Improving Word Representations via Global Context and Multiple Word Prototypes

Eric H. Huang, Richard Socher*, Christopher D. Manning, Andrew Y. Ng
Computer Science Department, Stanford University, Stanford, CA 94305, USA
{ehhuang,manning,ang}@stanford.edu, *richard@socher.org

Abstract

Unsupervised word representations are very useful in NLP tasks both as inputs to learning algorithms and as extra word features in NLP systems. However, most of these models are built with only local context and one representation per word. This is problematic because words are often polysemous and global context can also provide useful information for learning word meanings. We present a new neural network architecture which 1) learns word embeddings that better capture the semantics of words by incorporating both local and global document context, and 2) accounts for homonymy and polysemy by learning multiple embeddings per word. We introduce a new dataset with human judgments on pairs of words in sentential context, and evaluate our model on it, showing that our model outperforms competitive baselines and other neural language models. [1]

1 Introduction

Vector-space models (VSM) represent word meanings with vectors that capture semantic and syntactic information of words. These representations can be used to induce similarity measures by computing distances between the vectors, leading to many useful applications, such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002) and question answering (Tellex et al., 2003).

[1]The dataset and word vectors can be downloaded at http://ai.stanford.edu/~ehhuang/.

Despite their usefulness, most VSMs share a common problem that each word is only represented with one vector, which clearly fails to capture homonymy and polysemy. Reisinger and Mooney (2010b) introduced a multi-prototype VSM where word sense discrimination is first applied by clustering contexts, and then prototypes are built using the contexts of the sense-labeled words. However, in order to cluster accurately, it is important to capture both the syntax and semantics of words. While many approaches use local contexts to disambiguate word meaning, global contexts can also provide useful topical information (Ng and Zelle, 1997). Several studies in psychology have also shown that global context can help language comprehension (Hess et al., 1995) and acquisition (Li et al., 2000).

We introduce a new neural-network-based language model that distinguishes and uses both local and global context via a joint training objective. The model learns word representations that better capture the semantics of words, while still keeping syntactic information. These improved representations can be used to represent contexts for clustering word instances, which is used in the multi-prototype version of our model that accounts for words with multiple senses.

We evaluate our new model on the standard WordSim-353 (Finkelstein et al., 2001) dataset that includes human similarity judgments on pairs of words, showing that combining both local and global context outperforms using only local or global context alone, and is competitive with state-of-the-art methods. However, one limitation of this evaluation is that the human judgments are on pairs

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 873–882,
Jeju, Republic of Korea, 8-14 July 2012.

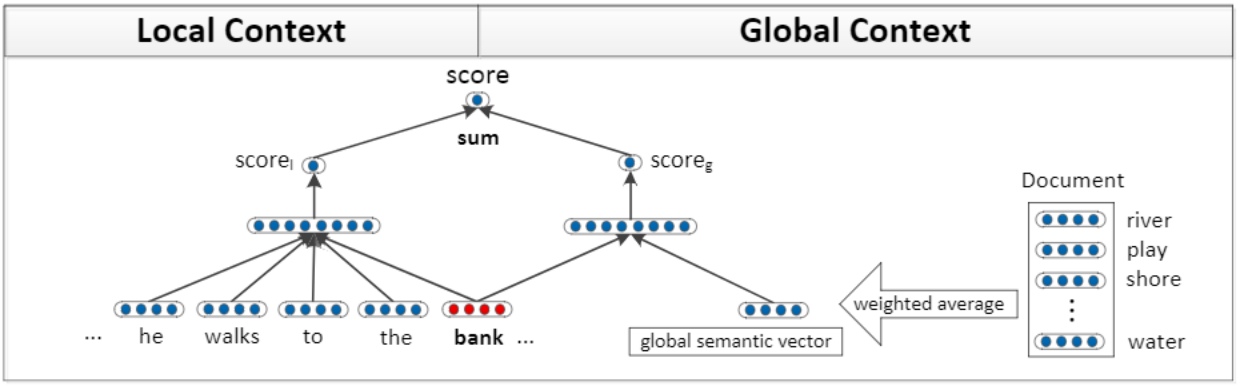

Figure 1: An overview of our neural language model. The model makes use of both local and global context to compute a score that should be large for the actual next word (*bank* in the example), compared to the score for other words. When word meaning is still ambiguous given local context, information in global context can help disambiguation.

of words presented in *isolation*, ignoring meaning variations in context. Since word interpretation in context is important especially for homonymous and polysemous words, we introduce a new dataset with human judgments on similarity between pairs of words in sentential context. To capture interesting word pairs, we sample different senses of words using WordNet (Miller, 1995). The dataset includes verbs and adjectives, in addition to nouns. We show that our multi-prototype model improves upon the single-prototype version and outperforms other neural language models and baselines on this dataset.

2 Global Context-Aware Neural Language Model

In this section, we describe the training objective of our model, followed by a description of the neural network architecture, ending with a brief description of our model's training method.

2.1 Training Objective

Our model jointly learns word representations while learning to discriminate the next word given a short word sequence (local context) and the document (global context) in which the word sequence occurs. Because our goal is to learn useful word representations and not the *probability* of the next word given previous words (which prohibits looking ahead), our model can utilize the entire document to provide global context.

Given a word sequence s and document d in which the sequence occurs, our goal is to discriminate the correct last word in s from other random words. We compute scores $g(s, d)$ and $g(s^w, d)$ where s^w is s with the last word replaced by word w, and $g(\cdot, \cdot)$ is the scoring function that represents the neural networks used. We want $g(s, d)$ to be larger than $g(s^w, d)$ by a margin of 1, for any other word w in the vocabulary, which corresponds to the training objective of minimizing the ranking loss for each (s, d) found in the corpus:

$$C_{s,d} = \sum_{w \in V} \max(0, 1 - g(s, d) + g(s^w, d)) \quad (1)$$

Collobert and Weston (2008) showed that this ranking approach can produce good word embeddings that are useful in several NLP tasks, and allows much faster training of the model compared to optimizing log-likelihood of the next word.

2.2 Neural Network Architecture

We define two scoring components that contribute to the final score of a (word sequence, document) pair. The scoring components are computed by two neural networks, one capturing local context and the other global context, as shown in Figure 1. We now describe how each scoring component is computed.

The score of local context uses the local word sequence s. We first represent the word sequence s as

an ordered list of vectors $x = (x_1, x_2, ..., x_m)$ where x_i is the embedding of word i in the sequence, which is a column in the embedding matrix $L \in \mathbb{R}^{n \times |V|}$ where $|V|$ denotes the size of the vocabulary. The columns of this embedding matrix L are the word vectors and will be learned and updated during training. To compute the score of local context, score_l, we use a neural network with one hidden layer:

$$a_1 = f(W_1[x_1; x_2; ...; x_m] + b_1) \quad (2)$$
$$\text{score}_l = W_2 a_1 + b_2 \quad (3)$$

where $[x_1; x_2; ...; x_m]$ is the concatenation of the m word embeddings representing sequence s, f is an element-wise activation function such as $\tanh$, $a_1 \in \mathbb{R}^{h \times 1}$ is the activation of the hidden layer with h hidden nodes, $W_1 \in \mathbb{R}^{h \times (mn)}$ and $W_2 \in \mathbb{R}^{1 \times h}$ are respectively the first and second layer weights of the neural network, and b_1, b_2 are the biases of each layer.

For the score of the global context, we represent the document also as an ordered list of word embeddings, $d = (d_1, d_2, ..., d_k)$. We first compute the weighted average of all word vectors in the document:

$$c = \frac{\sum_{i=1}^{k} w(t_i) d_i}{\sum_{i=1}^{k} w(t_i)} \quad (4)$$

where $w(\cdot)$ can be any weighting function that captures the importance of word t_i in the document. We use idf-weighting as the weighting function.

We use a two-layer neural network to compute the global context score, score_g, similar to the above:

$${a_1}^{(g)} = f(W_1^{(g)}[c; x_m] + b_1^{(g)}) \quad (5)$$
$$\text{score}_g = W_2^{(g)} a_1^{(g)} + b_2^{(g)} \quad (6)$$

where $[c; x_m]$ is the concatenation of the weighted average document vector and the vector of the last word in s, ${a_1}^{(g)} \in \mathbb{R}^{h^{(g)} \times 1}$ is the activation of the hidden layer with $h^{(g)}$ hidden nodes, $W_1^{(g)} \in \mathbb{R}^{h^{(g)} \times (2n)}$ and $W_2^{(g)} \in \mathbb{R}^{1 \times h^{(g)}}$ are respectively the first and second layer weights of the neural network, and $b_1^{(g)}$, $b_2^{(g)}$ are the biases of each layer. Note that instead of using the document where the sequence occurs, we can also specify a fixed $k > m$ that captures larger context.

The final score is the sum of the two scores:

$$\text{score} = \text{score}_l + \text{score}_g \quad (7)$$

The local score preserves word order and syntactic information, while the global score uses a weighted average which is similar to bag-of-words features, capturing more of the semantics and topics of the document. Note that Collobert and Weston (2008)'s language model corresponds to the network using only local context.

2.3 Learning

Following Collobert and Weston (2008), we sample the gradient of the objective by randomly choosing a word from the dictionary as a *corrupt* example for each sequence-document pair, (s, d), and take the derivative of the ranking loss with respect to the parameters: weights of the neural network and the embedding matrix L. These weights are updated via backpropagation. The embedding matrix L is the word representations. We found that word embeddings move to good positions in the vector space faster when using mini-batch L-BFGS (Liu and Nocedal, 1989) with 1000 pairs of good and corrupt examples per batch for training, compared to stochastic gradient descent.

3 Multi-Prototype Neural Language Model

Despite distributional similarity models' successful applications in various NLP tasks, one major limitation common to most of these models is that they assume only one representation for each word. This single-prototype representation is problematic because many words have multiple meanings, which can be wildly different. Using one representation simply cannot capture the different meanings. Moreover, using all contexts of a homonymous or polysemous word to build a single prototype could hurt the representation, which cannot represent any one of the meanings well as it is influenced by all meanings of the word.

Instead of using only one representation per word, Reisinger and Mooney (2010b) proposed the multi-prototype approach for vector-space models, which uses multiple representations to capture different senses and usages of a word. We show how our

model can readily adopt the multi-prototype approach. We present a way to use our learned single-prototype embeddings to represent each context window, which can then be used by clustering to perform word sense discrimination (Schütze, 1998).

In order to learn multiple prototypes, we first gather the fixed-sized context windows of all occurrences of a word (we use 5 words before and after the word occurrence). Each context is represented by a weighted average of the context words' vectors, where again, we use idf-weighting as the weighting function, similar to the document context representation described in Section 2.2. We then use spherical k-means to cluster these context representations, which has been shown to model semantic relations well (Dhillon and Modha, 2001). Finally, each word occurrence in the corpus is re-labeled to its associated cluster and is used to train the word representation for that cluster.

Similarity between a pair of words (w, w') using the multi-prototype approach can be computed with or without context, as defined by Reisinger and Mooney (2010b):

$$\text{AvgSimC}(w, w') = \frac{1}{K^2}\sum_{i=1}^{k}\sum_{j=1}^{k} p(c, w, i)p(c', w', j)d(\mu_i(w), \mu_j(w')) \quad (8)$$

where $p(c, w, i)$ is the likelihood that word w is in its cluster i given context c, $\mu_i(w)$ is the vector representing the i-th cluster centroid of w, and $d(v, v')$ is a function computing similarity between two vectors, which can be any of the distance functions presented by Curran (2004). The similarity measure can be computed in absence of context by assuming uniform $p(c, w, i)$ over i.

4 Experiments

In this section, we first present a qualitative analysis comparing the nearest neighbors of our model's embeddings with those of others, showing our embeddings better capture the semantics of words, with the use of global context. Our model also improves the correlation with human judgments on a word similarity task. Because word interpretation in context is important, we introduce a new dataset with human judgments on similarity of pairs of words in sentential context. Finally, we show that our model outperforms other methods on this dataset and also that the multi-prototype approach improves over the single-prototype approach.

We chose Wikipedia as the corpus to train all models because of its wide range of topics and word usages, and its clean organization of document by topic. We used the April 2010 snapshot of the Wikipedia corpus (Shaoul and Westbury, 2010), with a total of about 2 million articles and 990 million tokens. We use a dictionary of the 30,000 most frequent words in Wikipedia, converted to lower case. In preprocessing, we keep the frequent numbers intact and replace each digit of the uncommon numbers to "DG" so as to preserve information such as it being a year (e.g. "DGDGDGDG"). The converted numbers that are rare are mapped to a NUMBER token. Other rare words not in the dictionary are mapped to an UNKNOWN token.

For all experiments, our models use 50-dimensional embeddings. We use 10-word windows of text as the local context, 100 hidden units, and no weight regularization for both neural networks. For multi-prototype variants, we fix the number of prototypes to be 10.

4.1 Qualitative Evaluations

In order to show that our model learns more semantic word representations with global context, we give the nearest neighbors of our single-prototype model versus C&W's, which only uses local context. The nearest neighbors of a word are computed by comparing the cosine similarity between the center word and all other words in the dictionary. Table 1 shows the nearest neighbors of some words. The nearest neighbors of "market" that C&W's embeddings give are more constrained by the syntactic constraint that words in plural form are only close to other words in plural form, whereas our model captures that the singular and plural forms of a word are similar in meaning. Other examples show that our model induces nearest neighbors that better capture semantics.

Table 2 shows the nearest neighbors of our model using the multi-prototype approach. We see that the clustering is able to group contexts of different

Center Word	C&W	Our Model
markets	firms, industries, stores	market, firms, businesses
American	Australian, Indian, Italian	U.S., Canadian, African
illegal	alleged, overseas, banned	harmful, prohibited, convicted

Table 1: Nearest neighbors of words based on cosine similarity. Our model is less constrained by syntax and is more semantic.

Center Word	Nearest Neighbors
bank_1	corporation, insurance, company
bank_2	shore, coast, direction
star_1	movie, film, radio
star_2	galaxy, planet, moon
cell_1	telephone, smart, phone
cell_2	pathology, molecular, physiology
left_1	close, leave, live
left_2	top, round, right

Table 2: Nearest neighbors of word embeddings learned by our model using the multi-prototype approach based on cosine similarity. The clustering is able to find the different meanings, usages, and parts of speech of the words.

meanings of a word into separate groups, allowing our model to learn multiple meaningful representations of a word.

4.2 WordSim-353

A standard dataset for evaluating vector-space models is the WordSim-353 dataset (Finkelstein et al., 2001), which consists of 353 pairs of nouns. Each pair is presented without context and associated with 13 to 16 human judgments on similarity and relatedness on a scale from 0 to 10. For example, (cup,drink) received an average score of 7.25, while (cup,substance) received an average score of 1.92.

Table 3 shows our results compared to previous methods, including C&W's language model and the hierarchical log-bilinear (HLBL) model (Mnih and Hinton, 2008), which is a probabilistic, linear neural model. We downloaded these embeddings from Turian et al. (2010). These embeddings were trained on the smaller corpus RCV1 that contains one year of Reuters English newswire, and show similar correlations on the dataset. We report the result of

Model	Corpus	$\rho \times 100$
Our Model-g	Wiki.	22.8
C&W	RCV1	29.5
HLBL	RCV1	33.2
C&W*	Wiki.	49.8
C&W	Wiki.	55.3
Our Model	Wiki.	64.2
Our Model*	Wiki.	71.3
Pruned *tf-idf*	Wiki.	73.4
ESA	Wiki.	75
Tiered Pruned *tf-idf*	Wiki.	76.9

Table 3: Spearman's ρ correlation on WordSim-353, showing our model's improvement over previous neural models for learning word embeddings. C&W* is the word embeddings trained and provided by C&W. Our Model* is trained without stop words, while Our Model-g uses only global context. Pruned *tf-idf* (Reisinger and Mooney, 2010b) and ESA (Gabrilovich and Markovitch, 2007) are also included.

our re-implementation of C&W's model trained on Wikipedia, showing the large effect of using a different corpus.

Our model is able to learn more semantic word embeddings and noticeably improves upon C&W's model. Note that our model achieves higher correlation (64.2) than either using local context alone (C&W: 55.3) or using global context alone (Our Model-g: 22.8). We also found that correlation can be further improved by removing stop words (71.3). Thus, each window of text (training example) contains more information but still preserves some syntactic information as the words are still ordered in the local context.

4.3 New Dataset: Word Similarity in Context

The many previous datasets that associate human judgments on similarity between pairs of words, such as WordSim-353, MC (Miller and Charles, 1991) and RG (Rubenstein and Goodenough, 1965), have helped to advance the development of vector-space models. However, common to all datasets is that similarity scores are given to pairs of words in *isolation*. This is problematic because the meanings of homonymous and polysemous words depend highly on the words' contexts. For example, in the two phrases, "he swings the baseball *bat*" and "the

Word 1	Word 2
Located downtown along the east **bank** of the Des Moines River ...	This is the basis of all **money** laundering , a track record of depositing clean money before slipping through dirty money ...
Inside the ruins , there are **bats** and a bowl with Pokeys that fills with sand over the course of the race , and the music changes somewhat while inside ...	An aggressive lower order batsman who usually **bats** at No. 11 , Muralitharan is known for his tendency to back away to leg and slog ...
An example of legacy **left** in the Mideast from these nobles is the Krak des Chevaliers ' enlargement by the Counts of Tripoli and Toulouse ...	... one should not adhere to a particular explanation , only in such measure as to be ready to **abandon** it if it be proved with certainty to be false ...
... and Andy 's getting ready to **pack** his bags and head up to Los Angeles tomorrow to get ready to fly back home on Thursday	... she encounters Ben (Duane Jones) , who arrives in a pickup truck and defends the house against another **pack** of zombies ...
In **practice** , there is an unknown phase delay between the transmitter and receiver that must be compensated by " synchronization " of the receivers local oscillator	... but Gilbert did not believe that she was dedicated enough , and when she missed a **rehearsal** , she was dismissed ...

Table 4: Example pairs from our new dataset. Note that words in a pair can be the same word and have different parts of speech.

bat flies", *bat* has completely different meanings. It is unclear how this variation in meaning is accounted for in human judgments of words presented without context.

One of the main contributions of this paper is the creation of a new dataset that addresses this issue. The dataset has three interesting characteristics: 1) human judgments are on pairs of words presented in sentential context, 2) word pairs and their contexts are chosen to reflect interesting variations in meanings of homonymous and polysemous words, and 3) verbs and adjectives are present in addition to nouns. We now describe our methodology in constructing the dataset.

4.3.1 Dataset Construction

Our procedure of constructing the dataset consists of three steps: 1) select a list a words, 2) for each word, select another word to form a pair, 3) for each word in a pair, find a sentential context. We now describe each step in detail.

In step 1, in order to make sure we select a diverse list of words, we consider three attributes of a word: frequency in a corpus, number of parts of speech, and number of synsets according to WordNet. For frequency, we divide words into three groups, top 2,000 most frequent, between 2,000 and 5,000, and between 5,000 to 10,000 based on occurrences in Wikipedia. For number of parts of speech, we group words based on their number of possible parts of speech (noun, verb or adjective), from 1 to 3. We also group words by their number of synsets: [0,5], [6,10], [11, 20], and [20, $\max$]. Finally, we sample at most 15 words from each combination in the Cartesian product of the above groupings.

In step 2, for each of the words selected in step 1, we want to choose the other word so that the pair captures an interesting relationship. Similar to Manandhar et al. (2010), we use WordNet to first randomly select one synset of the first word, we then construct a set of words in various relations to the first word's chosen synset, including hypernyms, hyponyms, holonyms, meronyms and attributes. We randomly select a word from this set of words as the second word in the pair. We try to repeat the above twice to generate two pairs for each word. In addition, for words with more than five synsets, we allow the second word to be the same as the first, but with different synsets. We end up with pairs of words as well as the one chosen synset for each word in the pairs.

In step 3, we aim to extract a sentence from Wikipedia for each word, which contains the word and corresponds to a usage of the chosen synset. We first find all sentences in which the word occurs. We then POS tag[2] these sentences and filter out those that do not match the chosen POS. To find the

[2]We used the MaxEnt Treebank POS tagger in the python nltk library.

Model	$\rho \times 100$
C&W-S	57.0
Our Model-S	58.6
Our Model-M AvgSim	62.8
Our Model-M AvgSimC	**65.7**
tf-idf-S	26.3
Pruned *tf-idf*-S	62.5
Pruned *tf-idf*-M AvgSim	60.4
Pruned *tf-idf*-M AvgSimC	60.5

Table 5: Spearman's ρ correlation on our new dataset. Our Model-S uses the single-prototype approach, while Our Model-M uses the multi-prototype approach. AvgSim calculates similarity with each prototype contributing equally, while AvgSimC weighs the prototypes according to probability of the word belonging to that prototype's cluster.

word usages that correspond to the chosen synset, we first construct a set of related words of the chosen synset, including hypernyms, hyponyms, holonyms, meronyms and attributes. Using this set of related words, we filter out a sentence if the document in which the sentence appears does not include one of the related words. Finally, we randomly select one sentence from those that are left.

Table 4 shows some examples from the dataset. Note that the dataset also includes pairs of the same word. Single-prototype models would give the max similarity score for those pairs, which can be problematic depending on the words' contexts. This dataset requires models to examine context when determining word meaning.

Using Amazon Mechanical Turk, we collected 10 human similarity ratings for each pair, as Snow et al. (2008) found that 10 non-expert annotators can achieve very close inter-annotator agreement with expert raters. To ensure worker quality, we only allowed workers with over 95% approval rate to work on our task. Furthermore, we discarded all ratings by a worker if he/she entered scores out of the accepted range or missed a rating, signaling low-quality work.

We obtained a total of 2,003 word pairs and their sentential contexts. The word pairs consist of 1,712 unique words. Of the 2,003 word pairs, 1328 are noun-noun pairs, 399 verb-verb, 140 verb-noun, 97 adjective-adjective, 30 noun-adjective, and 9 verb-adjective. 241 pairs are same-word pairs.

4.3.2 Evaluations on Word Similarity in Context

For evaluation, we also compute Spearman correlation between a model's computed similarity scores and human judgments. Table 5 compares different models' results on this dataset. We compare against the following baselines: *tf-idf* represents words in a word-word matrix capturing co-occurrence counts in all 10-word context windows. Reisinger and Mooney (2010b) found pruning the low-value *tf-idf* features helps performance. We report the result of this pruning technique after tuning the threshold value on this dataset, removing all but the top 200 features in each word vector. We tried the same multi-prototype approach and used spherical k-means[3] to cluster the contexts using *tf-idf* representations, but obtained lower numbers than single-prototype (55.4 with AvgSimC). We then tried using pruned *tf-idf* representations on contexts with our clustering assignments (included in Table 5), but still got results worse than the single-prototype version of the pruned *tf-idf* model (60.5 with AvgSimC). This suggests that the pruned *tf-idf* representations might be more susceptible to noise or mistakes in context clustering.

By utilizing global context, our model outperforms C&W's vectors and the above baselines on this dataset. With multiple representations per word, we show that the multi-prototype approach can improve over the single-prototype version without using context (62.8 vs. 58.6). Moreover, using AvgSimC[4] which takes contexts into account, the multi-prototype model obtains the best performance (65.7).

5 Related Work

Neural language models (Bengio et al., 2003; Mnih and Hinton, 2007; Collobert and Weston, 2008; Schwenk and Gauvain, 2002; Emami et al., 2003) have been shown to be very powerful at language modeling, a task where models are asked to accurately predict the next word given previously seen words. By using distributed representations of

[3]We first tried movMF as in Reisinger and Mooney (2010b), but were unable to get decent results (only 31.5).

[4]probability of being in a cluster is calculated as the inverse of the distance to the cluster centroid.

words which model words' similarity, this type of models addresses the data sparseness problem that n-gram models encounter when large contexts are used. Most of these models used relative local contexts of between 2 to 10 words. Schwenk and Gauvain (2002) tried to incorporate larger context by combining partial parses of past word sequences and a neural language model. They used up to 3 previous head words and showed increased performance on language modeling. Our model uses a similar neural network architecture as these models and uses the ranking-loss training objective proposed by Collobert and Weston (2008), but introduces a new way to combine local and global context to train word embeddings.

Besides language modeling, word embeddings induced by neural language models have been useful in chunking, NER (Turian et al., 2010), parsing (Socher et al., 2011b), sentiment analysis (Socher et al., 2011c) and paraphrase detection (Socher et al., 2011a). However, they have not been directly evaluated on word similarity tasks, which are important for tasks such as information retrieval and summarization. Our experiments show that our word embeddings are competitive in word similarity tasks.

Most of the previous vector-space models use a single vector to represent a word even though many words have multiple meanings. The multi-prototype approach has been widely studied in models of categorization in psychology (Rosseel, 2002; Griffiths et al., 2009), while Schütze (1998) used clustering of contexts to perform word sense discrimination. Reisinger and Mooney (2010b) combined the two approaches and applied them to vector-space models, which was further improved in Reisinger and Mooney (2010a). Two other recent papers (Dhillon et al., 2011; Reddy et al., 2011) present models for constructing word representations that deal with context. It would be interesting to evaluate those models on our new dataset.

Many datasets with human similarity ratings on pairs of words, such as WordSim-353 (Finkelstein et al., 2001), MC (Miller and Charles, 1991) and RG (Rubenstein and Goodenough, 1965), have been widely used to evaluate vector-space models. Motivated to evaluate composition models, Mitchell and Lapata (2008) introduced a dataset where an intransitive verb, presented with a subject noun, is compared to another verb chosen to be either similar or dissimilar to the intransitive verb in context. The context is short, with only one word, and only verbs are compared. Erk and Padó (2008), Thater et al. (2011) and Dinu and Lapata (2010) evaluated word similarity in context with a modified task where systems are to rerank gold-standard paraphrase candidates given the SemEval 2007 Lexical Substitution Task dataset. This task only indirectly evaluates similarity as only reranking of already similar words are evaluated.

6 Conclusion

We presented a new neural network architecture that learns more semantic word representations by using both local and global context in learning. These learned word embeddings can be used to represent word contexts as low-dimensional weighted average vectors, which are then clustered to form different meaning groups and used to learn multi-prototype vectors. We introduced a new dataset with human judgments on similarity between pairs of words in context, so as to evaluate model's abilities to capture homonymy and polysemy of words in context. Our new multi-prototype neural language model outperforms previous neural models and competitive baselines on this new dataset.

Acknowledgments

The authors gratefully acknowledges the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181, and the DARPA Deep Learning program under contract number FA8650-10-C-7020. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, Jaz K, Thomas Hofmann, Tomaso Poggio, and John Shawe-taylor. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

James Richard Curran. 2004. From distributional to semantic similarity. Technical report.

Inderjit S. Dhillon and Dharmendra S. Modha. 2001. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42:143–175, January.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via cca. In *Advances in Neural Information Processing Systems (NIPS)*, volume 24.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1162–1172, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ahmad Emami, Peng Xu, and Frederick Jelinek. 2003. Using a connectionist model in a syntactical based language model. In *Acoustics, Speech, and Signal Processing*, pages 372–375.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: the concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 406–414, New York, NY, USA. ACM.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Thomas L Griffiths, Kevin R Canini, Adam N Sanborn, and Daniel J Navarro. 2009. Unifying rational models of categorization via the hierarchical dirichlet process. *Brain*, page 323328.

David J Hess, Donald J Foss, and Patrick Carroll. 1995. Effects of global and local context on lexical processing during language comprehension. *Journal of Experimental Psychology: General*, 124(1):62–82.

Ping Li, Curt Burgess, and Kevin Lund. 2000. The acquisition of word meaning through global lexical cooccurrences.

D. C. Liu and J. Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528, December.

Suresh Manandhar, Ioannis P Klapaftis, Dmitriy Dligach, and Sameer S Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. *Word Journal Of The International Linguistic Association*, (July):63–68.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language & Cognitive Processes*, 6(1):1–28.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

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

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 641–648, New York, NY, USA. ACM.

Andriy Mnih and Geoffrey Hinton. 2008. A scalable hierarchical distributed language model. In *In NIPS*.

Ht Ng and J Zelle. 1997. Corpus-based approaches to semantic interpretation in natural language processing. *AI Magazine*, 18(4):45–64.

Siva Reddy, Ioannis Klapaftis, Diana McCarthy, and Suresh Manandhar. 2011. Dynamic and static prototype vectors for semantic composition. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 705–713, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

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*, EMNLP '10, pages 1173–1182, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, HLT '10, pages 109–117, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yves Rosseel. 2002. Mixture models of categorization. *Journal of Mathematical Psychology*, 46:178–210.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM*, 8:627–633, October.

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

Holger Schwenk and Jean-luc Gauvain. 2002. Connectionist language modeling for large vocabulary continuous speech recognition. In *In International Conference on Acoustics, Speech and Signal Processing*, pages 765–768.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, March.

Cyrus Shaoul and Chris Westbury. 2010. The westbury lab wikipedia corpus.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*.

Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. 2011b. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011c. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 Search and Development in Information Retrieval*, pages 41–47. ACM Press.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: a simple and effective vector model. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, IJCNLP '11.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.