Redefining Context Windows for Word Embedding Models: An Experimental Study

Pierre Lison
Norwegian Computing Center
Oslo, Norway
plison@nr.no

Andrei Kutuzov
Language Technology Group
University of Oslo
andreku@ifi.uio.no

Abstract

Distributional semantic models learn vector representations of words through the *contexts* they occur in. Although the choice of context (which often takes the form of a sliding window) has a direct influence on the resulting embeddings, the exact role of this model component is still not fully understood. This paper presents a systematic analysis of context windows based on a set of four distinct hyper-parameters. We train continuous Skip-Gram models on two English-language corpora for various combinations of these hyper-parameters, and evaluate them on both lexical similarity and analogy tasks. Notable experimental results are the positive impact of cross-sentential contexts and the surprisingly good performance of right-context windows.

1 Introduction

Distributional semantic models represent words through real-valued vectors of fixed dimensions, based on the distributional properties of these words observed in large corpora. Recent approaches such as prediction-based models (Mikolov et al., 2013b) and *GloVe* (Pennington et al., 2014) have shown that it is possible to estimate dense, low-dimensional vectors (often called *embeddings*) able to capture various functional or topical relations between words. These embeddings are used in a wide range of NLP tasks, including part-of-speech tagging, syntactic parsing, named entity recognition and semantic role labelling; see (Collobert et al., 2011; Lin et al., 2015; Zhou and Xu, 2015; Lample et al., 2016), among others.

As recently shown by (Levy et al., 2015), the empirical variations between embedding models are largely due to differences in hyper-parameters (many of which are tied to the underlying definition of context) rather than differences in the embedding algorithms themselves. In this paper, we further develop their findings with a comprehensive analysis of the role played by context window parameters when learning word embeddings. Four specific aspects are investigated:

1. The *maximum size* of the context window;
2. The *weighting scheme* of context words according to their distance to the focus word;
3. The relative *position* of the context window (symmetric, left or right side);
4. The treatment of *linguistic boundaries* such as end-of-sentence markers.

The next section 2 provides a brief overview on word embeddings and context windows. Section 3 describes the experimental setup used to evaluate the influence of these four aspects. Finally, Section 4 presents and discusses the results.

2 Background

The works of (Bengio et al., 2003) and (Mikolov et al., 2013b) introduced a paradigm-shift for distributional semantic models with new prediction-based algorithms outperforming the existing count-based approaches (Baroni et al., 2014). The *word2vec* models from (Mikolov et al., 2013b), comprising the Continuous Skip-gram and the Continuous Bag-of-Words algorithms, are now a standard part of many NLP pipelines.

Despite their differences, all types of distributional semantic models require the definition of a context for each word observed in a given corpus. Given a set of (word, context) pairs extracted from the corpus, vector representations of words can be derived through various estimation methods, such as predicting words given their contexts

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 284–288,
Gothenburg, Sweden, 23-24 May 2017. ©2017 Linköping University Electronic Press

(CBOW), predicting the contexts from the words (Skip-Gram), or factorizing the log of their co-occurrence matrix (*GloVe*). In all of these approaches, the choice of context is a crucial factor that directly affects the resulting vector representations. The most common method for defining this context is to rely on a *window* centered around the word to estimate (often called the *focus word*)[1]. The context window thus determines which contextual neighbours are taken into account when estimating the vector representations.

The most prominent hyper-parameter associated to the context window is the maximum window size (i.e. the maximum distance between the focus word and its contextual neighbours). This parameter is the easiest one to adjust using existing software, which is why it is comparatively well studied. Larger windows are known to induce embeddings that are more 'topical' or 'associative', improving their performance on analogy test sets, while smaller windows induce more 'functional' and 'synonymic' models, leading to better performance on similarity test sets (Goldberg, 2016).

However, the context window is also affected by other, less obvious hyper-parameters. Inside a given window, words that are closer to the focus word should be given more weights than more distant ones. To this end, CBOW and Continuous Skip-gram rely on a dynamic window mechanism where the actual size of the context window is sampled uniformly from 1 to L, where L is the maximum window size. This mechanism is equivalent to sampling each context word w_j with a probability that decreases linearly with the distance $|j-i|$ to the focus word w_i:

$$\begin{aligned} P(w_j|w_i) &= \sum_{window=1}^{L} P(w_j|w_i, window)P(window) \\ &= \frac{1}{L}(L-|j-i|+1) \end{aligned}$$

where *window* is the actual window size (from 1 to L) sampled by the algorithm. Similarly, the co-occurrence statistics used by *GloVe* rely on harmonic series where words at distance d from the focus word are assigned a weight $\frac{1}{d}$. For example, with the window size 3, the context word at the position 2 will be sampled with the probability of $2/3$ in *word2vec* and the probability of $1/2$ in *GloVe*.

Another implicit hyper-parameter is the symmetric nature of the context window. The *word2vec* and *GloVe* models pay equivalent attention to the words to the left and to the right of the focus word. However, the relative importance of left or right contexts may in principle depend on the linguistic properties of the corpus language, in particular its word ordering constraints.

Finally, although distributional semantic models do not themselves enforce any theoretical limit on the boundaries of context windows, word embeddings are in practice often estimated on a sentence by sentence basis, thus constraining the context windows to stop at sentence boundaries. However, to the best of our knowledge, there is no systematic evaluation of how this sentence-boundary constraint affects the resulting embeddings.

3 Experimental setup

To evaluate how context windows affect the embeddings, we trained Continuous Skip-gram with Negative Sampling (SGNS) embeddings for various configurations of hyper-parameters, whose values are detailed in Table 1. In particular, the "weighting scheme" encodes how the context words should be weighted according to their distance with the focus word. This hyper-parameter is given two possible values: a linear weighting scheme corresponding to the default *word2vec* weights, or an alternative scheme using the squared root of the distance.

Hyper-parameter	**Possible values**
Max. window size	$\{1, 2, 5, 10\}$
Weighting scheme	$\{\frac{L-d+1}{L}, \frac{L-\sqrt{d}+1}{L}\}$
Window position	{left, right, symmetric}
Cross-sentential	{yes, no}
Stop words removal	{yes, no}

Table 1: Range of possible hyper-parameter values evaluated in the experiment.

The embeddings were trained on two English-language corpora: Gigaword v5 (Parker et al., 2011), a large newswire corpus of approx. 4 billion word tokens, and the English version of OpenSubtitles (Lison and Tiedemann, 2016), a large repository of movie and TV subtitles, of approx. 700 million word tokens. The two corpora correspond to distinct linguistic genres, Gigaword being a corpus of news documents (average sentence length

[1] Other types of context have been proposed, such as dependency-based contexts (Levy and Goldberg, 2014) or multilingual contexts (Upadhyay et al., 2016), but these are outside the scope of the present paper.

21.7 tokens) while OpenSubtitles is a conversational corpus (average sentence length 7.3 tokens). OpenSubtitles notably contains a large number of non-sentential utterances, which are utterances lacking an overt predicate and depend on the surrounding dialogue context for their interpretation (Fernández, 2006). The corpora were lemmatized and POS-tagged with the Stanford CoreNLP (Manning et al., 2014) and each token was replaced with its lemma and POS tag. Two versions of the corpora were used for the evaluation: one raw version with all tokens, and one filtered version after removal of stop words and punctuation.

The word embeddings were trained with 300-dimensional vectors, 10 negative samples per word and 5 iterations. Very rare words (less than 100 occurrences in Gigaword, less than 10 in OpenSubtitles) were filtered out. The models were then evaluated using two standard test workflows: Spearman correlation against SimLex-999 semantic similarity dataset (Hill et al., 2015) and accuracy on the semantic sections of the Google Analogies Dataset (Mikolov et al., 2013a).

4 Results

All in all, we trained 96 models on Gigaword (GW) and 96 models on OpenSubtitles (OS)[2]. Figure 1 illustrates the results for the SGNS embeddings on lexical similarity and analogy tasks using various types of context windows. The main findings from the experiments are as follows.

Window size

As expected for a lexical similarity task (Schütze and Pedersen, 1993), narrow context windows perform best with the SimLex999 dataset, which contains pairs of semantically similar words (not just related). For the analogy task, larger context windows are usually beneficial, but not always: the word embeddings trained on OpenSubtitles perform best with the window size of 10, while the best results on the analogy task for Gigaword are obtained with the window size of 2.

Window position

Table 2 shows how the position of the context window influences the average model performance. Note that symmetric windows of, for instance, 10 are in fact 2 times larger than the 'left' or 'right' windows of the same size, as they consider 10 words both to the left and to the right of the focus word. This is most likely why symmetric windows consistently outperform 'single-sided' ones on the analogy task, as they are able to include twice as much contextual input.

Window position	SimLex999	Analogies
OS left	0.40	0.35
OS right	0.43	0.35
OS symmetric	0.43	**0.45**
GW left	0.43	0.64
GW right	0.44	0.65
GW symmetric	0.45	**0.68**

Table 2: Average performance across all models depending on the window position.

However, the average performance on the semantic similarity task (as indicated by the Spearman correlation with the SimLex999 test set) does not exhibit the same trend. 'Left' windows are indeed worse than symmetric ones, but 'right' windows are on par with the symmetric windows for OpenSubtitles and only one percent point behind them for Gigaword. It means that in many cases (at least with English texts) taking into account only *n* context words to the right of the focus word is sufficient to achieve the same performance with SimLex999 as by using a model which additionally considers *n* words to the left, and thus requires significantly more training time.

Cross-sentential contexts

The utility of cross-sentential contexts depends on several covariates, most importantly the type of corpus and the nature of the evaluation task. For similarity tasks, cross-sentential contexts do not seem useful, and can even be detrimental for large window sizes. However, for analogy tasks, cross-sentential contexts lead to improved results thanks to the increased window it provides. This is especially pronounced for corpora with short sentences such as OpenSubtitles (see Table 3).

Weighting scheme

Our experimental results show that none of the two evaluated weighting schemes (with weights that decrease respectively linearly or with the square-root of the distance) gives a consistent advantage averaged across all models. However, the squared weighting scheme is substantially slower (as it

[2] Encompassing different values of window size, weighting scheme, window position, cross-sentential boundaries and stop-words removal ($4 \times 2 \times 3 \times 2 \times 2 = 96$).

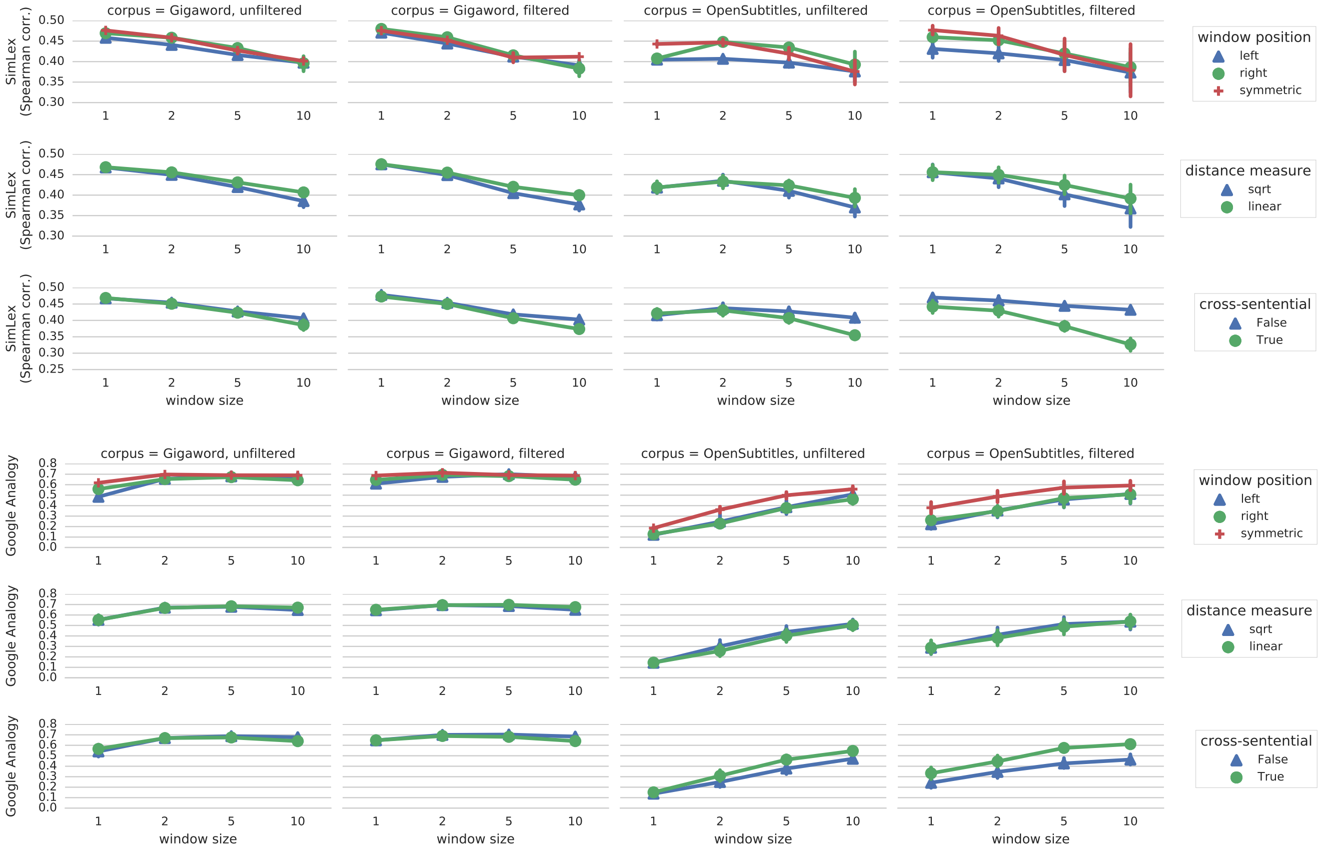

Figure 1: Results for the SGNS word embeddings trained with various types of context windows.

Cross-sentential	SimLex999	Analogies
OS False	**0.44**	0.34
OS True	0.40	**0.43**
GW False	0.44	0.66
GW True	0.44	0.65

Table 3: Average performance across all models with and without cross-sentential contexts.

Stop words removal	SimLex999	Analogies
OS no removal	0.41	0.34
OS with removal	0.42	**0.43**
GW no removal	0.44	0.64
GW with removal	0.44	**0.68**

Table 4: Average performance across all models depending on the removal of stop words.

increases the number of context words to consider for each focus word), decreasing the training speed about 25% with window size 5. Thus, the original linear weighting scheme proposed in (Mikolov et al., 2013b) should be preferred.

Stop words removal

As shown in Table 4, the removal of stop words does not really influence the average model performance for the semantic similarity task. The analogy task, however, benefits substantially from this filtering, for both corpora. Although not shown in the table, filtering stop words also significantly decreases the size of the corpus, thereby reducing the total time needed to train the word embeddings.

5 Conclusion

Our experiments demonstrate the importance of choosing the right type of context window when learning word embedding models. The two most prominent findings are (1) the positive role of cross-sentential contexts and (2) the fact that, at least for English corpora, right-side contexts seem to be more important than left-side contexts for similarity tasks, and achieve a performance comparable to that of symmetric windows.

In the future, we wish to extend this study to the CBOW algorithm, to other weighting schemes (such as the harmonic series employed by *GloVe*), and to non-English corpora.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

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

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

Raquel Fernández. 2006. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. Ph.D. thesis, King's College London.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, pages 302–308.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. 2015. Unsupervised pos induction with word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316, Denver, Colorado, May–June. Association for Computational Linguistics.

P. Lison and J. Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 923–929.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, June. LDC2011T07.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Hinrich Schütze and Jan Pedersen. 1993. A vector model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the UW Centre for the New OED and Text Research*, pages 104–113.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proc. of ACL*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1127–1137, Beijing, China, July. Association for Computational Linguistics.