A Structured Vector Space Model for Word Meaning in Context

Katrin Erk
Department of Linguistics
University of Texas at Austin
katrin.erk@mail.utexas.edu

Sebastian Padó
Department of Linguistics
Stanford University
pado@stanford.edu

Abstract

We address the task of computing vector space representations for the meaning of word occurrences, which can vary widely according to context. This task is a crucial step towards a robust, vector-based compositional account of sentence meaning. We argue that existing models for this task do not take syntactic structure sufficiently into account.

We present a novel *structured vector space* model that addresses these issues by incorporating the selectional preferences for words' argument positions. This makes it possible to integrate syntax into the computation of word meaning in context. In addition, the model performs at and above the state of the art for modeling the contextual adequacy of paraphrases.

1 Introduction

Semantic spaces are a popular framework for the representation of word meaning, encoding the meaning of lemmas as high-dimensional vectors. In the default case, the components of these vectors measure the co-occurrence of the lemma with context features over a large corpus. These vectors are able to provide a robust model of *semantic similarity* that has been used in NLP (Salton et al., 1975; McCarthy and Carroll, 2003; Manning et al., 2008) and to model experimental results in cognitive science (Landauer and Dumais, 1997; McDonald and Ramscar, 2001). Semantic spaces are attractive because they provide a model of word meaning that is independent of dictionary senses and their much-discussed problems (Kilgarriff, 1997; McCarthy and Navigli, 2007).

In a default semantic space as described above, each vector represents one *lemma*, averaging over all its possible usages (Landauer and Dumais, 1997; Lund and Burgess, 1996). Since the meaning of words can vary substantially between occurrences (e.g., for polysemous words), the next necessary step is to characterize the meaning of individual *words in context*.

There have been several approaches in the literature (Smolensky, 1990; Schütze, 1998; Kintsch, 2001; McDonald and Brew, 2004; Mitchell and Lapata, 2008) that compute meaning in context from lemma vectors. Most of these studies phrase the problem as one of vector composition: The meaning of a target occurrence a in context b is a single new vector c that is a function (for example, the centroid) of the vectors: $c = a \odot b$.

The context b can consist of as little as one word, as shown in Example (1). In (1a), the meaning of *catch* combined with *ball* is similar to *grab*, while in (1b), combined with *disease*, it can be paraphrased by *contract*. Conversely, verbs can influence the interpretation of nouns: In (1a), *ball* is understood as a spherical object, and in (1c) as a dancing event.

(1) a. catch a ball
b. catch a disease
c. attend a ball

In this paper, we argue that models of word meaning relying on this procedure of vector composition are limited both in their scope and scalability. The underlying shortcoming is a failure to consider *syntax* in two important ways.

The syntactic relation is ignored. The first problem concerns the manner of vector composition, which ignores the relation between the target a and its context b. This relation can have a decisive influence on their interpretation, as Example (2) shows:

(2) a. a horse draws
b. draw a horse

In (2a), the meaning of the verb *draw* can be paraphrased as *pull*, while in (2b) it is similar to *sketch*. This difference in meaning is due to the difference in relation: in (2a), *horse* is the subject, while in (2b) it is the object. On the modeling side, however, a vector combination function that ignores the relation will assign the same representation to (2a) and (2b). Thus, existing models are systematically unable to capture this class of phenomena.

Single vectors are too weak to represent phrases. The second problem arises in the context of the important open question of how semantic spaces can "scale up" to provide interesting meaning representations for entire sentences. We believe that the current vector composition methods, which result in a single vector c, are not informative enough for this purpose. One proposal for "scaling up" is to straightforwardly interpret $c = a \odot b$ as the *meaning of the phrase* $a + b$ (Kintsch, 2001; Mitchell and Lapata, 2008). The problem is that the vector c can only encode a fixed amount of structural information if its dimensionality is fixed, but there is no upper limit on sentence length, and hence on the amount of structure to be encoded. It is difficult to conceive how c could encode *deeper* semantic properties, like predicate-argument structure (distinguishing "dog bites man" and "man bites dog"), that are crucial for sentence-level semantic tasks such as the recognition of textual entailment (Dagan et al., 2006). An alternative approach to sentence meaning would be to use the vector space representation only for representing word meaning, and to represent sentence structure separately. Unfortunately, present models cannot provide this grounding either, since they compute a single vector c that provides the same representations for *both* the meanings of a and b in context.

In this paper, we propose a new, *structured vector space* model for word meaning (SVS) that addresses these problems. A SVS representation of a lemma comprises several vectors representing the word's lexical meaning as well as the *selectional preferences* that it has for its argument positions. The meaning of word a in context b is computed by combining a with b's selectional preference vector specific to the relation between a and b, addressing the first problem above. In an expression $a + b$, the meanings of a and b in this context are computed as two separate vectors a' and b'. These vectors can then be combined with a representation of the structure's expression (e.g., a parse tree), to address the second problem discussed above. We test the SVS model on the task of recognizing contextually appropriate paraphrases, finding that SVS performs at and above the state-of-the-art.

Plan of the paper. Section 2 reviews related work. Section 3 presents the SVS model for word meaning in context. Sections 4 to 6 relate experiments on the paraphrase appropriateness task.

2 Related Work

In this section we give a short overview over existing vector space based approaches to computing word meaning in context.

General context effects. The first category of models aims at integrating the widest possible range of context information without recourse to linguistic structure. The best-known work in this category is Schütze (1998). He first computes "first-order" vector representations for word meaning by collecting co-occurrence counts from the entire corpus. Then, he determines "second-order" vectors for individual word instances in their context, which is taken to be a simple surface window, by summing up all first-order vectors of the words in this context. The resulting vectors form sense clusters.

McDonald and Brew (2004) present a similar model. They compute the expectation for a word w_i in a sequence by summing the first-order vectors for the words w_1 to w_{i-1} and showed that the distance between expectation and first-order vector for w_i correlates with human reading times.

Predicate-argument combination. The second category of prior studies concentrates on contexts consisting of a single word only, typically modeling the combination of a predicate p and an argument a. Kintsch (2001) uses vector representations of p and a to identify the set of words that are similar to both p and a. After this set has been narrowed down in a self-inhibitory network, the meaning of the predicate-argument combination is obtained by computing the centroid of its members' vectors. The procedure does not take the relation between p and a into account.

Mitchell and Lapata (2008) propose a framework to represent the meaning of the combination $p + a$ as a function f operating on four components:

$$c = f(p, a, R, K) \qquad (3)$$

R is the relation holding between p and a, and K additional knowledge. This framework allows sensitivity to the relation. However, the concrete instantiations that Mitchell and Lapata consider disregards K and R, thus sharing the other models' limitations. They focus instead on methods for the direct combination of p and a: In a comparison between component-wise addition and multiplication of p and a, they find far superior results for the multiplication approach.

Tensor product-based models. Smolensky (1990) uses tensor product to combine two word vectors a and b into a vector c representing the expression $a+b$. The vector c is located in a very high-dimensional space and is thus capable of encoding the structure of the expression; however, this makes the model infeasible in practice, as dimensionality rises with every word added to the representation. Jones and Mewhort (2007) represent lemma meaning by using circular convolution to encode n-gram co-occurrence information into vectors of fixed dimensionality. Similar to Brew and McDonald (2004), they predict most likely next words in a sequence, without taking syntax into account.

Kernel methods. One of the main tests for the quality of models of word meaning in context is the ability to predict the appropriateness of paraphrases in given a context. Typically, a paraphrase applies only to some senses of a word, not all, as can be seen in the paraphrases "grab" and "contract" of "catch". Vector space models generally predict paraphrase appropriateness based on the similarity between vectors. This task can also be addressed with kernel methods, which project items into an implicit feature space for efficient similarity computation. Consequently, vector space methods and kernel methods have both been used for NLP tasks based on similarity, notably Information Retrieval and Textual Entailment. Nevertheless, they place their emphasis on different types of information. Current kernels are mostly tree kernels that compare syntactic structure, and use semantic information mostly for smoothing syntactic similarity (Moschitti and Quarteroni, 2008). In contrast, vector-space models focus on the interaction between the lexical meaning of words in composition.

3 A structured vector space model for word meaning in context

In this section, we define the structured vector space (SVS) model of word meaning.

The main intuition behind our model is to view the interpretation of a word in context as guided by *expectations about typical events*. For example, in (1a), we assume that upon hearing the phrase "catch a ball", the hearer will interpret the meaning of "catch" to match *typical actions that can be performed with a ball*. Similarly, the interpretation of "ball" will reflect the hearer's expectations about *typical things that can be caught*. This move to include typical arguments and predicates into a model of word meaning can be motivated both on cognitive and linguistic grounds.

In cognitive science, the central role of expectations about typical events for human language processing is well-established. Expectations affect reading times (McRae et al., 1998), the interpretation of participles (Ferretti et al., 2003), and sentence processing generally (Narayanan and Jurafsky, 2002; Padó et al., 2006). Expectations exist both for verbs and nouns (McRae et al., 1998; McRae et al., 2005).

In linguistics, expectations, in the form of *selectional restrictions* and *selectional preferences*, have long been used in semantic theories (Katz and Fodor, 1964; Wilks, 1975), and more recently induced from corpora (Resnik, 1996; Brockmann and Lapata, 2003). Attention has mostly been limited to selectional preferences of verbs, which have been used for example for syntactic disambiguation (Hindle and Rooth, 1993), word sense disambiguation (McCarthy and Carroll, 2003) and semantic role labeling (Gildea and Jurafsky, 2002). Recently, a vector-spaced model of selectional preferences has been proposed that computes the typicality of an argument simply through similarity to previously seen arguments (Erk, 2007; Padó et al., 2007).

We first present the SVS model of word meaning

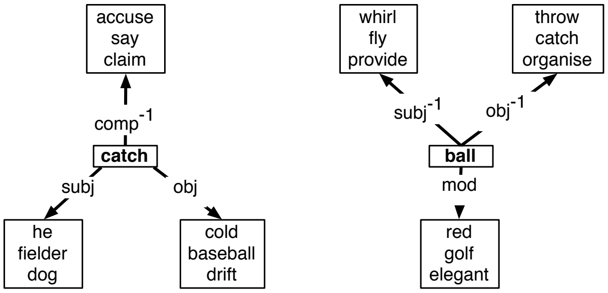

Figure 1: Structured meaning representations for noun *ball* and verb *catch*: lexical information plus expectations

that integrates lexical information with selectional preferences. Then, we show how the SVS model provides a new way of computing meaning in context.

Representing lemma meaning. We abandon the traditional choice of representing word meaning as a single vector. Instead, we encode each word as a combination of (a) one vector that models the *lexical* meaning of the word, and (b) a set of vectors, each of which represents the *semantic expectations/selectional preferences* for one particular relation that the word supports.[1]

The idea is illustrated in Fig. 1. In the representation of the verb *catch*, the central square stands for the lexical vector of *catch* itself. The three arrows link it to *catch*'s preferences for its subjects ($subj$), its objects (obj), and for verbs for which it appears as a complement ($comp^{-1}$). The figure shows the selectional preferences as word lists for readability; in practice, each selectional preference is a single vector (cf. Section 4). Likewise, *ball* is represented by one vector for *ball* itself, one for *ball*'s preferences for its modifiers (mod), one vector for the verbs of which it is a subject ($subj^{-1}$), and one for the verbs of which is an object (obj^{-1}).

This representation includes selectional preferences (like $subj$, obj, mod) exactly parallel to *inverse* selectional preferences ($subj^{-1}$, obj^{-1}, $comp^{-1}$). To our knowledge, preferences of the latter kind have not been studied in computational linguistics. However, their existence is supported in psycholinguistics by priming effects from nouns to typical verbs (McRae et al., 2005).

Formally, let D be a vector space (the set of possible vectors), and let $\mathcal{R}$ be some set of relation labels. In the *structured vector space (*SVS*)* model, we represent the meaning of a lemma w as a triple

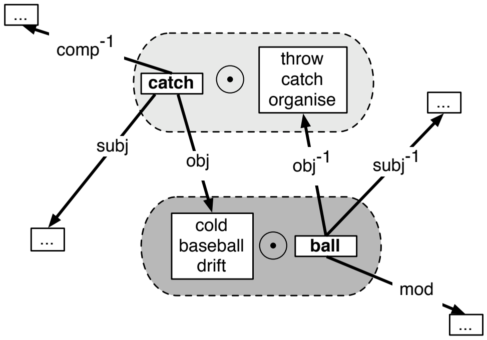

Figure 2: Combining predicate and argument via relation-specific semantic expectations

$$w = (v, R, R^{-1})$$

where $v \in D$ is a lexical vector describing the word w itself, $R : \mathcal{R} \to D$ maps each relation label onto a vector that describes w's selectional preferences, and $R^{-1} : \mathcal{R} \to D$ maps from role labels to vectors describing inverse selectional preferences of w. Both R and R^{-1} are partial functions. For example, the direct object preference would be undefined for intransitive verbs.

Computing meaning in context. The SVS model of lemma meaning permits us to compute the meaning of a word a in the context of another word b in a new way, via their selectional preferences. Let (v_a, R_a, R_a^{-1}) and (v_b, R_b, R_b^{-1}) be the representations of the two words, and let $r \in \mathcal{R}$ be the relation linking a to b. Then, we define the meaning of a and b in this context as a pair (a', b') of vectors, where a' is the meaning of a in the context of b, and b' the meaning of b in the context of a:

$$\begin{aligned} a' &= \big(v_a \odot R_b^{-1}(r), R_a - \{r\}, R_a^{-1}\big) \\ b' &= \big(v_b \odot R_a(r), R_b, R_b^{-1} - \{r\}\big) \end{aligned} \qquad (4)$$

where $v_1 \odot v_2$ is a direct vector combination function as in traditional models, e.g. addition or componentwise multiplication. If either $R_a(r)$ or $R_b^{-1}(r)$ are not defined, the combination fails. Afterwards, the argument position r is considered filled, and is deleted from R_a and R_b^{-1}.

[1] We do not commit to a particular set of relations; see the discussion at the end of this section.

Figure 2 illustrates this procedure on the representations from Figure 1. The dotted lines indicate that the lexical vector for *catch* is combined with the inverse object preference of *ball*. Likewise, the lexical vector for *ball* is combined with the object preference vector of *catch*.

Note that our procedure for computing meaning in context can be expressed within the framework of Mitchell and Lapata (Eq. (3)). We can encode the expectations of a and b as additional knowledge K. The combined representation c is the pair (a', b') that is computed according to our model (Eq. (4)).

The SVS scheme we have proposed incorporates syntactic information in a more general manner than previous models, and thus addresses the issues we have discussed in Section 1. Since the representation retains individual selectional preferences for all relations, combining the same words through different relations can (and will in general) result in different adapted representations. For instance, in the case of Example (2), we would expect the inverse subject preference of *horse* ("things that a horse typically does") to push the lexical vector of *draw* into the direction of *pulling*, while its inverse object preference ("things that are done to horses") suggest a different interpretation.

Rather than yielding a single, joint vector for the whole expression, our procedure for computing meaning in context results in one context-adapted meaning representation per word, similar to the output of a WSD system. As a consequence, our model can be combined with any formalism representing the structure of an expression. (The formalism used then determines the set $\mathcal{R}$ of relations.) For example, combining SVS with a dependency tree would yield a tree in which each node is labeled by a SVS tuple that represents the word's meaning in context.

4 Experimental setup

This section provides the background to the following experimental evaluation of SVS, including parameters used for computing the SVS representations that will be used in the experiments.

4.1 Experimental rationale

In this paper, we evaluate the SVS model against the task of predicting, given a predicate-argument pair, how appropriate a paraphrase (of either the predicate or the argument) is in that context. We perform two experiments that both use the paraphrase task, but differ in their emphasis. Experiment 1 replicates an existing evaluation against human judgments. This evaluation uses synthetic dataset, limited to one particular construction, and constructed to provide maximally distinct paraphrase candidates. Experiment 2 considers a broader class of constructions along with annotator-generated paraphrase candidates that are not screened for distinctness. In both experiments, we compare the SVS model against the state-of-the-art model by Mitchell and Lapata 2008 (henceforth M&L; cf. Sec. 2 for model details).

4.2 Parameter choices

Vector space. In our parameterization of the vector space, we largely follow M&L because their model has been rigorously evaluated and found to outperform a range of other models.

Our first space is a traditional "bag-of-words" vector space (BOW, (Lund and Burgess, 1996)). For each pair of a target word and context word, the BOW space records a function of their co-occurrence frequency within a surface window of size 10. The space is constructed from the British National Corpus (BNC), and uses the 2,000 most frequent context words as dimensions.

We also consider a "dependency-based" vector space (SYN, (Padó and Lapata, 2007)). In this space, target and context words have to be linked by a "valid" dependency path in a dependency graph to count as co-occurring.[2] This space was built from BNC dependency parses obtained from Minipar (Lin, 1993).

For both spaces, we used pre-experiments to compare two methods for the computation of vector components, namely raw co-occurrence counts, the standard model, and the pointwise mutual information (PMI) definition employed by M&L.

Selectional preferences. We use a simple, knowledge-lean representation for selectional preferences inspired by Erk (2007), who models selectional preference through similarity to seen filler vectors $\vec{v}_a$: We compute the selectional preference vector for word b and relation r as the weighted

[2] More specifically, we used the minimal context specification and plain weight function. See Padó and Lapata (2007).

centroid of seen filler vectors $\vec{v}_a$. We collect seen fillers from the Minipar-parse of the BNC.

Let $f(a, r, b)$ denote the frequency of a occurring in relation r to b in the parsed BNC, then

$$R_b(r)_{\text{SELPREF}} = \sum_{a: f(a,r,b)>0} f(a, r, b) \cdot \vec{v}_a \quad (5)$$

We call this base model SELPREF. We will also study two variants of SELPREF, based on two different hypotheses about what properties of the selectional preferences are particularly important for meaning adaption. The first model aims specifically at alleviating noise introduced by infrequent fillers, a common problem in data-driven approaches. It only uses fillers seen more often than a threshold θ. We call this model SELPREF-CUT:

$$R_b(r)_{\text{SELPREF-CUT}} = \sum_{a: f(a,r,b)>\theta} f(a, r, b) \cdot \vec{v}_a \quad (6)$$

Our second variant again aims at alleviating noise, but noise introduced by low-valued dimensions rather than infrequent fillers. It achieves this by taking each component of the selectional preference vector to the nth power. In this manner, dimensions with high counts are further inflated, while dimensions with low counts are depressed.[3] This model, SELPREF-POW, is defined as follows: If $R_b(r)_{\text{SELPREF}} = \langle v_1, \ldots, v_m \rangle$,

$$R_b(r)_{\text{SELPREF-POW}} = \langle v_1^n, \ldots, v_m^n \rangle \quad (7)$$

The inverse selectional preferences R_b^{-1} are defined analogously for all three model variants. We instantiate the vector combination function $\odot$ as component-wise multiplication, following M&L.

Baselines and significance testing. All tasks that we consider below involve judgments for the meaning of a word a in the context of a word b. A first baseline that every model must beat is simply using the original vector for a. We call this baseline "target only". Since we assume that the selectional preferences of b model the expectations for a, we use b's selectional preference vector for the given relation as a second baseline, "selpref only".

verb	subject	landmark	sim	judgment
slump	shoulder	slouch	high	7
slump	shoulder	decline	low	2
slump	value	slouch	low	3
slump	value	decline	high	7

Figure 3: Experiment 1: Human similarity judgements for subject-verb pair with high- and low-similarity landmarks

Differences between the performance of models were tested for significance using a stratified shuffling-based randomization test (Yeh, 2000).[4].

5 Exp. 1: Predicting similarity ratings

In our first experiment, we attempt to predict human similarity judgments. This experiment is a replication of the evaluation of M&L on their dataset[5].

Dataset. The M&L dataset comprises a total of 3,600 human similarity judgements for 120 experimental items. Each item, as shown in Figure 3, consists of an intransitive verb and a subject noun that are combined with a "landmark", a synonym of the verb that is chosen to be either similar or dissimilar to the verb in the context of the given subject.

The dataset was constructed by extracting pairs of subjects and intransitive verbs from a parsed version of the BNC. Each item was paired with two landmarks, chosen to be as dissimilar as possible according to a WordNet similarity measure. All nouns and verbs were subjected to a pretest, where only those with highly significant variations in human judgments across landmarks were retained.

For each item of the final dataset, judgements on a 7-point scale were elicited. For example, judges considered the compatible landmark "slouch" to be much more similar to "shoulder slumps" than the incompatible landmark "decline". In Figure 3, the column *sim* shows whether the experiment designers considered the respective landmark to have high or low similarity to the verb, and the column *judgment* shows a participant's judgments.

Experimental procedure. We used cosine to compute similarity to the lexical vector of the landmark.

[3] Since we focus on the size-invariant cosine similarity, the use of this model does not require normalization.

[4] The software is available at `http://www.nlpado.de/~sebastian/sigf.html`.

[5] We thank J. Mitchell and M. Lapata for providing their data.

Model	high	low	ρ
BOW space			
Target only	0.32	0.32	0.0
Selpref only	0.46	0.4	0.06**
M&L	0.25	0.15	0.20**
SELPREF	0.32	0.26	0.12**
SELPREF-CUT, θ=10	0.31	0.24	0.11**
SELPREF-POW, n=20	0.11	0.03	**0.27****
Upper bound	–	–	0.4
SYN space			
Target only	0.2	0.2	0.08**
Selpref only	0.27	0.21	0.16**
M&L	0.13	0.06	**0.24****
SELPREF	0.22	0.16	0.13**
SELPREF-CUT, θ=10	0.2	0.13	0.13**
SELPREF-POW, n=30	0.08	0.04	0.22**
Upper bound	–	–	0.4

Table 1: Experiment 1: Mean cosine similarity for items with high- and low-similarity landmarks; correlation with human judgements (ρ). (**: $p < 0.01$)

"Target only" compares the landmark against the lexical vector of the verb, and "selpref only" compares it to the noun's $subj^{-1}$ preference. For the M&L model, the comparison is to the combined lexical vectors of verb and noun. For our models SELPREF, SELPREF-CUT and SELPREF-POW, we combine the verb's lexical vector with the $subj^{-1}$ preference of the noun. We used a held-out dataset of 10% of the data to optimize the parameters of θ of SELPREF-CUT and n of SELPREF-POW. Vectors with PMI components could model the data, while raw frequency components could not; we report only the former.

We use the same two evaluation scores as M&L: The first score is the average similarity to compatible landmarks (high) and incompatible landmarks (low). The second is Spearman's ρ, a nonparametric correlation coefficient. We compute ρ between individual human similarity scores and our predictions. Based on agreement between human judges, M&L estimate an upper bound ρ of 0.4 for the dataset.

Results and discussion. Table 1 shows the results of Exp. 1 on the test set. In the upper half (BOW), we replicate M&L's main finding that simple component-wise multiplication of the predicate and argument vectors results in a highly significant correlation of $\rho = 0.2$, significantly outperforming both baselines. It is interesting, though, that the $subj^{-1}$ preference itself ("Selpref only") is already highly significantly correlated with the human judgments.

Model	lex. vector	obj^{-1} selpref
SELPREF	0.23 (0.09)	0.88 (0.07)
SELPREF-CUT (10)	0.20 (0.10)	0.72 (0.18)
SELPREF-POW (30)	0.03 (0.08)	0.52 (0.48)

Table 2: Experiment 1: Average similarity (and standard deviation) between the inverse subject preferences of a noun and (left) its lexical vector and (right) inverse object preferences vector (cosine similarity in SYN space)

A comparison of the upper half (BOW) with the lower half (SYN) shows that the dependency-based space generally shows better correlation with human judgements. This corresponds to a beneficial effect of syntactic information found for other applications of semantic spaces (Lin, 1998; Padó and Lapata, 2007).

All instances of the SELPREF model show highly significant correlations. SELPREF and SELPREF-CUT show very similar performance. They do better than both baselines in the BOW space; however, in the cleaner SYN space, their performance is numerically lower than using selectional preferences only ($\rho = 0.13$ vs. 0.16). SELPREF-POW is always significantly better than SELPREF and SELPREF-CUT, and shows the best result of all tested models ($\rho = 0.27$, BOW space). The performance is somewhat lower in the SYN space ($\rho = 0.22$). However, this difference, and the difference to the best M&L model at $\rho = 0.24$, are not statistically significant.

The SVS model computes meaning in context by combining a word's lexical representation with the preference vector of its context. In this, it differs from previous models, including that by M&L, which used what we have been calling "direct combination". So it is important to ask to what extent this difference in method translate to a difference in predictions. We analyzed this by measuring the similarity by the nouns' lexical vectors, used by direct combination methods, and their inverse subject preferences, which SVS uses. The result is shown in the first column in Table 2, computed as mean cosine similarities and standard deviations between noun vectors and selectional preferences. The table shows that these vectors have generally low similarity, which is further reduced by applying cutoff and potentiation. Thus, the predictions of SVS will differ from those of direct combination models like M&L.

A related question is whether syntax-aware vector combination makes a difference: Does the model encode different expectations for different syntactic relations (cf. Example 2)? The second column of Table 2 explores this question by comparing inverse selectional preferences for the subject and object slots. We observe that the similarity is very high for raw preferences, but becomes lower when noise is eliminated. Since the SELPREF-POW model performed best in our evaluation, we read this as evidence that potentiation helps to suppress noise introduced by mis-identified subject and object fillers.

In Experiment 1, all experimental items were verbs, which means that all disambiguation was done through inverse selectional preferences. As inverse selectional preferences are currently largely unexplored, it is interesting to note that the evidence that they provide for the paraphrase task is as strong as that of the context nouns themselves.

6 Exp. 2: Ranking paraphrases

This section reports on a second, more NLP-oriented experiment whose task is to distinguish between appropriate and inappropriate paraphrases on a broader range of constructions.

Dataset. For this experiment, we use the SemEval-1 *lexical substitution* (lexsub) dataset (McCarthy and Navigli, 2007), which contains 10 instances each of 200 target words in sentential contexts, drawn from Sharoff's (2006) English Internet Corpus. Contextually appropriate paraphrases for each instance of each target word were elicited from up to 6 participants. Fig. 4 shows two instances for the verb *to work*. The distribution over paraphrases can be seen as a characterization of the target word's meaning in each context.

Experimental procedure. In this paper, we predict appropriate paraphrases solely on the basis of a single context word that stands in a direct predicate-argument relation to the target word. We extracted all instances from the lexsub test data with such a relation. After parsing all sentences with verbal and nominal targets with Minipar, this resulted in three sets of sentences: (a), target intransitive verbs with noun subjects (V-SUBJ, 48 sentences); (b), target transitive verbs with noun objects (V-OBJ, 213 sent.); and (c), target nouns occurring as objects of verbs (N-OBJ, 102 sent.).[6] Note that since we use only part of the lexical substitution dataset in this experiment, a direct comparison with results from the SemEval task is not possible.

Sentence	Substitutes
By asking people who **work** there, I have since determined that he didn't. (# 2002)	be employed 4; labour 1
Remember how hard your ancestors **worked**. (# 2005)	toil 4; labour 3; task 1

Figure 4: Lexical substitution example items for "work"

As in the original SemEval task, we phrase the task as a ranking problem. For each target word, the paraphrases given for all 10 instances are pooled. The task is to rank the list for each item so that appropriate paraphrases (such as "be employed" for # 2002) rank higher than paraphrases not given (e.g., "toil").

Our model ranks paraphrases by their similarity to the following combinations (Eq. (4)): for V-SUBJ, verb plus the noun's $subj^{-1}$ preferences; for V-OBJ, verb plus the noun's obj^{-1} preferences; and for N-OBJ, the noun plus the verb's obj preferences. Our comparison model, M&L, ranks all paraphrases by their similarity to the direct noun-verb combination.

To avoid overfitting, we consider only the two models that performed optimally in in the SYN space in Experiment 1 (SELPREF-POW with n=30 and M&L). However, since we found that vectors with raw frequency components could model the data, while PMI components could not, we only report the former.

For evaluation, we adopt the SemEval "out of ten" precision metric $\mathrm{P}_{\mathrm{OOT}}$. It uses the model's ten top-ranked paraphrases as its guesses for appropriate paraphrases. Let G_i be the gold paraphrases for item i, M_i the model's top ten paraphrases for i, and $f(s,i)$ the frequency of s as paraphrase for i:

$$\mathrm{P}_{\mathrm{OOT}} = 1/|I| \sum_i \frac{\sum_{s \in M_i \cap G_i} f(s,i)}{\sum_{s \in G_i} f(s,i)} \qquad (8)$$

McCarthy and Navigli propose this metric for the

[6] The specification of this dataset will be made available.

Model	V-SUBJ	V-OBJ	N-OBJ
Target only	47.9	47.4	49.6
Selpref only	54.8	51.4	55.0
M&L	50.3	52.0	53.4
SELPREF-POW, n=30	**63.1**	**55.8**	**56.9**

Table 3: Experiment 2: Mean "out of ten" precision (P_{OOT})

dataset for robustness. Due to the sparsity of paraphrases, a metric that considers fewer guesses leads to artificially low results when a "good" paraphrase was not mentioned by the annotators by chance but is ranked highly by a model.

Results and discussion. Table 6 shows the mean out-of-ten precision for all models. The behavior is fairly uniform across all three datasets. Unsurprisingly, "target only", which uses the same ranking for all instances of a target, yields the worst results.[7]

M&L's direct combination model outperforms "target only" significantly ($p < 0.05$). However, on both the V-SUBJ and the N-OBJ the "selpref only" baseline does better than direct combination. The best results on all datasets are obtained by SELPREF-POW. The difference between SELPREF-POW and the "target only" baseline is highly significant ($p < 0.01$). The difference to M&L's model is significant at $p = 0.05$.

We interpret these results as encouraging evidence for the usefulness of selectional preferences for judging substitutability in context. Knowledge about the selectional preferences of a single context word can already lead to a significant improvement in precision. We find this overall effect even though the word is not informative in all cases. For instance, the subject of item 2002 in Fig. 4, "who", presumably helps little in determining the verb's context-adapted meaning.

It is interesting that the improvement of SELPREF-POW over "selpref only" is smallest for the N-OBJ dataset (1.9% P_{OOT}). N-OBJ uses selectional preferences for nouns that may fill the direct object position, , while V-SUBJ and V-OBJ use inverse selectional preferences for verbs (cf. the two graphs in Fig. 1).

7 Conclusion

In this paper, we have considered semantic space models that can account for the meaning of word occurrences in context. Arguing that existing models do not sufficiently take syntax into account, we have introduced the new structured vector space (SVS) model of word meaning. In addition to a vector representing a word's lexical meaning, it contains vectors representing the word's selectional preferences. These selectional preferences play a central role in the computation of meaning in context.

We have evaluated the SVS model on two datasets on the task of predicting the felicitousness of paraphrases in given contexts. On the M&L dataset, SVS outperforms the state-of-the-art model of M&L, though the difference is not significant. On the Lexical Substitution dataset, SVS significantly outperforms the state-of-the-art. This is especially interesting as the Lexical Substitution dataset, in contrast to the M&L data, uses "realistic" paraphrase candidates that are not necessarily maximally distinct.

The most important limitation of the evaluation that we have given in this paper is that we have only considered single words as context. Our next step will be to integrate information from multiple relations (such as both the subject and object positions of a verb) into the computation of context-specific meaning. Our eventual aim is a model that can give a compositional account of a word's meaning in context, where all words in an expression disambiguate one another according to the relations between them.

We will explore the usability of vector space models of word meaning in NLP applications, formulated as the question of how to perform inferences on them in the context of the Textual Entailment task (Dagan et al., 2006). Paraphrase-based inference rules play a large role in several recent approaches to Textual Entailment (e.g. Szpektor et al (2008)); appropriateness judgments of paraphrases in context, the task of Experiments 1 and 2 above, can be viewed as testing the applicability of these inferences rules.

Acknowledgments. Many thanks for helpful discussion to Jason Baldridge, David Beaver, Dedre Gentner, James Hampton, Dan Jurafsky, Alexander Koller, Brad Love, and Ray Mooney.

[7] "Target only" still does very much better than a random baseline, which performs at 22% P_{OOT}.

References

C. Brockmann, M. Lapata. 2003. Evaluating and combining approaches to selectional preference acquisition. In *Proceedings of EACL*, 27–34.

I. Dagan, O. Glickman, B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges*, Lecture Notes in Computer Science, 177–190. Springer.

K. Erk. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of ACL*, 216–223.

T. Ferretti, C. Gagné, K. McRae. 2003. Thematic role focusing by participle inflections: evidence form conceptual combination. *Journal of Experimental Psychology*, 29(1):118–127.

D. Gildea, D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

D. Hindle, M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

M. Jones, D. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, 114:1–37.

J. J. Katz, J. A. Fodor. 1964. The structure of a semantic theory. In *The Structure of Language*. Prentice-Hall.

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

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

T. Landauer, S. Dumais. 1997. A solution to Platos problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

D. Lin. 1993. Principle-based parsing without overgeneration. In *Proceedings of ACL*, 112–120.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL*, 768–774.

K. Lund, C. Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28:203—208.

C. D. Manning, P. Raghavan, H. Schütze. 2008. *Introduction to Information Retrieval*. CUP.

D. McCarthy, J. Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computational Linguistics*, 29(4):639–654.

D. McCarthy, R. Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, 48–53.

S. McDonald, C. Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of ACL*, 17–24.

S. McDonald, M. Ramscar. 2001. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of CogSci*, 611–616.

K. McRae, M. Spivey-Knowlton, M. Tanenhaus. 1998. Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38:283–312.

K. McRae, M. Hare, J. Elman, T. Ferretti. 2005. A basis for generating expectancies for verbs from nouns. *Memory and Cognition*, 33(7):1174–1184.

J. Mitchell, M. Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, 236–244.

A. Moschitti, S. Quarteroni. 2008. Kernels on linguistic structures for answer extraction. In *Proceedings of ACL*, 113–116, Columbus, OH.

S. Narayanan, D. Jurafsky. 2002. A Bayesian model predicts human parse preference and reading time in sentence processing. In *Proceedings of NIPS*, 59–65.

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

U. Padó, F. Keller, M. W. Crocker. 2006. Combining syntax and thematic fit in a probabilistic model of sentence processing. In *Proceedings of CogSci*, 657–662.

S. Padó, U. Padó, K. Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of EMNLP/CoNLL*, 400–409.

P. Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61:127–159.

G. Salton, A. Wang, C. Yang. 1975. A vector-space model for information retrieval. *Journal of the American Society for Information Science*, 18:613–620.

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

S. Sharoff. 2006. Open-source corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics*, 11(4):435–462.

P. Smolensky. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216.

I. Szpektor, I. Dagan, R. Bar-Haim, J. Goldberger. 2008. Contextual preferences. In *Proceedings of ACL*, 683–691, Columbus, OH.

Y. Wilks. 1975. Preference semantics. In *Formal Semantics of Natural Language*. CUP.

A. Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceeedings of COLING*, 947–953.