). In the simplest case, this could be the frequency of w occurring together with w' in relation r in a corpus of dependency trees. In the experiments reported below, we use *pointwise mutual information* (Church and Hanks, 1990) instead, as it proved superior to raw frequency counts:

$$PMI(w,r,w') = \log \frac{p(w,w' \mid r)}{p(w,\cdot \mid r)p(\cdot,w' \mid r)}$$

Here the dots stand for marginalization over the relevant variables.

Given an occurrence of a word w in the context of another word w_c, related by the syntactic relation r_c, we now define a contextualized version of $\mathbf{v}(w)$ by *reweighting* the vector components. We set

$$\mathbf{v}_{r_c,w_c}(w) := \sum_{r \in R, w' \in W} \alpha_{r_c,w_c,r,w'} \cdot f(w,r,w') \cdot \mathbf{e}_{(r,w')}$$

Here, the weights $\alpha_{r_c,w_c,r,w'}$ quantify the degree to which a vector dimension (r,w') is compatible with the observed context (r_c,w_c). We consider three alternative definitions of these weights, corresponding to the three cases shown in Figure 1:

No contextualization: $\alpha_{r_c,w_c,r,w'} := 1$

In this case the definition of $\mathbf{v}_{r_c,w_c}(w)$ coincides with that of $\mathbf{v}(w)$.

Strict contextualization:

$$\alpha_{r_c,w_c,r,w'} := \delta_{r_c,r}\delta_{w_c,w'} = \begin{cases} 1 & \text{if } r_c = r \text{ and } w_c = w' \\ 0 & \text{else} \end{cases}$$

Here, we only retain the one dimension (r_c, w_c) that is licensed by the context and set all other dimensions to 0.

Similarity-based contextualization:

$$\alpha_{r_c,w_c,r,w'} := \delta_{r_c,r} \cdot \mathrm{sim}(w_c, w') = \begin{cases} \mathrm{sim}(w_c, w') & \text{if } r_c = r \\ 0 & \text{else} \end{cases}$$

Here, we generalize over the surface context and license all words w' that are semantically similar to the context word w_c.

While any measure of semantic similarity can be employed, in the experiments reported below we compute the similarity between w_c and w' as the cosine of the angle between their basic vector representations $\mathbf{v}(w_c)$ and $\mathbf{v}(w')$.

Of course, we want to take into account more than a single context word for a given occurrence of w. Given context words $w_1, \ldots, w_n$ and corresponding syntactic relations $r_1, \ldots, r_n$, we obtain a contextualized vector of w by superimposing the vectors $\mathbf{v}_{r_i,w_i}$ $(1 \leq i \leq n)$ through vector addition:

$$\mathbf{v}_{r_1,w_1,\ldots,r_n,w_n}(w) := \sum_{i=1}^{n} \mathbf{v}_{r_i,w_i}(w)$$

The resulting vector $\mathbf{v}_{r_1,w_1,\ldots,r_n,w_n}(w)$ is our completely contextualized representation for the word w that contains information about all context words.

4 Ranking Paraphrases

In this section, we evaluate to what extent our model supports the choice of contextually appropriate paraphrases for different uses of a target word. We follow previous work (Thater et al., 2010; Erk and Padó, 2010; Dinu and Lapata, 2010) and consider the following task: We are given a target word w in a sentential context and a set of reference words $w_1, \ldots, w_k$, where each w_i is a lexical paraphrase of w in one of w's senses. The task is to rank the candidate words w_i according to their appropriateness as paraphrases of *w in the given context*. Ideally, the model will rank, for instance, *levy* higher than *recharge* as a paraphrase of *charge* in *charge a fee*, and lower in *charge the battery.*

4.1 Experimental Set-up

Gold standard. We derive our gold standard from the SemEval 2007 lexical substitution task dataset (McCarthy and Navigli, 2007). The original dataset contains 10 instances for each of 201 target words (nouns, verbs, adjectives and adverbs) in different sentential contexts. For each instance, five subjects were asked to name appropriate paraphrases. Table 1 shows an example of three instances of *charge* together with their gold standard paraphrases. Each paraphrase comes with a weight, which corresponds to the number of times it was chosen by the different subjects.

The original task addresses two subtasks: identifying paraphrase candidates and ranking them according to the context. Here, we restrict ourselves to the second subtask. Following previous work, we pool all annotated gold-standard paraphrases of a target word w across all contexts into a set of *paraphrase candidates* for w, which our model is supposed to rank with respect to contextual appropriateness for the individual instances of w. We do not extract multi-word expressions, for which our model cannot compute vector representations, and obtain a dataset consisting of 1986 instances for 197 different words. In our derived dataset, each word type has an average of 17 paraphrases, 3.5 of which are correct (on average) for individual instances of the word.

Vector space. We draw on dependency trees obtained by parsing the English Gigaword corpus (LDC2003T05) to build our vector space model. The corpus consists of news from several newswire services, and contains over four million documents. We used the Stanford parser (de Marneffe et al., 2006) to parse the corpus. The resulting dependency trees were modified in a post-processing step by folding prepositions into edge labels to make the relation between a head word and the head noun of a prepositional phrase explicit. Furthermore, we collapsed particle verb constructions into single nodes. To facilitate processing and reduce noise, we excluded all dependency triples that occurred less than 3 times or had a PMI score below 0, which resulted in a corpus of about 888 million dependency triples accounting for 28 million triple types.

To further reduce computational costs, we set higher frequency and PMI thresholds for the computation of the similarity scores used in the contextualization of vectors: in the experiments reported

Sentence	Substitution candidates
Annual fees are *charged* on a pro-rata basis [...]	levy 2; require 1; impose 1; demand 1
Plug in you h 10 in the usb outlet and it will *charge* without the plug in adaptor.	recharge 2; supply electricity 1; charge up 1
Pauline Gilmore, 32, was *charged* with possessing a blast bomb.	indict 3; accuse of 2; accuse 1

Table 1: Three examples from the lexical substitution task data set for the target word *charge*.

below, we consider only (vectors based on) dependency triples that occur at least 5 times and have a PMI score of at least 2. Note that these thresholds are used only to speed up processing. The effect on the overall performance is minimal: an experiment on a randomly chosen 10% subset of the test set shows that we obtain almost identical scores, but runtime is reduced by a factor of more than 35.

Scoring. We rank the paraphrase candidates for a target word in context by the similarity of their basic vectors to the contextualized vector of the target. Contextualizing both the target and the paraphrase candidate has been observed to reduce performance (Thater et al., 2010; Dinu and Lapata, 2010). Similarity is measured in terms of the dot product of the vectors. In cases where the Stanford parser produced dependency trees that are inconsistent with the information about the target word in the gold standard, or where the contextualized vector is zero, we use the basic vector of the target as a fallback. This fallback method applies to 7% of all instances in the dataset.

Evaluation method. Following previous work (Thater et al., 2010; Erk and Padó, 2010), we use *Generalized Average Precision* (Kishida, 2005) to compare the ranking predicted by our model with the gold standard. GAP takes values between 1.0 and 0.0, where a value of 1.0 indicates that all correct items are ranked before all incorrect ones, and that higher-weighted items are ranked before lower-weighted ones. Statistical significance of differences in performance are computed by approximate randomization (Chinchor et al., 1993).

4.2 Results

Table 2 shows results for three versions of our model, corresponding to the three definitions of the weighting factors that were detailed in Section 3:

(a) No contextualization

(b) Strict contextualization

(c) Similarity-based contextualization

POS	Random	No context	Strict	Sim.-based
Verb	27.4	38.4	41.6	48.8
Noun	30.1	45.2	47.3	52.9
Adj	28.4	42.2	45.8	51.1
Adv	36.4	51.6	50.6	55.3
All	30.0	43.7	46.0	51.7

Table 2: Results for our model using different contextualization methods, compared to a random baseline.

In addition, we show the performance of a baseline that ranks paraphrase candidates randomly.

We observe that similarity-based contextualization is very effective, improving performance by 8% compared to the "no context" variant, and still by almost 6% compared to the strict variant that uses surface context only. The differences are statistically significant ($p < 0.001$).

Figure 2 provides a different view on system performance. It shows how often the k first candidates in the ranking contain at least one gold standard paraphrase. In particular, we can observe that similarity-based contextualization predicts a good top-ranked candidate in 55% of the cases; the top three contain a correct paraphrases in more than 80% of the cases.

Table 3 compares our model to previous models that have been evaluated using the Lexical Substitution Task (LST) dataset. Our model outperforms all previously proposed methods. Although all models have been evaluated on test-sets derived from the LST dataset in essentially the same way, the datasets differ slightly due to technical details, so strictly speaking the results cannot be compared directly. However, since all authors report similar

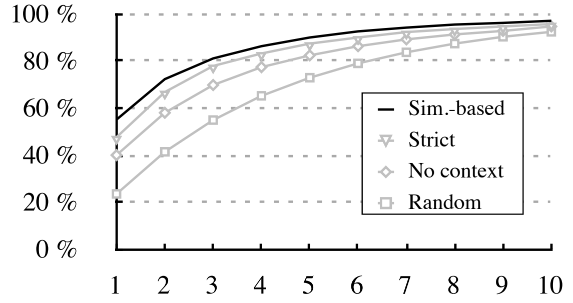

Figure 2: The figure shows how often the k first candidates in the ranking contain at least one gold standard paraphrase (for $k \leq 10$).

Model	GAP	Random
Erk and Padó (2008)	27.4[†]	N/A
Erk and Padó (2010)	38.6[‡]	28.5
Dinu and Lapata (2010)	42.9	30.3
Thater et al. (2010)	46.0	30.0
Our model	51.7	30.0

[†] Cited from Erk and Padó (2010). The result refers to a small subset of the Lexical Substitution Task dataset.

[‡] Evaluated on nouns, verbs, and adjectives (not adv.).

Table 3: Comparison to previous work

scores for the random baselines, we assume that the complexity of the subsets used in previous work is more or less comparable.

Learning curve. The corpus used in our study is much larger than the British National Corpus (BNC) that has been used, for instance, in Erk and Padó's (2008; 2010) models. To assess the contribution of the corpus size to the performance of our model, we randomized the order of dependency trees in the parsed Gigaword corpus and constructed vector space models using increasing subsets of the complete corpus with a step size of 5%. The resulting learning curve is shown in Figure 3. We see that our model performs well even on small subsets of Gigaword. When we use only 5% of the dependency trees, which is roughly two third of the size of BNC, we already obtain a GAP score of 46.0%, which is 5.7% less than our result with full Gigaword, but 7.4% more than the best reported BNC-based model.

Syntactic information. Finally, we investigated the impact of syntactic information by comparing our model against two variants: (i) a "bag of words" variant that does not use syntactic information at

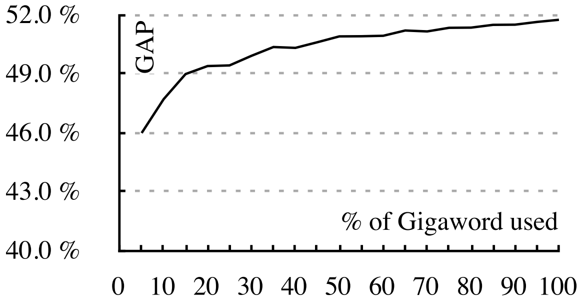

Figure 3: Learning curve: GAP with varying corpus size.

all and (ii) a "syntactically filtered" variant similar to Padó and Lapata (2007) that uses syntactic information but does not explicitly represent syntactic role information in the vector representations. Variant (i) is based on co-occurrence statistics on pairs (w, w') of content words within a five-word window; for variant (ii) we consider all pairs (w, w') such that w and w' are linked by some syntactic relation. Technically, we represent these pairs as dependency triples involving some arbitrary fixed syntactic role label.

We observe that syntactic information contributes to the success of our approach both by selecting relevant context words and by characterizing their syntactic relations: In terms of GAP, the "bag of words" variant achieves 48.7%, the "syntactically filtered" variant 50.9%, and our full model 51.7%. The relatively small difference between the two syntactic variants, while maybe surprising at first sight, is explained by the fact that in most cases the syntactic role of a dependency triple is predictable from the words it connects: For more than 88% of all dependency triples in Gigaword, the syntactic role is actually the most frequent one for the respective pair of words. Yet, the difference between the two variants is statistically significant ($p < 0.05$): The model supports correct decisions in those cases where syntactic role information matters.

5 Word Sense Disambiguation

In a second experiment, we applied our model to the task of word-sense disambiguation. For an individual instance of a word, we predict the correct WordNet sense (Fellbaum, 1998) of the target based on its immediate syntactic context, without relying on any manually annotated training data. Our system is *knowledge-based*, according to the classification of WSD approaches proposed in McCarthy

(2009) and Navigli (2009). It is a *knowledge-lean* system, in contrast to many other systems that exploit external resources, since it uses only a small subset of the structural information provided by WordNet – just as much as is required to adapt our contextualization model to the WSD task.

The state of the art in knowledge-based WSD systems not trained on annotated data is defined by the models of Navigli and Velardi (2005), Ponzetto and Navigli (2010) and Li et al. (2010). The former two rely on a rich inventory of additional knowledge resources. Li et al. (2010) restricts itself to WordNet information in a similar way as our approach, and therefore is our natural benchmark.

5.1 Method

We frame the task of choosing the right WordNet sense as a paraphrase ranking task like the one considered in Section 4, with all possible synonyms of the target word constituting the set of (lexical) paraphrase candidates. The basic idea for predicting a sense of the target word is to choose the synset that contains the most similar paraphrase. As the WordNet synsets of the target word are often singletons, just containing the target itself, we additionally include all words from direct hypernym, hyponym, and similar synsets (WordNet relation "similar to"). We ignore multiword expressions since our model does not provide vector representations for them.

While we generally found the richer collection of candidates to improve system performance, the inclusion of hypernyms can have a negative effect on sense discrimination, since different word senses frequently share the same hypernym. To counter this effect, we consider the average similarity scores of the best two paraphrase candidates of each sense rather than relying on the most similar candidate alone. More technically speaking, we collect all relevant sense paraphrases $c_{i,1}, \ldots, c_{i,k_i}$ for each sense s_i of the target word. We compare the contextualized vector of the target word to the basic meaning representations of these candidate words, obtaining a similarity score for each of them. The score of the sense s_i is then defined as the average of the scores of the two top-scoring candidate words, and the sense with the highest such score is predicted. Our model fails to predict a sense for an ambiguous target if the candidate set of any sense is empty, which can happen in cases where all applicable sense paraphrases are multiword expressions.

We will experiment with two instantiations of this model: the basic version described above, and a version that additionally integrates information about prior sense distributions by multiplying the score of each synset with its prior probability, and falls back to the most frequent sense in cases where the basic model fails to make a prediction. Prior probabilities are estimated by using sense frequency information from WordNet.

5.2 Experimental setup

Gold standard. We evaluate our model on the SemEval 2007 Coarse-grained English All-words Task (Navigli et al., 2007) test set. The test set consists of 5,377 words of running text from five documents from different genres. All open-class words in this corpus are annotated with coarse-grained sense labels, which are defined as clusters of WordNet senses and are obtained by mapping WordNet 2.1 senses to the Oxford Dictionary of English (Soanes and Stevenson, 2003). On a subset of 710 instances an inter-annotator agreement of 93.80% was reported, which can be considered the upper bound for any WSD system on the data set.

Predicting coarse-grained senses. The method described in Section 5.1 predicts (fine-grained) WordNet senses. It can be straightforwardly extended to the coarse-grained WSD task by picking the sense cluster containing the top-ranked synset. We achieved slightly better results by applying a different method: We normalize the scores of all synsets so that they sum up to 1, which allows us to interpret them as a probability distribution. We then compute probabilities for each sense cluster by aggregating over its constituent synsets, and predict the most probable one (which need not be the one containing the most probable synset).

Baselines. We compare our model against a random baseline and the most frequent sense (MFS) baseline that always predicts the sense with the highest sense frequency according to WordNet.

5.3 Results

Table 4 summarizes results on the test set in terms of precision, and compares them to two baselines and the state-of-the-art system of Li et al. (2010). Except in the case of our basic system (-MFS) without prior information, which cannot use information about most frequent senses as fallback and covers only 74.6% of the test cases, coverage is 100% and therefore precision coincides with recall.

Model	+MFS	-MFS
Random	52.4	52.4
Most frequent sense (MFS)	78.9	—
Li et al. (2010)	81.3‡	78.8‡
Our Model	80.9	78.7†
Combined system	82.2	78.9

† Covers 74.6% of the dataset.

‡ Results reported here are higher than the results reported by Li et al. (2010). Our results are based on the scoring script provided by the organizers of the SemEval 2007 shared task. Differences are due to details such as sensitivity to capitalization when system predictions are compared with the gold standard.

Table 4: Precision of our model on the WSD task, with (+MFS) and without (-MFS) prior knowledge about sense distributions, compared to the state-of-the-art system by Li et al.

We can see that our model's performance is competitive with the state of the art: In both settings our model outperforms the two baselines, and reaches the performance level of the benchmark system of Li et al. (2010).

Interestingly, the strengths of our and Li el al.'s systems are complementary. For example, in the sentence "The *diners* at my table simply lit more Gauloises [...]," our model correctly predicts the sense "person eating a meal" of the target *diners*, based on the leading sense paraphrase *eater*. The system by Li et al. (2010), on the other hand, predicts the sense "passenger car where food is served", which fits the general topic similarly well, but is highly implausible in the given syntactic context. However, in the sentence "The program text, or source, was converted into machine instructions using a special program called a *compiler*," the system by Li et al. (2010) is able to leverage topical clues to correctly predict the software sense of *compiler*, whereas our system ranks the sense paraphrase *author* over *program* and thus incorrectly predicts the sense "person who compiles encyclopedias."

Given this complementary nature of the two systems, we tried to combine them in a straightforward way, by averaging their predicted probability distributions (defaulting to Li et al. for instances not covered by our model). Table 4 shows that the combined system outperforms both individual systems both with and without MFS information. In the former case (with MFS), the improvement of 0.9% is statistically significant ($p < 0.01$) according to McNemar's test.

6 Conclusions and Future Work

We have presented a technically simple and intuitively transparent vector space model of word meaning in context. Contextualization of a vector is realized by reweighting its components, using semantic similarity information about the words occurring in the target's local syntactic context.

We evaluated our method on a paraphrase ranking task derived from the SemEval 2007 Lexical Substitution Task dataset and showed that it substantially outperforms all previous approaches, exceeding the state of the art by almost 6% in terms of generalized average precision. We showed that our model supports a wider range of application by evaluating it on a word sense disambiguation task. The model reaches the performance level of the state-of-the-art benchmark system of Li et al. (2010). The combination of the two systems performs significantly better than either system used in isolation, and outperforms the most-frequent-sense baseline by over 3%.

The contextualization operation takes only the *words* in the targets *local* syntactic context into account. A natural direction for future research is to generalize the contextualization operation so that the context words themselves can be contextualized in a recursive fashion and all words in the target's complete syntactic environment can contribute information.

Our present model incorporates syntactic relations, although semantic information should ideally be expressed in terms of underlying semantic roles. We have seen that the use of syntactically structured vector representations leads to a relatively small, but statistically significant increase in performance, compared to variants of our model that do not represent rich syntactic information. We expect that further progress can be made by integrating semantic role information.

Acknowledgments. We would like to thank Georgiana Dinu and Linlin Li for their support and helpful comments. This work was supported by the Cluster of Excellence "Multimodal Computing and Interaction", funded by the German Excellence Initiative, and the project SALSA, funded by DFG (German Science Foundation).

References

Nancy Chinchor, David D. Lewis, and Lynette Hirschmant. 1993. Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3). *Computational Linguistics*, 19(3):409–449.

Kenneth W. Church and Patrick Hanks. 1990. Word association, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI, USA.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, Uppsala, Sweden.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Christiane Fellbaum, editor. 1998. *Wordnet: An Electronic Lexical Database*. Bradford Book.

Adam Kilgarriff. 1997. I don't believe in word senses. *Computers and Humanities*, 31(2):91–113.

Walter Kintsch. 2001. Predication. *Cognitive Science*, 25:173–202.

Kazuaki Kishida. 2005. Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. *NII Technical Report*.

Linlin Li, Benjamin Roth, and Caroline Sporleder. 2010. Topic models for word sense disambiguation and token-based idiom detection. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, Prague, Czech Republic.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, Barcelona, Spain.

Diana McCarthy. 2009. Word sense disambiguation: An overview. *Language and Linguistics Compass*, 3(2):537–558.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, Columbus, OH, USA.

Roberto Navigli and Paola Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1088.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic.

Roberto Navigli. 2009. Word Sense Disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Sebastian Padó and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden.

Joseph Reisinger and Raymond Mooney. 2010a. A mixture model with sharing for lexical semantics. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA.

Joseph Reisinger and Raymond J. Mooney. 2010b. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Catherine Soanes and Angus Stevenson, editors. 2003. *Oxford Dictionary of English*. Oxford University Press.

Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

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*, Uppsala, Sweden.