Measuring Distributional Similarity in Context

Georgiana Dinu
Department of Computational Linguistics
Saarland University
Saarbrücken, Germany
dinu@coli.uni-sb.de

Mirella Lapata
School of Informatics
University of Edinburgh
Edinburgh, UK
mlap@inf.ed.ac.uk

Abstract

The computation of meaning similarity as operationalized by vector-based models has found widespread use in many tasks ranging from the acquisition of synonyms and paraphrases to word sense disambiguation and textual entailment. Vector-based models are typically directed at representing words in isolation and thus best suited for measuring similarity out of context. In his paper we propose a probabilistic framework for measuring similarity in context. Central to our approach is the intuition that word meaning is represented as a probability distribution over a set of latent senses and is modulated by context. Experimental results on lexical substitution and word similarity show that our algorithm outperforms previously proposed models.

1 Introduction

The computation of meaning similarity as operationalized by vector-based models has found widespread use in many tasks within natural language processing (NLP). These range from the acquisition of synonyms (Grefenstette, 1994; Lin, 1998) and paraphrases (Lin and Pantel, 2001) to word sense disambiguation (Schuetze, 1998), textual entailment (Clarke, 2009), and notably information retrieval (Salton et al., 1975).

The popularity of vector-based models lies in their unsupervised nature and ease of computation. In their simplest incarnation, these models represent the meaning of each word as a point in a high-dimensional space, where each component corresponds to some co-occurring contextual element (Landauer and Dumais, 1997; McDonald, 2000; Lund and Burgess, 1996). The advantage of taking such a geometric approach is that the similarity of word meanings can be easily quantified by measuring their distance in the vector space, or the cosine of the angle between them.

Vector-based models do not explicitly identify the different senses of words and consequently represent their meaning invariably (i.e., irrespective of co-occurring context). Consider for example the adjective *heavy* which we may associate with the general meaning of "dense" or "massive". However, when attested in context, *heavy* may refer to an overweight person (e.g., *She is short and heavy but she has a heart of gold.*) or an excessive cannabis user (e.g., *Some heavy users develop a psychological dependence on cannabis.*).

Recent work addresses this issue *indirectly* with the development of specialized models that represent word meaning in context (Mitchell and Lapata, 2008; Erk and Padó, 2008; Thater et al., 2009). These methods first extract typical co-occurrence vectors representing a *mixture* of senses and then use vector operations to either obtain contextualized representations of a target word (Erk and Padó, 2008) or a representation for a set of words (Mitchell and Lapata, 2009).

In this paper we propose a probabilistic framework for representing word meaning and measuring similarity in context. We model the meaning of isolated words as a probability distribution over a set of latent senses. This distribution reflects the *a priori*, out-of-context likelihood of each sense. Because sense ambiguity is taken into account *directly* in the

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1162–1172,
MIT, Massachusetts, USA, 9-11 October 2010.

vector construction process, contextualized meaning can be modeled naturally as a change in the original sense distribution. We evaluate our approach on word similarity (Finkelstein et al., 2002) and lexical substitution (McCarthy and Navigli, 2007) and show improvements over competitive baselines.

In the remainder of this paper we give a brief overview of related work, emphasizing vector-based approaches that compute word meaning in context (Section 2). Next, we present our probabilistic framework and different instantiations thereof (Sections 3 and 4). Finally, we discuss our experimental results (Sections 5 and 6) and conclude the paper with future work.

2 Related work

Vector composition methods construct representations that go beyond individual words (e.g., for phrases or sentences) and thus by default obtain word meanings in context. Mitchell and Lapata (2008) investigate several vector composition operations for representing short sentences (consisting of intransitive verbs and their subjects). They show that models performing point-wise multiplication of component vectors outperform earlier proposals based on vector addition (Landauer and Dumais, 1997; Kintsch, 2001). They argue that multiplication approximates the intersection of the meaning of two vectors, whereas addition their union. Mitchell and Lapata (2009) further show that their models yield improvements in language modeling.

Erk and Padó (2008) employ selectional preferences to contextualize occurrences of target words. For example, the meaning of a verb in the presence of its object is modeled as the multiplication of the verb's vector with the vector capturing the inverse selectional preferences of the object; the latter are computed as the centroid of the verbs that occur with this object. Thater et al. (2009) improve on this model by representing verbs in a second order space, while the representation for objects remains first order. The meaning of a verb boils down to restricting its vector to the features active in the argument noun (i.e., dimensions with value larger than zero).

More recently, Reisinger and Mooney (2010) present a method that uses clustering to produce multiple sense-specific vectors for each word. Specifically, a word's contexts are clustered to produce groups of similar context vectors. An average prototype vector is then computed separately for each cluster, producing a set of vectors for each word. These cluster vectors can be used to determine the semantic similarity of both isolated words and words in context. In the second case, the distance between prototypes is weighted by the probability that the context belongs to the prototype's cluster. Erk and Padó (2010) propose an exemplar-based model for capturing word meaning in context. In contrast to the prototype-based approach, no clustering takes place, it is assumed that there are as many senses as there are instances. The meaning of a word in context is the set of exemplars most similar to it.

Unlike Reisinger and Mooney (2010) and Erk and Padó (2010) our model is probabilistic (we represent word meaning as a distribution over a set of latent senses), which makes it easy to integrate and combine with other systems via mixture or product models. More importantly, our approach is conceptually simpler as we use a single vector representation for isolated words as well as for words in context. A word's different meanings are simply modeled as changes in its sense distribution. We should also point out that our approach is not tied to a specific sense induction method and can be used with different variants of vector-space models.

3 Meaning Representation in Context

In this section we first describe how we represent the meaning of individual words and then move on to discuss our model of inducing meaning representations in context.

Observed Representations Most vector space models in the literature perform computations on a co-occurrence matrix where each row represents a *target* word, each column a document or another neighboring word, and each entry their co-occurrence frequency. The raw counts are typically mapped into the components of a vector in some space using for example conditional probability, the log-likelihood ratio or tf-idf weighting. Under this representation, the similarity of word meanings can be easily quantified by measuring their distance in the vector space, the cosine of the angle between them, or their scalar product.

Our model assumes the same type of input data, namely a co-occurrence matrix, where rows correspond to *target* words and columns to *context features* (e.g., co-occurring neighbors). Throughout this paper we will use the notation t_i with $i : 1..I$ to refer to a target word and c_j with $j : 1...J$ to refer to context features. A cell (i, j) in the matrix represents the frequency of occurrence of target t_i with context feature c_j over a corpus.

Meaning Representation over Latent Senses We further assume that the target words t_i $i : 1...I$ found in a corpus share a global set of meanings or senses $Z = \{z_k|k : 1...K\}$. And therefore the meaning of individual target words can be described as a distribution over this set of senses. More formally, a target t_i is represented by the following vector:

$$\mathbf{v(t_i)} = (\mathbf{P(z_1|t_i)}, ..., \mathbf{P(z_K|t_i)}) \quad (1)$$

where component $P(z_1|t_i)$ is the probability of sense z_1 given target word t_i, component $P(z_2|t_i)$ the probability of sense z_2 given t_i and so on.

The intuition behind such a representation is that a target word can be described by a set of core meanings and by the frequency with which these are attested. Note that the representation in (1) is not fixed but parametrized with respect to an input corpus (i.e., it only reflects word usage as attested in that corpus). The senses $z_1 \ldots z_K$ are latent and can be seen as a means of reducing the dimensionality of the original co-occurrence matrix.

Analogously, we can represent the meaning of a target word given a context feature as:

$$\mathbf{v(t_i, c_j)} = (\mathbf{P(z_1|t_i, c_j)}, ..., \mathbf{P(z_K|t_i, c_j)}) \quad (2)$$

Here, target t_i is again represented as a distribution over senses, but is now modulated by a specific context c_j which reflects actual word usage. This distribution is more "focused" compared to (1); the context helps disambiguate the meaning of the target word, and as a result fewer senses will share most of the probability mass.

In order to create the context-aware representations defined in (2) we must estimate the probabilities $P(z_k|t_i, c_j)$ which can be factorized as the product of $P(t_i, z_k)$, the joint probability of target t_i and latent sense z_k, and $P(c_j|z_k, t_i)$, the conditional probability of context c_j given target t_i and sense z_k:

$$P(z_k|t_i, c_j) = \frac{P(t_i, z_k)P(c_j|z_k, t_i)}{\sum_k P(t_i, z_k)P(c_j|z_k, t_i)} \quad (3)$$

Problematically, the term $P(c_j|z_k, t_i)$ is difficult to estimate since it implies learning a total number of $K \times I$ J-dimensional distributions. We will therefore make the simplifying assumption that target words t_i and context features c_j are conditionally independent given sense z_k:

$$P(z_k|t_i, c_j) \approx \frac{P(z_k|t_i)P(c_j|z_k)}{\sum_k P(z_k|t_i)P(c_j|z_k)} \quad (4)$$

Although not true in general, the assumption is relatively weak. We do not assume that words and context features occur independently of each other, but only that they are generated independently given an assigned meaning. A variety of latent variable models can be used to obtain senses $z_1 \ldots z_K$ and estimate the distributions $P(z_k|t_i)$ and $P(c_j|z_k)$; we give specific examples in Section 4.

Note that we abuse terminology here, as the senses our models obtain are not lexicographic meaning distinctions. Rather, they denote coarse-grained senses or more generally topics attested in the document collections our model is trained on. Furthermore, the senses are not word-specific but global (i.e., shared across all words) and modulated either within or out of context probabilistically via estimating $P(z_k|t_i, c_j)$ and $P(z_k|t_i)$, respectively.

4 Parametrizations

The general framework outlined above can be parametrized with respect to the input co-occurrence matrix and the algorithm employed for inducing the latent structure. Considerable latitude is available when creating the co-occurrence matrix, especially when defining its columns, i.e., the linguistic contexts a target word is attested with. These contexts can be a small number of words surrounding the target word (Lund and Burgess, 1996; Lowe and McDonald, 2000), entire paragraphs, documents (Salton et al., 1975; Landauer and Dumais, 1997) or even syntactic dependencies (Grefenstette, 1994; Lin, 1998; Padó and Lapata, 2007).

Analogously, a number of probabilistic models can be employed to induce the latent senses. Examples include Probabilistic Latent Semantic Analysis (PLSA, Hofmann (2001)), Probabilistic Principal Components Analysis (Tipping and Bishop, 1999), non-negative matrix factorization (NMF, Lee and Seung (2000)), and latent Dirichlet allocation (LDA, Blei et al. (2003)). We give a more detailed description of the latter two models as we employ them in our experiments.

Non-negative Matrix Factorization Non-negative matrix factorization algorithms approximate a non-negative input matrix V by two non-negative factors W and H, under a given loss function. W and H are reduced-dimensional matrices and their product can be regarded as a compressed form of the data in V:

$$V_{I,J} \approx W_{I,K} H_{K,J} \tag{5}$$

where W is a basis vector matrix and H is an encoded matrix of the basis vectors in equation (5). Several loss functions are possible, such as mean squared error and Kullback-Leibler (KL) divergence. In keeping with the formulation in Section 3 we opt for a probabilistic interpretation of NMF (Gaussier and Goutte, 2005; Ding et al., 2008) and thus minimize the KL divergence between WH and V.

$$\min \sum_{i,j} (V_{i,j} \log \frac{V_{i,j}}{WH_{i,j}} - V_{i,j} + WH_{i,j}) \tag{6}$$

Specifically, we interpret matrix V as $V_{ij} = P(t_i, c_j)$, and matrices W and H as $P(t_i, z_k)$ and $P(c_j|z_k)$, respectively. We can also obtain the following more detailed factorization: $P(t_i, c_j) = \sum_k P(t_i)P(z_k|t_i)P(c_j|z_k)$.

Le WH denote the factors in a NMF decomposition of an input matrix V and B be a diagonal matrix with $B_{kk} = \sum_j H_{kj}$. $B^{-1}H$ gives a row-normalized version of H. Similarly, given matrix WB, we can define a diagonal matrix A, with $A_{ii} = \sum_k (WB)_{ik}$. $A^{-1}WB$ row-normalizes matrix WB. The factorization WH can now be rewritten as:

$$WH = AA^{-1}WBB^{-1}H = A(A^{-1}WB)(B^{-1}H)$$

which allows us to interpret A as $P(t_i)$, $A^{-1}WB$ as $P(z_k|t_i)$ and $B^{-1}H$ as $P(c_j|z_k)$. These interpretations are valid since the rows of $A^{-1}WB$ and of $B^{-1}H$ sum to 1, matrix A is diagonal with trace 1 because elements in WH sum to 1, and all entries are non-negative.

Latent Dirichlet Allocation LDA (Blei et al., 2003) is a probabilistic model of text generation. Each document d is modeled as a distribution over K topics, which are themselves characterized by distributions over words. The individual words in a document are generated by repeatedly sampling a topic according to the topic distribution and then sampling a single word from the chosen topic.

More formally, we first draw the mixing proportion over topics θ_d from a Dirichlet prior with parameters α. Next, for each of the N_d words w_{dn} in document d, a topic z_{dn} is first drawn from a multinomial distribution with parameters θ_{dn}. The probability of a word token w taking on value i given that topic $z = j$ is parametrized using a matrix β with $b_{ij} = P(w = i|z = j)$. Integrating out θ_d's and z_{dn}'s, gives $P(D|\alpha, \beta)$, the probability of a corpus (or document collection):

$$\prod_{d=1}^{M} \int P(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} P(z_{dn}|\theta_d) P(w_{dn}|z_{dn}, \beta) \right) d\theta_d$$

The central computational problem in topic modeling is to obtain the posterior distribution $P(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$ of the hidden variables $\mathbf{z} = (z_1, z_2, \ldots, z_N)$. given a document $\mathbf{w} = (w_1, w_2, \ldots, w_N)$. Although this distribution is intractable in general, a variety of approximate inference algorithms have been proposed in the literature. We adopt the Gibbs sampling procedure discussed in Griffiths and Steyvers (2004). In this model, $P(w = i|z = j)$ is also a Dirichlet mixture (denoted ϕ) with symmetric priors (denoted β).

We use LDA to induce senses of target words based on context words, and therefore each row t_i in the input matrix transforms into a document. The frequency of t_i occurring with context feature c_j is the number of times word c_j is encountered in the "document" associated with t_i. We train the LDA model on this data to obtain the θ and ϕ distribu-

tions. θ gives the sense distributions of each target t_i: $\theta_{ik} = P(z_k|t_i)$ and ϕ the context-word distribution for each sense z_k: $\phi_{kj} = P(c_j|z_k)$.

5 Experimental Set-up

In this section we discuss the experiments we performed in order to evaluate our model. We describe the tasks on which it was applied, the corpora used for model training and our evaluation methodology.

Tasks The probabilistic model presented in Section 3 represents words via a set of induced senses. We experimented with two types of semantic space based on NMF and LDA and optimized parameters for these models on a word similarity task. The latter involves judging the similarity $sim(t_i, t'_i) = sim(v(t_i), v(t'_i))$ of words t_i and t'_i out of context, where $v(t_i)$ and $v(t'_i)$ are obtained from the output of NMF or LDA, respectively. In our experiments we used the data set of Finkelstein et al. (2002). It contains 353 pairs of words and their similarity scores as perceived by human subjects.

The contextualized representations were next evaluated on lexical substitution (McCarthy and Navigli, 2007). The task requires systems to find appropriate substitutes for target words occurring in context. Typically, systems are given a set of substitutes, and must produce a ranking such that appropriate substitutes are assigned a higher rank compared to non-appropriate ones. We made use of the SemEval 2007 Lexical Substitution Task benchmark data set. It contains 200 target words, namely nouns, verbs, adjectives and adverbs, each of which occurs in 10 distinct sentential contexts. The total set contains 2,000 sentences. Five human annotators were asked to provide substitutes for these target words. Table 1 gives an example of the adjective *still* and its substitutes.

Following Erk and Padó (2008), we pool together the total set of substitutes for each target word. Then, for each instance the model has to produce a ranking for the total substitute set. We rank the candidate substitutes based on the similarity of the contextualized target and the out-of-context substitute, $sim(v(t_i, c_j), v(t'_i))$, where t_i is the target word, c_j a context word and t'_i a substitute. Contextualizing just one of the words brings higher discriminative power to the model rather than performing comparisons with the target and its substitute embedded in an identical context (see also Thater et al. (2010) for a similar observation).

Sentences	Substitutes
It is important to apply the herbicide on a *still* day, because spray drift can kill non-target plants.	calm (5) not-windy (1) windless (1)
A movie is a visual document comprised of a series of *still* images.	motionless (3) unmoving (2) fixed (1) stationary (1) static (1)

Table 1: Lexical substitution data example for the adjective *still*; numbers in parentheses indicate the frequency of the substitute.

Model Training All the models we experimented with use identical input data, i.e., a bag-of-words matrix extracted from the GigaWord collection of news text. Rows in this matrix are target words and columns are their co-occurring neighbors, within a symmetric window of size 5. As context words, we used a vocabulary of the 3,000 most frequent words in the corpus.[1]

We implemented the classical NMF factorization algorithm described in Lee and Seung (2000). The input matrix was normalized so that all elements summed to 1. We experimented with four dimensions K: $[600 - 1000]$ with step size 200. We ran the algorithm for 150 iterations to obtain factors W and H which we further processes as described in Section 4 to obtain the desired probability distributions. Since the only parameter of the NMF model is the factorization dimension K, we performed two independent runs with each K value and averaged their predictions.

The parameters for the LDA model are the number of topics K and Dirichlet priors α and β. We experimented with topics K: $[600 - 1400]$, again with step size 200. We fixed β to 0.01 and tested two values for α: $\frac{2}{K}$ (Porteous et al., 2008) and $\frac{50}{K}$ (Griffiths and Steyvers, 2004). We used Gibbs sampling on the "document collection" obtained from the input matrix and estimated the sense distributions as described in Section 4. We ran the chains for 1000 iter-

[1]The GigaWord corpus contains 1.7B words; we scale down all the counts by a factor of 70 to speed up the computation of the LDA models. All models use this reduced size input data.

ations and averaged over five iterations $[600-1000]$ at lag 100 (we observed no topic drift).

We measured similarity using the scalar product, cosine, and inverse Jensen-Shannon (IJS) divergence (see (7), (8), and (9), respectively):

$$\text{sp}(v,w) =< v,w >= \sum_i v_i w_i \quad (7)$$

$$\cos(v,w) = \frac{\langle v,w \rangle}{||v||\,||w||} \quad (8)$$

$$\text{IJS}(v,w) = \frac{1}{\text{JS(v, w)}} \quad (9)$$

$$\text{JS(v, w)} = \frac{1}{2}\text{KL}(v|m) + \frac{1}{2}\text{KL}(w|m) \quad (10)$$

where m is a shorthand for $\frac{1}{2}(v+w)$ and KL the Kullback-Leibler divergence, $\text{KL}(v|w) = \sum_i v_i log(\frac{v_i}{w_i})$.

Among the above similarity measures, the scalar product has the most straightforward interpretation as the probability of two targets sharing a common meaning (i.e., the sum over all possible meanings). The scalar product assigns 1 to a pair of identical vectors if and only if $P(z_i) = 1$ for some i and $P(z_j) = 0, \forall j \neq i$. Thus, only fully disambiguated words receive a score of 1. Beyond similarity, the measure also reflects how "focused" the distributions in question are, as very ambiguous words are unlikely to receive high scalar product values.

Given a set of context words, we contextualize the target using one context word at a time and compute the overall similarity score by multiplying the individual scores.

Baselines Our baseline models for measuring similarity out of context are Latent Semantic Analysis (Landauer and Dumais, 1997) and a simple semantic space without any dimensionality reduction.

For LSA, we computed the $U\Sigma V$ SVD decomposition of the original matrix to rank $k = 1000$. Any decomposition of lower rank can be obtained from this by setting rows and columns to 0. We evaluated decompositions to ranks K: $[200 - 1000]$, at each 100 step. Similarity computations were performed in the lower rank approximation matrix $U\Sigma V$, as originally proposed in Deerwester et al. (1990), and in matrix U which maps the words into the concept space. It is common to compute SVD decompositions on matrices to which prior weighting schemes have been applied. We experimented with tf-idf weighting and line normalization.

Our second baseline, the simple semantic space, was based on the original input matrix on which we applied several weighting schemes such as pointwise mutual information, tf-idf, and line normalization. Again, we measured similarity using cosine, scalar product and inverse JS divergence. In addition, we also experimented with Lin's (1998) similarity measure:

$$\text{lin}(v,w) = \frac{\sum_{i \in I(v) \cap I(w)} (v_i + w_i)}{\sum_{i \in I(v)} v_i + \sum_{l \in I(w)} w_i} \quad (11)$$

where the values in v and w are point-wise mutual information, and $I(\cdot)$ gives the indices of positive values in a vector.

Our baselines for contextualized similarity were vector addition and vector multiplication which we performed using the simple semantic space (Mitchell and Lapata, 2008) and dimensionality reduced representations obtained from NMF and LDA. To create a ranking of the candidate substitutes we compose the vector of the target with its context and compare it with each substitute vector. Given a set of context words, we contextualize the target using each context word at a time and multiply the individual scores.

Evaluation Method For the word similarity task we used correlation analysis to examine the relationship between the human ratings and their corresponding vector-based similarity values. We report Spearman's ρ correlations between the similarity values provided by the models and the mean participant similarity ratings in the Finkelstein et al. (2002) data set. For the lexical substitution task, we compare the system ranking with the gold standard ranking using Kendall's τ_b rank correlation (which is adjusted for tied ranks). For all contextualized models we defined the context of a target word as the words occurring within a symmetric context window of size 5. We assess differences between models using stratified shuffling (Yeh, 2000).[2]

[2]Given two system outputs, the null hypothesis (i.e., that the two predictions are indistinguishable) is tested by randomly mixing the individual instances (in our case sentences) of the two outputs. We ran a standard number of 10000 iterations.

Model	Spearman ρ
SVS	38.35
LSA	49.43
NMF	**52.99**
LDA	**53.39**
LSA_{MIX}	49.76
NMF_{MIX}	51.62
LDA_{MIX}	51.97

Table 2: Results on out of context word similarity using a simple co-occurrence based vector space model (SVS), latent semantic analysis, non-negative matrix factorization and latent Dirichlet allocation as individual models with the best parameter setting (LSA, NMF, LDA) and as mixtures (LSA_{MIX}, NMF_{MIX}, LDA_{MIX}).

6 Results

Word Similarity Our results on word similarity are summarized in Table 2. The simple co-occurrence based vector space (SVS) performed best with tf-idf weighting and the cosine similarity measure. With regard to LSA, we obtained best results with initial line normalization of the matrix, $K = 600$ dimensions, and the scalar product similarity measure while performing computations in matrix U. Both NMF and LDA models are generally better with a larger number of senses. NMF yields best performance with $K = 1000$ dimensions and the scalar product similarity measure. The best LDA model also uses the scalar product, has $K = 1200$ topics, and α set to $\frac{50}{K}$.

Following Reisinger and Mooney (2010), we also evaluated mixture models that combine the output of models with varying parameter settings. For both NMF and LDA we averaged the similarity scores returned by all runs. For comparison, we also present an LSA mixture model over the (best) middle interval K values. As can be seen, the LSA model improves slightly, whereas NMF and LDA perform worse than their best individual models.[3] Overall, we observe that NMF and LDA yield significantly ($p < 0.01$) better correlations than LSA and the simple semantic space, both as individual models and as mixtures.

[3]It is difficult to relate our results to Reisinger and Mooney (2010), due to differences in the training data and the vector representations it gives rise to. As a comparison, a baseline configuration with tf-idf weighting and the cosine similarity measure yields a correlation of 0.38 with our data and 0.49 in Reisinger and Mooney (2010).

Model	Kendall's τ_b
SVS	11.05
Add-SVS	12.74
Add-NMF	12.85
Add-LDA	12.33
Mult-SVS	14.41
Mult-NMF	13.20
Mult-LDA	12.90
Cont-NMF	14.95
Cont-LDA	13.71
Cont-NMF_{MIX}	**16.01**
Cont-LDA_{MIX}	**15.53**

Table 3: Results on lexical substitution using a simple semantic space model (SVS), additive and multiplicative compositional models with vector representations based on co-occurrences (Add-SVS, Mult-SVS), NMF (Add-NMF, Mult-NMF), and LDA (Add-LDA, Mult-LDA) and contextualized models based on NMF and LDA with the best parameter setting (Cont-NMF, Cont-LDA) and as mixtures (Cont-NMF_{MIX}, Cont-LDA_{MIX}).

Lexical Substitution Our results on lexical substitution are shown in Table 3. As a baseline we also report the performance of the simple semantic space that does not use any contextual information. This model returns the same ranking of the substitute candidates for each instance, based solely on their similarity with the target word. This is a relatively competitive baseline as observed by Erk and Padó (2008) and Thater et al. (2009).

We report results with contextualized NMF and LDA as individual models (the best word similarity settings) and as mixtures (as described above). These are in turn compared against additive and multiplicative compositional models. We implemented an additive model with pmi weighting and Lin's similarity measure which is defined in an additive fashion. The multiplicative model uses tf-idf weighting and cosine similarity, which involves multiplication of vector components. Other combinations of weighting schemes and similarity measures delivered significantly lower results. We also report results for these models when using the NMF and LDA reduced representations.

Model	Adv	Adj	Noun	Verb
SVS	22.47	14.38	09.52	7.98
Add-SVS	22.79	14.56	11.59	10.00
Mult-SVS	22.85	16.37	13.59	11.60
Cont-NMF$_{\text{MIX}}$	**26.13**	**17.10**	15.16	**14.18**
Cont-LDA$_{\text{MIX}}$	21.21	16.00	**16.31**	13.67

Table 4: Results on lexical substitution for different parts of speech with a simple semantic space model (SVS), two compositional models (Add-SVS, Mult-SVS), and contextualized mixture models with NMF and LDA (Cont-NMF$_{\text{MIX}}$, Cont-LDA$_{\text{MIX}}$), using Kendall's τ_b correlation coefficient.

All models significantly ($p < 0.01$) outperform the context agnostic simple semantic space (see SVS in Table 3). Mixture NMF and LDA models are significantly better than all variants of compositional models ($p < 0.01$); the individual models are numerically better, however the difference is not statistically significant. We also find that the multiplicative model using a simple semantic space (Mult-SVS) is the best performing compositional model, thus corroborating the results of Mitchell and Lapata (2009). Interestingly, dimensionality compositional models. This indicates that the better results we obtain are due to the probabilistic formulation of our contextualized model as a whole rather than the use of NMF or LDA. Finally, we observe that the Cont-NMF model is slightly better than Cont-LDA, however the difference is not statistically significant.

To allow comparison with previous results reported on this data set, we also used the Generalized Average Precision (GAP, Kishida (2005)) as an evaluation measure. GAP takes into account the order of candidates ranked correctly by a hypothetical system, whereas average precision is only sensitive to their relative position. The best performing models are Cont-NMF$_{\text{MIX}}$ and Cont-LDA$_{\text{MIX}}$ obtaining a GAP of 42.7% and 42.9%, respectively. Erk and Padó (2010) report a GAP of 38.6% on this data set with their best model.

Table 4 shows how the models perform across different parts of speech. While verbs and nouns seem to be most difficult, we observe higher gains from the use of contextualized models. Cont-LDA$_{\text{MIX}}$ obtains approximately 7% absolute gain for nouns and Cont-NMF$_{\text{MIX}}$ approximately 6% for verbs. All contextualized models obtain smaller improvements for adjectives. For adverbs most models do not improve over the no-context setting, with the exception Cont-NMF$_{\text{MIX}}$.

Senses	Word Distributions
TRAFFIC (0.18)	*road, traffic, highway, route, bridge*
MUSIC (0.04)	*music, song, rock, band, dance, play*
FAN (0.04)	*crowd, fan, people, wave, cheer, street*
VEHICLE (0.04)	*car, truck, bus, train, driver, vehicle*

Table 5: Induced senses of *jam* and five most likely words given these senses using an LDA model; sense probabilities are shown in parentheses.

Finally, we also qualitatively examined how the context words influence the sense distributions of target words using examples from the lexical substitution dataset and the output of an individual Cont-LDA model. In many cases, a target word starts with a distribution spread over a larger number of senses, while a context word shifts this distribution to one majority sense. Consider, for instance, the target noun *jam* in the following sentence:

(1) With their transcendent, improvisational jams and Mayan-inspired sense of a higher, metaphysical purpose, the band's music delivers a spiritual sustenance that has earned them a very devoted core following.

Table 5 shows the out-of-context senses activated for *jam* together with the five most likely words associated with them.[4] Sense probabilities are also shown in parentheses. As can be seen, initially two traffic-related and two music-related senses are activated, however with low probabilities. In the presence of the context word *band*, we obtain a much more "focused" distribution, in which the MUSIC sense has 0.88 probability. The system ranks *riff* and *gig* as the most likely two substitutes for *jam*. The gold annotation also lists *session* as a possible substitute.

In a large number of cases, the target is only partially disambiguated by a context word and this is also reflected in the resulting distribution. An ex-

[4]Sense names are provided by the authors in an attempt to best describe the clusters (i.e., topics for LDA) to which words are assigned.

ample is the word *bug* which initially has a distribution triggering the SOFTWARE (0.09, *computer, software, microsoft, windows*) and DISEASE (0.06, *disease, aids, virus, cause*) senses. In the context of *client, bug* remains ambiguous between the senses SECRET-AGENCY (0.34, *agent, secret, intelligence, FBI*)) and SOFTWARE (0.29):

(2) We wanted to give our client more than just a list of bugs and an invoice — we wanted to provide an audit trail of our work along with meaningful productivity metrics.

There are also cases where the contextualized distributions are not correct, especially when senses are domain specific. An example is the word *function* occurring in its mathematical sense with the context word *distribution*. However, the senses that are triggered by this pair all relate to the "service" sense of *function*. This is a consequence of the newspaper corpus we use, in which the mathematical sense of *function* is rare. We also see several cases where the target word and one of the context words are assigned senses that are locally correct, but invalid in the larger context. In the following example:

(3) Check the shoulders so it hangs well, stops at hips or below, and make sure the pants are long enough.

The pair *(check, shoulder)* triggers senses INJURY (0.81, *injury, left, knee, shoulder*) and BALL-SPORTS (0.10, *ball, shot, hit, throw*). However, the sentential context ascribes a meaning that is neither related to injury nor sports. This suggests that our models could benefit from more principled context feature aggregation.

Generally, verbs are not as good context words as nouns. To give an example, we often encounter the pair *(let, know)*, used in the common "inform" meaning. The senses we obtain for this pair, are, however, rather uninformative general verb classes: {*see, know, think, do*} (0.57) and {*go, say, do, can*} (0.20). This type of error can be eliminated in a space where context features are designed to best reflect the properties of the target words.

7 Conclusions

In this paper we have presented a general framework for computing similarity in context. Key in this framework is the representation of word meaning as a distribution over a set of global senses where contextualized meaning is modeled as a change in this distribution. The approach is conceptually simple, the same vector representation is used for isolated words and words in context without being tied to a specific sense induction method or type of semantic space.

We have illustrated two instantiations of this framework using non-negative matrix factorization and latent Dirichlet allocation for inducing the latent structure, and shown experimentally that they outperform previously proposed methods for measuring similarity in context. Furthermore, both of them benefit from mixing model predictions over a set of different parameter choices, thus making parameter tuning redundant.

The directions for future work are many and varied. Conceptually, we have defined our model in an asymmetric fashion, i.e., by stipulating a difference between target words and contextual features. However, in practice, we used vector representations that do not distinguish the two: target words and contextual features are both words. This choice was made to facilitate comparisons with the popular bag-of-words vector space models. However, differentiating target from context representations may be beneficial particularly when the similarity computations are embedded within specific tasks such as the acquisition of paraphrases, the recognition of entailment relations, and thesaurus construction. Also note that our model currently contextualizes target words with respect to individual contexts. Ideally, we would like to compute the collective influence of several context words on the target. We plan to further investigate how to select or to better aggregate the entire set of features extracted from a context.

Acknowledgments The authors acknowledge the support of the DFG (Dinu; International Research Training Group "Language Technology and Cognitive Systems") and EPSRC (Lapata; grant GR/T04540/01).

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

Daoud Clarke. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119, Athens, Greece.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Chris Ding, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927.

Katrin Erk and Sabastian 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*, pages 897–906, Honolulu, Hawaii.

Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, Uppsala, Sweden.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Eric Gaussier and Cyril Goutte. 2005. Relation between PLSA and NMF and implications. In *Proceedings of the 28th Annual international ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, New York, NY.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235.

Thomas Hofmann. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 41(2):177–196.

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*.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):342–360.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the joint Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics*, pages 768–774, Montréal, Canada.

Will Lowe and Scott McDonald. 2000. The direct route: Mediated priming in semantic space. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 675–680, Philadelphia, PA.

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

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 Task 10: English Lexical Substitution Task. In *Proceedings of SemEval*, pages 48–53, Prague, Czech Republic.

Scott McDonald. 2000. *Environmental Determinants of Lexical Processing Effort*. Ph.D. thesis, University of Edinburgh.

Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, Columbus, Ohio.

Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 430–439, Suntec, Singapore.

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

Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Fast collapsed gibbs sampling for latent Dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, New York, NY.

Joseph Reisinger and Raymond J. Mooney. 2010. 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*, pages 109–117, Los Angeles, California.

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

Hinrich Schuetze. 1998. Automatic word sense discrimination. *Journal of Computational Linguistics*, 24:97–123.

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

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

Michael E. Tipping and Chris M. Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 947–953, Saarbrücken, Germany.