Investigating Different Syntactic Context Types and Context Representations for Learning Word Embeddings

Bofang Li[1,2] libofang@ruc.edu.cn
Tao Liu[1,2] tliu@ruc.edu.cn
Zhe Zhao[1,2] helloworld@ruc.edu.cn
Buzhou Tang[3] tangbuzhou@gmail.com
Aleksandr Drozd[4] alex@smg.is.titech.ac.jp
Anna Rogers[5] arogers@cs.uml.edu
Xiaoyong Du[1,2] duyong@ruc.edu.cn

[1] School of Information, Renmin University of China
[2] Key Laboratory of Data Engineering and Knowledge Engineering, MOE
[3] Shenzhen Graduate School, Harbin Institute of Technology
[4] Global Scientific Information and Computing Center, Tokyo Institute of Technology
[5] Department of Computer Science, University of Massachusetts Lowell

Abstract

The number of word embedding models is growing every year. Most of them are based on the co-occurrence information of words and their contexts. However, it is still an open question what is the best definition of context. We provide a systematical investigation of 4 different syntactic context types and context representations for learning word embeddings. Comprehensive experiments are conducted to evaluate their effectiveness on 6 extrinsic and intrinsic tasks. We hope that this paper, along with the published code, would be helpful for choosing the best context type and representation for a given task.

1 Introduction

Recently, there is a growing interest in word embedding models, where words are embedded into low-dimensional (dense) real-valued vectors. The trained word embeddings can be directly used for solving intrinsic tasks like word similarity and word analogy. They are also helpful for solving extrinsic tasks, such as part-of-speech tagging, chunking, named entity recognition (Collobert and Weston, 2008; Collobert et al., 2011) and text classification (Socher et al., 2013; Kim, 2014).

The training objectives of word embedding models are based on the Distributional Hypothesis (Harris, 1954) that can be stated as follows: "words that occur in similar contexts tend to have similar meanings". In most word embedding models, the "context" is defined as the words which precede and follow the target word within some fixed distance (Bengio et al., 2003; Mnih and Hinton, 2007; Mikolov et al., 2013a; Pennington et al., 2014). Among them, Global Vectors (GloVe) proposed by Pennington et al. (2014), Continuous Skip-Gram (CSG) [1] and Continuous Bag-Of-Words (CBOW) proposed by Mikolov et al. (2013b) achieve state-of-the-art results on a range of linguistic tasks, and scale to corpora with billions of words.

The traditional sparse vector-space models have explored many different types of context. Curran (2004); Padó and Lapata (2007); Clark (2012) have discussed a set of context definitions beyond simple linear context. For example, a sentence or document could be used as the boundary instead of window size. Contextual words could be associated with their relative sides (left/right) or positions (+1/-2) to the target word. They could also be associated with part-of-speech or grammatical relation labels. The weight of each contextual word can be explicitly defined. Moreover, words that are connected to target word in dependency parse

[1] Many researches refer to Continuous Skip-Gram as SG. However, in order to distinguish linear (continuous) context and DEPS (dependency-based) context, we refer it as CSG.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2404–2414
Copenhagen, Denmark, September 7–11, 2017.

Basic Model	Context Representation \ Context Type	Linear	DEPS
generalized Skip-Gram	unbound	CSG (Mikolov et al., 2013a)	this work
	bound	Structured SG (Ling et al., 2015) POSIT (Levy and Goldberg, 2014b)	DEPS (Levy and Goldberg, 2014a)
generalized Bag-Of-Words	unbound	CBOW (Mikolov et al., 2013a)	this work
	bound	CWINDOW (Ling et al., 2015)	this work
original GloVe	unbound	GloVe (Pennington et al., 2014)	this work
	bound	this work	this work

Table 1: Summary of prior research on word embedding models with different syntactic context types and context representations. For linear context, *bound* indicates words associated with positional information. For DEPS context, *bound* indicates words associated with dependency relation.

tree can be considered as context.

Recent word embedding models have also explored some of the above context types. Levy and Goldberg (2014b); Ling et al. (2015)[2] improve CSG and CBOW by introducing position-aware context representation. Levy and Goldberg (2014a) propose dependency-based context (DEPS) for CSG.

However, different types of syntactic context have not been systematically compared for different word embeddings. This paper explores two context types (linear or DEPS) and two context representations (bound or unbound), as shown in Table 1. Three popular word embedding models (CBOW, GloVe, and CSG) are compared on word similarity, word analogy, part-of-speech tagging, chunking, named entity recognition, and text classification tasks.

2 Related Work

Several studies directly compare different word embedding models. Lai et al. (2016) compare 6 word embedding models using different corpora and hyper-parameters. Nayak and Manning (2016) provide a set of evaluations, along with an online tool, for word embedding models. Levy and Goldberg (2014c) show the theoretical equivalence of CSG and PPMI matrix factorization. Levy et al. (2015) further discuss the connections between 4 word embedding models (PPMI, SVD, CSG, GloVe) and re-evaluate them with the same hyper-parameters. Suzuki and Nagata (2015) investigate different configurations of CSG and GloVe and merge them together. Yin and Schutze (2016) propose 4 ensemble methods and show their effectiveness over individual ones.

There is also research evaluating different context types in learning word embeddings. Heylen et al. (2008) compares dependency-based and linear vector space model for finding semantically related nouns in Dutch. Vulic and Korhonen (2016) compare CSG and dependency-based models on various languages. Their results suggest that dependency-based models are better at detecting functional similarity in English, although that does not necessarily hold for other languages. Bansal et al. (2014) show that DEPS context is preferable to linear context on parsing task. Melamud et al. (2016) investigate the performance of CSG, DEPS and a substitute-based word embedding model (Yatbaz et al., 2012)[3], which shows that different types of intrinsic tasks have clear preference for particular types of contexts. On the other hand, for extrinsic tasks, the optimal context types need to be carefully tuned on specific dataset.

The contribution of this study is that in addition to linear and dependency-based context we also consider bound and unbound context representations, as will be described below. Furthermore, we systematically evaluate three word embedding models: CSG, CBOW and GLoVe.

[2]In these two papers, the description of position-aware (bound) context are quite different. However, their ideas are actually identical.

[3]We do not consider this type of context, since in our pilot studies it performed consistently worse than the other two context types. The same observation is also made by Melamud et al. (2016); Vulic and Korhonen (2016).

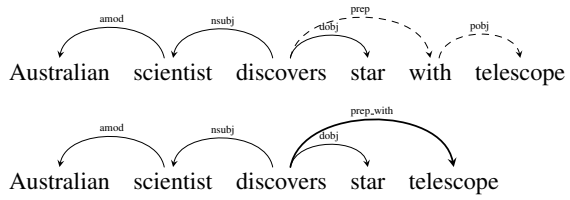

Figure 1: Illustration of dependency parse tree.

3 Word Embeddings Models

In this section, we first introduce different contexts in detail, and discuss their strengths and weaknesses. We then show how CSG, CBOW and GloVe can be generalized to use these contexts.

3.1 Context Types

There are many different types of context, both on document and sentence level. For syntactic contexts, the current literature discusses mainly the linear (used in most word embedding models) and dependency-based contexts (DEPS (Levy and Goldberg, 2014a)). Linear context is defined as the positional neighbors of the target word in texts. DEPS context is defined as the syntactic neighbors of the target word based on dependency parse tree, as shown in Figure 1[4].

Compared to the linear context, DEPS context can capture more relevant words that are further away from the target word in the text. For example in Figure 1, linear context does not include the word-context pair "discovers telescope", while DEPS context contains this information. DEPS context can also exclude some uninformative word-context pairs like "with star" and "telescope with".

Note that dependency parsing is time-consuming. Despite its parallelizability, our implementation still takes nearly a month to finish dependency parsing for the Wikipedia corpus on a 32-core machine. it is only fair to compare linear and DEPS context if we ignore the time complexity. it is also worth noting that part-of-speech labels are required when performing dependency parsing.

3.2 Context Representations

In the original CSG, CBOW and GloVe models, contexts are represented by words without any additional information. Ling et al. (2015) modify CSG and CBOW by introducing position-bound words, where each contextual word is associated with their relative position to the target word. This allows CSG and CBOW to distinguish different sequential positions and capture the structural information from the context. We refer to methods that bind positional information with the contextual word as bound (context) representation, as opposed to unbound (context) representation where contextual words are treated the same irrespective of their positions with regards to the target word.

The original DEPS uses "bound" representation by default: each word is associated with its dependency relation to the target word. In this paper, we also investigate the simpler context representation where no dependency relation is associated with a word. This enables a fair comparison with conventional models like CSG, CBOW and GloVe, since they do not use bound representation either. An example of different syntactic context types and context representations is shown in Table 2.

Context Representation \ Context Type	Linear	DEPS
unbound	australian, scientist, star, with	scientist, star, telescope
bound	australian/-2, scientist/-1, star/+1, with/+2	scientist/nsubj, star/dobj, telescope/prep_with

Table 2: Illustration of bound and unbound representations under linear and DEPS context types. This example is based on Figure 1, and the target word is "discovers".

Intuitively, bound representation should work better than unbound representation, since it uses information about relative word positions. However, this is not always the case in practice. An obvious drawback is that bound representation is more sparse than unbound representation, especially for DEPS context type. In our data, there were 47 dependency relations in dependency parse tree. Although not every combination of dependency relations and words appear in the word-context pair collection, in practice it still enlarges the contextual words' vocabulary about 5 times.

Both syntactic context types (linear and DEPS) and the choice of context representations (bound and unbound) have a dramatic effect on the word embeddings. Bound linear representation transfers each contextual word into a new one, and the

[4]This example is from Levy and Goldberg (2014a)

	Linear (window size 1)	DEPS
P	(**australian**, scientist) (**scientist**, australian) (**scientist**, discovers) (**discovers**, scientist) (**discovers**, star) ...	(**australian**, scientist) (**scientist**, australian) (**scientist**, discovers) (**discovers**, scientist) (**discovers**, star) (**discovers**, telescope) ...
M	(**australian**, scientist) (**scientist**, australian, discovers) (**discovers**, scientist, star) ...	(**australian**, scientist) (**scientist**, australian, discovers) (**discovers**, scientist, star, telescope) ...
$\overline{M}$	(**australian**, scientist, 1) (**scientist**, australian, 1) (**scientist**, discovers, 1) (**discovers**, scientist, 1) (**discovers**, star, 1) ...	(**australian**, scientist, 1) (**scientist**, australian, 1) (**scientist**, discovers, 1) (**discovers**, scientist, 1) (**discovers**, star, 1) (**discovers**, telescope, 1) ...

Table 3: Illustration of collection P, M and $\overline{M}$ for sentence "australian scientist discovers star with telescope". Unbound representation is used in this example. Words in the collections are **Bold**. .

word-context pairs are changed completely. DEPS, as compared to the linear contexts, increases the likelihood that the contextual words are in a meaningful relation with the target word, although some words captured by DEPS would also be found in the linear contexts if the window is wide enough. For example, in Table 2, "scientist" and "star" are considered as the contextual words of "discovers" in both linear and DEPS context types.

3.3 Generalization

Let P be a collection of word-context pairs. P can be merged based on the words to form a collection M with size of $|V|$, where V is the vocabulary. Each element $(w, c_1, c_2, .., c_{n_w}) \in M$ is word w and its contexts, where n_w is the number of word w's contexts. P can also be merged based on both words and contexts to form a collection $\overline{M}$. Each element $(w, c, \#(w, c)) \in \overline{M}$ is the word w, context c, and the times they appear in collection P. An example of these collections is shown in Table 3.

3.3.1 Generalized Bag-Of-Words

The objective function of Generalized Bag-Of-Words (GBOW) is defined as:

$$\sum_{(w,c_1,..,c_{n_w})\in M} \log p\left(w \middle| \sum_{i=1}^{n_w} \vec{c_i}\right) \quad (1)$$

With negative sampling technique, the log probability is calculated by:

$$\log \sigma\left(\vec{w} \cdot \sum_{i=1}^{n_w} \vec{c_i}\right) - \sum_{k=1}^{K} \log \sigma\left(\vec{w_{N_k}} \cdot \sum_{1=i}^{n_w} \vec{c_i}\right) \quad (2)$$

where σ is the sigmoid function, K is the negative sampling size, $\vec{w}$ and $\vec{c}$ is the vector for word w and c respectively. The negatively sampled word w_{N_k} is randomly selected on the basis of its unigram distribution $(\frac{\#(w)}{\sum_w \#(w)})^{ds}$, where $\#(w)$ is the number of times that word w appears in the corpus, and ds is the distribution smoothing hyperparameter which is usually defined as 0.75.

Note that with negative sampling technique, both GBOW and original CBOW (Mikolov et al., 2013a) will learn two sets of embeddings (word embeddings and context embeddings). In the original CBOW, the context embeddings can also be considered as word embeddings, since the vocabulary set of words and contexts are the same. However, for bound context, the words (i.e. scientist) and contexts (i.e. scientist/nsubj) are quite different. It is necessary to distinguish conditioned and conditioning variables. For example, in Figure 1, the context "scientist/nsubj" can only be predicted by word "discovers". However, most of the word is connected to several contextual words. Due to this, the sum of contextual word embeddings should be used for predicting the target word.

3.3.2 Generalized Skip-Gram

For generalized Skip-Gram (GSG), the definition is more straightforward and the objective function actually needs no specification (Levy and Goldberg, 2014b). Nonetheless, in order to make it consistent with our GBOW, we also specify the conditioned and conditioning variables in the objective function:

$$\begin{aligned} &\sum_{(w,c)\in P} \log p\left(w|\vec{c}\right) \\ &= \sum_{(w,c)\in P} \left[\log \sigma\left(\vec{w} \cdot \vec{c}\right) - \sum_{k=1}^{K} \log \sigma\left(\vec{w_{N_k}} \cdot \vec{c}\right)\right] \end{aligned} \quad (3)$$

Note that this generalization does not change the nature of the models for linear context. In our pilot experiments on word analogy and word similarity, the performance of both GSG and GBOW is almost identical to their original versions.

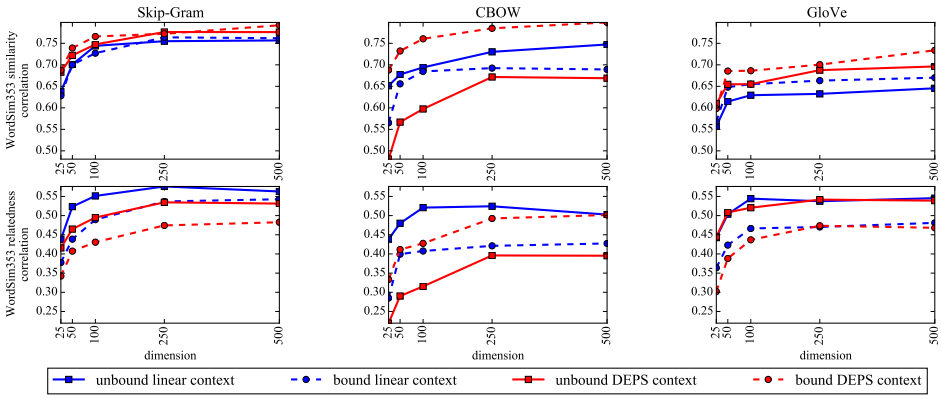

Figure 2: Correlation results for similarity and relatedness categories on WordSim353 (word similarity) dataset.

3.3.3 GloVe

Unlike GSG and GBOW, GloVe explicitly optimizes a log-bilinear regression model based on word co-occurrence matrix. Since GloVe is already a very generalized model, with the previous defined collection $\overline{M}$, the final objective function is written as:

$$\sum_{(w,c)\in\overline{M}} f(\#(w,c))(\vec{w}\cdot\vec{c}+\vec{b_w}+\vec{b_c}-\log\#(w,c)) \quad (4)$$

where $\vec{b_w}$ and $\vec{b_c}$ are biases for word and context. f is a non-decreasing weighting function and ensures that large $\#(w,c)$ is not over-weighted.

Note that the inputs of GSG, GBOW and Glove are the collections P, M and $\overline{M}$ respectively. Once the corpus and hyper-parameters are fixed, these collections (and thus the learned word embeddings) are determined only by the choice of context types and representations.

4 Experiments

We evaluate the effectiveness of different syntactic context types and context representations on word similarity, word analogy, part-of-speech tagging, chunking, named entity recognition, and text classification tasks. In this section we describe our models, and then report and discuss the experimental results on each task.

4.1 Word Embeddings

Previously, the `word2vecf` toolkit [5] (Levy et al., 2015) extended the `word2vec` toolkit [6] (Mikolov et al., 2013b) to accept the input of collection P rather than raw corpus. This makes CSG model accept arbitrary contexts (e.g. DEPS context). However, CBOW and GloVe are not considered in that toolkit. We implement `word2vecPM` toolkit, a further extension of `word2vecf`, which supports generalized CSG, CBOW and GloVe with the input of collection P, M and $\overline{M}$ respectively. For fair comparison, as suggested by Levy et al. (2015), we use the same hyper-parameters [7] for all embedding models. English Wikipedia (August 2013 dump) is used as the training corpus. The Stanford CoreNLP (Manning et al., 2014) is used for dependency parsing. After parsing, tokens are converted to lowercase. Words and contexts that appear fewer than 100 times in the collection P are ignored.

4.2 Word Similarity Task

Word similarity task aims at producing semantic similarity scores of word pairs, which are compared with the human scores using Spearman's correlation. The cosine distance is used for generating similarity scores between two word vectors. We use the WordSim353 (Finkelstein et al., 2001) dataset, divided into similarity and relatedness categories (Zesch et al., 2008; Agirre et al., 2009).

Previous research (Levy and Goldberg, 2014a; Melamud et al., 2016) concluded that compared to linear context, DEPS context can capture more functional similarity (e.g. tiger/cat) rather than topical similarity (relatedness) (e.g. tiger/jungle). However, their experiments do not distinguish the

[5] https://bitbucket.org/yoavgo/word2vecf

[6] http://code.google.com/p/word2vec/

[7] Negative sampling size is set to 5 for SG and 2 for CBOW. Distribution smoothing is set to 0.75. No dynamic context or "dirty" sub-sampling is used. The window size is fixed to 2. The number of iterations is set to 2, 5 and 30 for SG, CBOW and GloVe respectively.

Model	Context Type	Context Representation	Similarity			Relatedness	Similarity+Relatedness	
			WS353	Rare Words	SimLex-999	WS353	MEN	Mech Turk
GSG	linear	unbound	.757	.414	.417	**.563**	**.732**	.632
		bound	.762	.421	**.434**	.543	.695	.608
	dep	unbound	.776	**.422**	.418	.531	.728	**.644**
		bound	**.792**	.413	.421	.483	.674	.643
GBOW	linear	unbound	.747	**.436**	**.439**	**.503**	**.718**	**.644**
		bound	.689	.403	.428	.427	.659	.512
	dep	word	.669	.412	.386	.395	.667	.541
		bound	**.799**	.434	.403	.502	.640	.587
GloVe	linear	unbound	.645	.354	.323	**.545**	.662	.587
		bound	.670	.400	.363	.481	.563	.587
	dep	unbound	.696	.371	.342	.539	**.692**	**.603**
		bound	**.734**	**.409**	**.406**	.468	.541	.557

Table 4: Numerical results on word similarity datasets. Best results in group are marked **Bold**.

Model	Context Type	Context Representation	Google Sem	Google Syn	MSR	Inflectional morphology	Derivational morphology	Encyclopedia	Lexicography
GSG	linear	unbound	.708	.639	.642	.678	.110	.242	.083
		bound	.702	.454	**.653**	.668	.111	.208	**.099**
	dep	unbound	**.716**	**.661**	.644	**.691**	**.122**	**.253**	.095
		bound	.600	.307	.600	.668	.112	.170	**.099**
GBOW	linear	unbound	**.628**	**.566**	**.601**	**.618**	**.096**	.201	.074
		bound	.602	.376	.569	.572	.091	.157	**.081**
	dep	unbound	.573	.553	.520	.496	.094	**.216**	.076
		bound	.495	.248	.516	.563	.086	.126	.078
GloVe	linear	unbound	.471	**.719**	.454	.425	.033	.226	.054
		bound	.502	.218	**.542**	**.559**	**.044**	.129	**.095**
	dep	unbound	**.513**	.700	.525	.491	.043	**.227**	.063
		bound	.402	.121	.525	.446	.033	.093	.083

Table 5: Numerical results on word analogy datasets. Best results in group are marked **Bold**.

effect of different context representations: unbound representation is used for linear context (Mikolov et al., 2013b), while bound representation is used for dependency-based context (Levy and Goldberg, 2014a). Moreover, only CSG model is considered.

We revisit those claims with more systematical experiments. As shown in the top-left sub-figure of Figure 2, DEPS does outperform the linear contextin GSG and GloVe in the similarity section of WordSim353, confirming its ability to capture functional similarity. However, the advantage of DEPS does not fully transfer to GBOW. Although bound DEPS context for GBOW is still the best performer, unbound DEPS context performs the worst, which shows the importance of bound vs unbound representation.

Note that the results are also reversed on WordSim353 relatedness section (the right subfigure of Figure 2), which shows that linear context is more suitable for capturing topical similarity.

Overall, DEPS context type does not get all the credit for capturing functional similarity. Context representations play an important role for word similarity task. it is only safe to say that DEPS context captures functional similarity with the "help" of bound representation. In contrast, linear context type captures topical similarity with the "help" of unbound representation.

However, the above findings come with a major caveat: a lot seems to depend on the particular dataset, in addition to the model and context type. We experimented with MEN dataset (Bruni et al., 2012), Mechanical Turk dataset (Radinsky et al., 2011), Rare Words dataset (Luong et al., 2013), and SimLex-999 dataset (Hill et al., 2016) (Table 4), and we were not able to observe uniform trends even for datasets that are supposed to capture the same relation - like the similarity part of WordSim353, Rare Words and SimLex.

Still, some models do favor a certain context type for both similarity and relatedness: e.g. GBOW favors linear unbound contexts, while GLoVE in most cases prefers DEPS over the linear context. In case of GCG, however, context type needs to be optimized for the particular dataset.

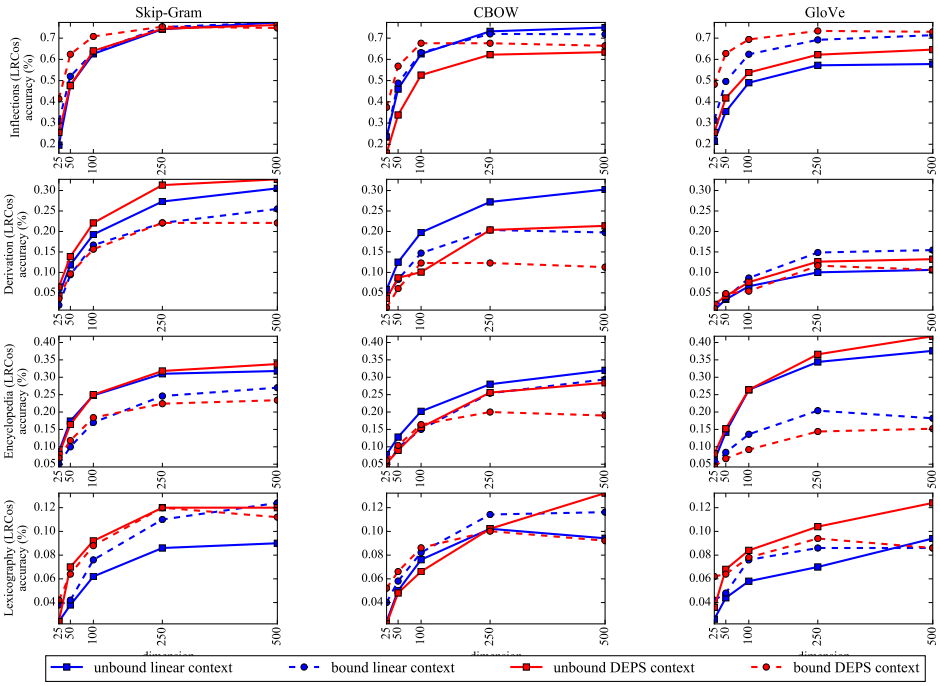

Figure 3: Averaged accuracy results for all Inflections, Derivation, Encyclopedia and Lexicography categories on BATS word analogy dataset.

4.3 Word Analogy Task

Word analogy task aims at answering the questions like "a is to a' as b is to __ ?", such as "London is to Britain as Tokyo is to Japan". We follow the evaluation protocol in Levy and Goldberg (2014b), which answers the questions using LRCos method (Drozd et al., 2016). LRCos shows significant improvement over the traditional vector offset method. We use BATS analogy dataset (Gladkova et al., 2016) in our experiments.

As shown in Figure 3, context representation plays an important role in word analogy task. The choice of context representation (bound or unbound) actually has much larger impact than the choice of context type (linear or DEPS). The results on Encyclopedia category are perhaps the most evident. The performance of unbound linear context and unbound DEPS context is similar. However, for most models and categories, bound representation seems to outperform unbound representation. When bound representation is used, the performance drops around $5-15$ percent for DEPS context in terms of accuracy. This is consistent with the findings of Levy and Goldberg (2014a), who report that DEPS context did not work well for the analogy task.

As shown in Table 5, we have also experimented on two much smaller datasets: MSR analogy dataset (Mikolov et al., 2013c), and Google analogy dataset (Mikolov et al., 2013a) (with semantic and syntactic questions). They also show that the choice of context representation has more impact than the choice of context type.

4.4 POS, Chunking and NER Tasks

Although intrinsic evaluations like word similarity and word analogy tasks could provide direct insights about different context types and representations, they have certain methodological problems (Gladkova and Drozd, 2016), and the experimental results above cannot be directly translated to the typical uses of word embeddings in downstream tasks (Schnabel et al., 2015; Linzen, 2016; Chiu et al., 2016). Thus extrinsic tasks should also be considered.

In this subsection, we evaluate the effectiveness of different word embedding models with different contexts on Part-of-Speech Tagging (POS), Chunking[8] and Named Entity Recognition (NER) tasks [9]. For these tasks, a NLP system assigns labels to elements of texts. Note that in practice, one should NOT use DEPS context for POS-tagging and chunking tasks, since their labels are used in

[8]CoNLL 2000 shared task http://www.cnts.ua.ac.be/conll2000/chunking

[9]CoNLL 2003 shared task http://www.cnts.ua.ac.be/conll2003/ner

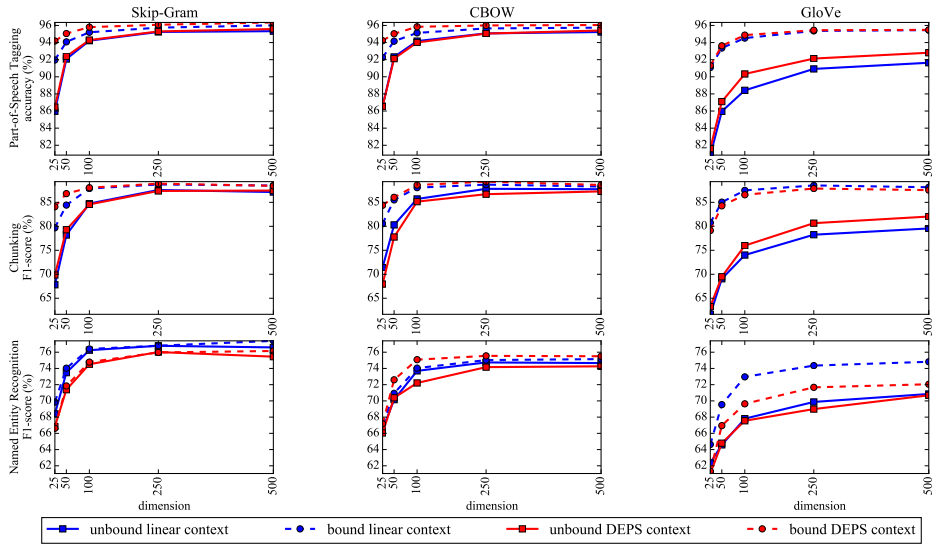

Figure 4: Accuracy or F_1-score results on Part-of-Speech Tagging, Chunking and Named Entity Recognition tasks.

parsing the source corpus.

Following the evaluation protocol used in Kiros et al. (2015), we restrict the predicting model to Logistic Regression Classifier[10]. The classifier's input for predicting the label of word w_i is simply the concatenation of word vectors $\vec{w_{i-2}}$, $\vec{w_{i-1}}$, $\vec{w_i}$, $\vec{w_{i+1}}$, $\vec{w_{i+2}}$. This ensures that the quality of embedding models is directly evaluated, and their strengths and weaknesses are easily observed.

Model	Context Type	Context Representation	POS	Chunking	NER
GSG	linear	unbound	95.3	87.2	76.6
		bound	96.0	**88.5**	**77.4**
	dep	unbound	95.6	87.5	75.5
		bound	**96.3**	**88.5**	76.2
GBOW	linear	unbound	95.2	87.7	74.7
		bound	95.7	88.3	75.2
	dep	unbound	95.4	87.3	74.3
		bound	**96.0**	**88.6**	**75.5**
GloVe	linear	unbound	91.6	79.6	70.8
		bound	**95.5**	**88.2**	**74.8**
	dep	unbound	92.8	82.0	70.7
		bound	**95.5**	87.5	72.0

Table 6: Numerical results on Part-of-Speech Tagging, Chunking and Named Entity Recognition tasks. Best results in group are marked **Bold**.

As shown in Figure 4 and Table 6, GSG, GBOW and GloVe exhibit overall similar trends. When the same context type is used, bound representation outperforms unbound representation on all tasks. Sequence labeling tasks are not sensitive to syntax. For bound representation, the ignorance of syntax becomes beneficial, since it decreases the amount of noise and sparsity.

Moreover, DEPS context type works slightly better than linear context type in most cases. These results suggest that unbound linear context (as in traditional CSG and CBOW) may not be the best choice of input word vectors for sequence labeling. Bound representations should always be used and DEPS context type is also worth considering. Again, similar to the word analogy task, GloVe is more sensitive to different context representations than Skip-Gram and CBOW.

4.5 Text Classification Task

Finally, we evaluate the effectiveness of different word embedding models with different syntactic contexts on text classification task. Text classification is one of the most popular and well-studied tasks in natural language processing. Recently, deep neural networks achieve state-of-the-art results on this task (Socher et al., 2013; Kim, 2014; Dai and Le, 2015). They often need pre-trained word embeddings as inputs to improve their performances. Similarly to the previous evaluation of sequence labeling tasks, instead of building complex deep neural networks, we use a simpler classification method called Neural Bag-of-Words (Li et al., 2017) to directly evaluate the word embeddings: texts are first represented by the sum of their word vectors, then a Logistic Regression Classifier (the same as that in previous subsection)

[10]The implementation by scikit is used http://scikit-learn.org/

Model	Context Type	Context Rep.	Sentence-level			Document-level	
			MR	CR	Subj	RT-2k	IMDB
GSG	linear	unbound	**76.1**	78.3	**90.9**	83.5	**85.2**
		bound	75.3	**79.0**	90.4	82.2	**85.2**
	dep	unbound	76.0	77.7	90.7	**84.8**	85.1
		bound	75.0	77.5	90.0	84.7	84.5
GBOW	linear	unbound	74.9	77.9	**90.4**	82.0	**85.0**
		bound	74.1	77.8	90.3	80.7	84.1
	dep	unbound	**75.0**	77.6	90.1	**82.4**	84.9
		bound	73.5	**78.2**	89.9	80.7	83.4
GloVe	linear	unbound	73.4	76.7	89.6	79.2	**83.5**
		bound	73.2	77.5	**90.0**	79.8	83.4
	dep	unbound	**74.0**	**77.7**	89.5	**81.3**	**83.5**
		bound	72.5	76.7	88.8	79.2	**83.5**
random word embeddings			63.9	72.8	79.9	72.2	77.2

Table 7: Accuracy results on 5 text classification datasets. Best results in group are **Bold**

is built upon these text representations for classification.

Different word embedding models are evaluated on 5 text classification datasets. The first 3 datasets are sentence-level: short movie review sentiment (MR) (Pang and Lee, 2005), customer product reviews (CR) (Nakagawa et al., 2010), and subjectivity/objectivity classification (SUBJ) (Pang and Lee, 2004). The other 2 datasets are document-level with multiple sentences: full-length movie review (RT-2k) (Pang and Lee, 2004), and IMDB movie review (IMDB) (Maas et al., 2011)[11].

As shown in Table 7, pre-trained word embeddings outperform random word embeddings by a large margin. This strengthens the previous claim that pre-trained word embeddings are highly useful for text classification (Iyyer et al., 2015; Li et al., 2017). Unlike in the other tasks, in text classification all models exhibit similar performance. Text classification has less focus on syntax and function similarity. Because of that, models with bound representation perform worse than those with unbound representation on almost all datasets except CR. Models with DEPS context type and linear context type are comparable. These observations suggest that simple unbound linear context type (as in traditional CSG and CBOW) is still the best choice of pre-training word embeddings for text classification, which is already used in most studies.

5 Conclusion

This paper provides a first systematical investigation of different syntactic context types (linear vs dependency-based) and different context representations (bound vs unbound) for learning word embeddings. We evaluate GSG, GBOW and GloVe models on intrinsic property analysis tasks (word similarity and word analogy), sequence labeling tasks (POS, Chunking and NER) and text classification task.

We find that most tasks have clear preference for different context types and representations. Context representation plays a more important role than context type for learning word embeddings. Only with the "help" of bound representation does DEPS context capture functional similarity. Word analogies seem to prefer unbound representation, although performance varies by question type No matter which syntactic context type is used, bound representation is essential for sequence labeling tasks, which benefits from its ability of capturing functional similarity. GSG with unbound linear context is still the best choice for text classification task. Linear context is sufficient for capturing topical similarity compared to more labor-intensive DEPS context. Words' position information is generally useless for text classification, which makes bound representation contribute less to this task.

In the spirit of transparent and reproducible experiments, the `word2vecPM` toolkit [12] is published along with this paper. We hope researchers will take advantage of the code for further improvements and applications to other tasks.

Acknowledgments

We thank the authors of the `word2vecf` toolkit and the accompanied evaluation scripts (Levy et al., 2015). Their work systematically investigated the effects of different hyper-parameters on various word embedding models, which directly influenced us a lot.

This work is supported by National Natural Science Foundation of China with grant No. 61472428, the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 14XNLQ06. This work is partially supported by ECNU-RUC-InfoSys Joint Data Science Lab and CCF-Tencent Open Research Fund (RAGR20160102).

[11] Please see Wang and Manning (2012) for more detailed introduction and pre-processing of these datasets.

[12] The current version can be found at `https://github.com/libofang/word2vecPM`. The entire code is also planned to be re-written in Python and integrated into VSMlib (`http://vsm.blackbird.pw/`) for easier use.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL*, pages 19–27. Association for Computational Linguistics.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *ACL*, pages 809–815.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *ACL*, pages 136–145. Association for Computational Linguistics.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *ACL*, pages 406–414.

Stephen Clark. 2012. Vector space models of lexical meaning. In *Handbook ofContemporary Semantics second edition.Wiley-Blackwell*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

James Richard Curran. 2004. *From distributional to semantic similarity*. Ph.D. thesis, University of Edinburgh, UK.

Andrew M. Dai and Quoc V. Le. 2015. Semi-supervised sequence learning. In *NIPS*, pages 3079–3087.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *COLING*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *WWW*, pages 406–414. ACM.

Anna Gladkova and Aleksandr Drozd. 2016. Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of The 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 36–42, Berlin, Germany. ACL.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesnt. In *Proceedings of naacl-hlt*, pages 8–15.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Kris Heylen, Yves Peirsman, Dirk Geeraerts, and Dirk Speelman. 2008. Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In *LREC*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL*, pages 1681–1691.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

Siwei Lai, Kang Liu, Shizhu He, and Jun Zhao. 2016. How to generate a good word embedding. *IEEE Intelligent Systems*, 31:5–14.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *ACL*, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *CoNLL*, pages 171–180.

Omer Levy and Yoav Goldberg. 2014c. Neural word embedding as implicit matrix factorization. In *NIPS*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Bofang Li, Tao Liu, Zhe Zhao, Puwei Wang, and Xiaoyong Du. 2017. Neural bag-of-n-grams. In *AAAI*, pages 3067–3074.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *HLT-NAACL*, pages 1299–1304.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. *CoRR*, abs/1606.07736.

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, pages 55–60.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The role of context types and dimensionality in learning word embeddings. In *HLT-NAACL*, pages 1030–1040.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751.

Andriy Mnih and Geoffrey E. Hinton. 2007. Three new graphical models for statistical language modelling. In *ICML*, pages 641–648.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *NAACL*, pages 786–794. Association for Computational Linguistics.

Neha Nayak and Christopher D. Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. In *ACL workshop*.

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

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, pages 271–278. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Tobias Schnabel, Igor Labutov, David M. Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *EMNLP*, pages 649–657.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, volume 1631, page 1642. Citeseer.

Jun Suzuki and Masaaki Nagata. 2015. A unified learning framework of skip-grams and global vectors. In *ACL*, page 186.

Ivan Vulic and Anna Korhonen. 2016. Is "universal syntax" universally useful for learning distributed word representations? In *ACL*, page 518.

Sida I. Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *EMNLP-CoNLL*, pages 940–951.

Wenpeng Yin and Hinrich Schutze. 2016. Learning word meta-embeddings. In *ACL*, pages 327–332.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using wiktionary for computing semantic relatedness. In *AAAI*, volume 8, pages 861–866.