A Simple Word Embedding Model for Lexical Substitution

Oren Melamud **Omer Levy** **Ido Dagan**
Department of Computer Science
Bar-Ilan University
Ramat-Gan, Israel
{melamuo,omerlevy,dagan}@cs.biu.ac.il

Abstract

The lexical substitution task requires identifying meaning-preserving substitutes for a target word instance in a given sentential context. Since its introduction in SemEval-2007, various models addressed this challenge, mostly in an unsupervised setting. In this work we propose a simple model for lexical substitution, which is based on the popular skip-gram word embedding model. The novelty of our approach is in leveraging explicitly the *context embeddings* generated within the skip-gram model, which were so far considered only as an internal component of the learning process. Our model is efficient, very simple to implement, and at the same time achieves state-of-the-art results on lexical substitution tasks in an unsupervised setting.

1 Introduction

Lexical substitution tasks have become very popular for evaluating context-sensitive lexical inference models since the introduction of the original task in SemEval-2007 (McCarthy and Navigli, 2007) and additional later variants (Biemann, 2013; Kremer et al., 2014). In these tasks, systems are required to predict substitutes for a target word instance, which preserve its meaning in a given sentential context. Recent models addressed this challenge mostly in an unsupervised setting. They typically generated a word instance representation, which is biased towards its given context, and then identified substitute words based on their similarity to this biased representation. Various types of models were proposed, from sparse syntax-based vector models (Thater et al., 2011), to probabilistic graphical models (Moon and Erk, 2013) and LDA topic models (Ó Séaghdha and Korhonen, 2014).

Word embeddings are low-dimensional vector representations of word types that recently gained much traction in various semantic tasks. Probably the most popular word embedding model today is *skip-gram*, introduced in Mikolov et al. (2013) and available as part of the *word2vec* toolkit.[1] word2vec learns for every word type two distinct representations, one as a target and another as a context, both embedded in the same space. However, the context representations are considered internal to the model and are discarded after training. The output word embeddings represent context-insensitive target word types.

Few recent models extended word embeddings by learning a distinct representation for each sense of a target word type, as induced by clustering the word's contexts (Huang et al., 2012; Neelakantan et al., 2014). They then identify the relevant sense(s) for a given word instance, in order to measure context-sensitive similarities. Although these models may be considered for lexical substitution, they have so far been applied only to 'softer' word similarity tasks which include topical relations.

In this work we propose a simple approach for *directly* utilizing the skip-gram model for context-sensitive lexical substitution. Instead of discarding the learned context embeddings, we use them in conjunction with the target word embeddings to model target word instances. A suitable substitute for a

[1] https://code.google.com/p/word2vec/

target word instance is then identified via its combined similarity to the embeddings of both the target and its given context. [2] Our model is efficient, can be implemented literally in a few lines of code, and at the same time achieves state-of-the-art results on two lexical substitution datasets in an unsupervised setting.

2 Skip-gram Word Embeddings

In this section we provide technical background on skip-gram embeddings, which are used in our model. As mentioned, skip-gram embeds both target words and contexts in the same low-dimensional space. In this space, the vector representations of a target and context are pushed closer together the more frequently they co-occur in a learning corpus. Thus, the Cosine distance between them can be viewed as a first-order *target-to-context* similarity measure, indicative of their syntagmatic compatibility. Indirectly, this also results in assigning similar vector representations to target words that share similar contexts, thereby suggesting the Cosine distance between word embeddings as a second-order *target-to-target* distributional similarity measure.

word2vecf[3] (Levy and Goldberg, 2014a) is an extension of the skip-gram implementation in word2vec, which supports arbitrary types of contexts rather than only word window contexts. Levy and Goldberg (2014a) used word2vecf to produce syntax-based word embeddings, where context elements are the syntactic contexts of the target words. Specifically, for a target word t with modifiers m_1,...,m_k and head h, they considered the context elements (m_1, r_1),...,(m_k, r_k),(h, r_h^{-1}), where r is the type of the ('collapsed') dependency relation between the head and the modifier (e.g. *dobj*, *prep_of*) and r^{-1} denotes an inverse relation. Similarly to traditional syntax-based vector space models (Padó and Lapata, 2007), they show that these embeddings tend to capture functional word similarity (as in *manage* ∼ *supervise*) rather than topical similarity or relatedness (as in *manage* ∼ *manager*). Figure 1 illustrates a syntax-based embedding space using t-SNE (Van der Maaten and Hinton, 2008), which visualizes the similarities in the original higher-dimensional space.

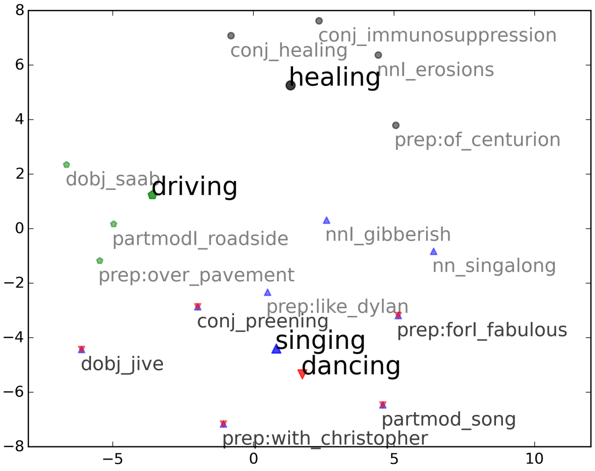

Figure 1: A 2-dimensional visualization of the gerunds *singing*, *dancing*, *driving*, and *healing* with their top syntactic contexts in an embedded space. *singing* and *dancing* share many similar contexts (e.g. *partmod_song* and *dobj_jive*) and therefore end up with very similar vector representations.

3 Lexical Substitution Model

Our model is based on the natural assumption that a good lexical substitute for a target word instance, under a given context, needs to be both (1) semantically similar to the target word and (2) compatible with the given context. Hence, we wish to propose a context-sensitive substitutability measure for potential substitutes, which reflects a combination of the above. We estimate the semantic similarity between a substitute word and the target word using a second-order target-to-target similarity measure, and the compatibility of a substitute word with the given context using a first-order target-to-context similarity measure. Conveniently, as described in Section 2, both target-to-target and target-to-context similarities can be estimated by the vector Cosine distance between the respective skip-gram embeddings, i.e. using both target word embeddings and the 'internal' context embeddings. Specifically, we choose syntax-based skip-gram embeddings (Levy

[2] While in this work we focus on skip-gram embeddings, we note that there are also other potentially relevant word embedding methods that can generate context representations in addition to the 'standard' target word representations. See, for example, GloVe (Pennington et al., 2014) and SVD-based methods (Levy et al., 2015).

[3] `https://bitbucket.org/yoavgo/word2vecf`

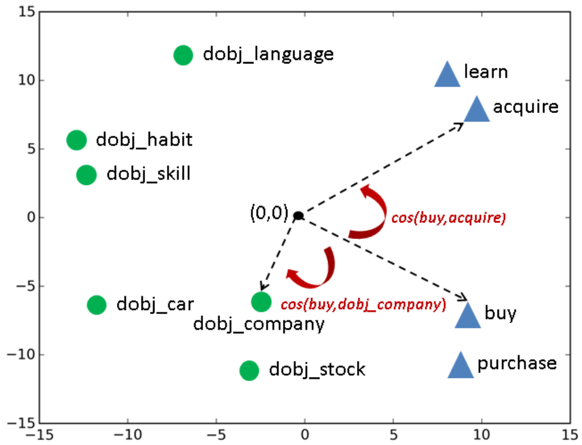

Figure 2: Identifying substitutes for the target word *acquire* under the syntactic context *dobj_company*, visualized in a 2-dimensional embedded space. Even though *learn* is the closest word to *acquire*, the word *buy* is both reasonably close to *acquire* as well as to the context *dobj_company* and is therefore considered a better substitute.

Add	$\frac{cos(s,t)+\sum_{c\in C} cos(s,c)}{\vert C\vert+1}$
BalAdd	$\frac{\vert C\vert\cdot cos(s,t)+\sum_{c\in C} cos(s,c)}{2\cdot\vert C\vert}$
Mult	$\sqrt[\vert C\vert+1]{pcos(s,t)\cdot\prod_{c\in C} pcos(s,c)}$
BalMult	$\sqrt[2\cdot\vert C\vert]{pcos(s,t)^{\vert C\vert}\cdot\prod_{c\in C} pcos(s,c)}$

Table 1: The different substitutability measures considered in our model for a lexical substitute s of the target word t in sentential context C. C is represented by the set of the target word's context elements in the context sentence, where c denotes an individual context element. cos is the vector Cosine function applied to the vector representations of the words or contexts, and $pcos(v, v') = \frac{cos(v,v')+1}{2}$ is used to avoid negative values in Mult and BalMult.

and Goldberg, 2014a) since lexical substitutes need to exhibit strict functional similarity to the target word. Figure 2 illustrates our approach.

We next describe the details of our model. Our model introduces a context-sensitive substitutability measure (or metric) for estimating the suitability of a lexical substitute for a target word in a given sentential context. This measure weighs the semantic similarity score between the substitute and the target word type, together with one or more context compatibility scores, estimating the compatibility of the substitute with each of the target's context elements in the given sentential context.

To keep our method as simple as possible we do not employ any tunable weighting parameters to optimize our proposed measure. Instead, we choose to focus only on evaluating the four measure variants described in Table 1. These measures reflect two basic metric design choices. The first choice is between using an arithmetic mean (as in Add and BalAdd) and a geometrical mean (as in Mult and BalMult) to combine the score elements together. These are two common methods, which were recently investigated in the context of analogy detection tasks (Levy and Goldberg, 2014b). The multiplicative combinations, Mult and BalMult, reflect a stricter logical 'AND-like' approach requiring high similarities in all elements of the product to get a high score. In particular, they reward substitutes that show substantial similarity to both the target word and each of the context elements. In contrast, the additive combinations, Add and BalAdd, can yield a high score even if one of the elements in the sum is zero.

The second design choice concerns the relative contribution of the context compatibility component with respect to the word similarity component. In Add and Mult, the relative weight of context compatibility with respect to word similarity becomes greater the more there are context elements for the target word in the given context sentence. In contrast, the balanced combinations, BalAdd and BalMult, keep an equal balance between these two factors, under the hypothesis that the overall contribution of the context compatibility should be fixed regardless to the number of context elements.

As a motivating example, Table 2 shows how our model uses a single informative context element to generate context-sensitive lexical substitutions for polysemous target words.

4 Evaluation

In the original lexical substitution task (McCarthy and Navigli, 2007) all of the participating sys-

$t = jaguar$		$t = cool$		$t = employ$	
$c = poss^{-1}\ engine$	$c = poss^{-1}\ paws$	$c = amod^{-1}\ outfit$	$c = amod^{-1}\ weather$	$c = dobj\ technique$	$c = dobj\ specialist$
daimler	cheetah	preppy	wintery	employs	employing
lancia	tiger	old-skool	drizzly	employing	employs
maserati	puma	kick-ass	spring-like	employed	recruit
bmw	leopard	sexy	unseasonably	adopt	employed
rover	jaguars	chilled-out	warm	utilise	appoint
daihatsu	cat	snazzy	balmy	using	redeploy
lamborghini	hyena	funky	hot	utilize	remunerate
volvo	wildcat	super-cool	anticyclonic	re-learn	recruited
gt6	panda	half-decent	cooler	adopting	recruiting

Table 2: The top lexical substitutes for example target words t under different syntactic contexts c, using Mult.

tems predicted substitutes by first using manually-constructed thesauri to generate substitute candidates and then developing candidate ranking models to choose the most appropriate ones. Later works focused mostly on the candidate ranking part, where candidates are provided as part of the datasets. In this section we present the evaluation of our model both on the substitute candidates ranking task, and on the original substitutes prediction task (no candidates provided), using two different lexical substitution datasets.

4.1 Lexical substitutions datasets

The dataset introduced in the lexical substitution task of SemEval-2007 (McCarthy and Navigli, 2007), denoted here LS-SE, is the most widely used for the evaluation of lexical substitution. It consists of 10 sentences extracted from a web corpus for each of 201 target words (nouns, verbs, adjectives and adverbs), or altogether 2,010 word instances in sentential context, split into 300 trial sentences and 1,710 test sentences. The gold standard provided with this dataset is a weighted lemmatized substitute list for each word instance, based on manual annotations.

A more recent large-scale 'all-words' dataset, called 'Concepts in Context', was introduced in Kremer et al. (2014) and denoted here LS-CIC. This dataset provides the same kind of data as LS-SE, but instead of choosing specific target words that tend to be ambiguous as done in LS-SE, the target words here are all the content words in text documents extracted from news and fiction corpora, and are therefore more naturally distributed. LS-CIC is also much larger than LS-SE with over 15K target word instances.

4.2 Compared methods

We used ukWaC (Ferraresi et al., 2008), a two billion word web corpus, as our learning corpus. We parsed both ukWaC and the sentences in the lexical substitution datasets with Stanford's Neural Network Dependency Parser (Chen and Manning, 2014).[4] Following Levy and Goldberg (2014a), we learned syntax-based skip-gram word and context embeddings using word2vecf (with 600 dimensions and 15 negative sampling), converting all tokens to lowercase, discarding words and syntactic contexts that appear less than 100 times in the corpus and 'collapsing' dependencies that include prepositions. This resulted in a vocabulary of about 200K word embeddings and 1M context embeddings. [5] Finally, for every instance in the lexical substitution datasets, we extracted the syntactic contexts of the target word and used each of our measures, Add, BalAdd, Mult and BalMult, to compute the substitute scores. In addition to our measures, we evaluated as a baseline a context-insensitive method, denoted Base, assigning scores according to the Cosine similarity between the target and the substitute word embeddings, ignoring the context. We also compare our results to the state-of-the-art.

4.3 Candidate ranking task

Most works that used the LS-SE dataset after SemEval-2007, as well as the one on LS-CIC, focused on ranking substitute candidates. They obtained the set of substitute candidates for a target type by pooling the annotated gold-standard substi-

[4] http://nlp.stanford.edu/software/nndep.shtml

[5] Our embeddings are available at: www.cs.biu.ac.il/nlp/resources/downloads/lexsub_embeddings/

Method	Resources	LS-SE	LS-CIC
Mult	ukWaC	**53.6**	**48.1**
BalMult		51.5	48.3
Add		**52.9**	**48.3**
BalAdd		50.3	48.0
Base		44.9	46.2
Random	n/a	30.0	33.8
Kremer, 2014†	Gigaword	52.5	47.8
Séaghdha, 2014	Wikipedia,BNC	49.5	n/a
Moon, 2013	ukWaC,BNC,WN	47.1	n/a
	Gigaword,WN	46.7	n/a
Szarvas, 2013	LLC,WN	55.0*	n/a

Table 3: GAP scores for compared methods on the candidate ranking task. Resources used by these methods: ukWaC, Wikipedia, Gigaword (Parker et al., 2011), WN = WordNet (Fellbaum, 2010), BNC = British National Corpus (Aston and Burnard, 1998), and LLC (Richter et al., 2006).

† A re-implementation of the model in Thater, 2011.

* Obtained by a supervised method.

tutes from all of its instances.[6] Furthermore, all of these works discarded multi-word expression substitutes from the gold standards, and omitted instances who thus remained with no gold substitutes.[7] The quality of the rankings with respect to the gold standard was measured using Generalized Average Precision (GAP) (Kishida, 2005). We follow the same evaluation settings for this task, using the substitute scores of our compared methods to rank the candidates.

Our results, compared with the most recent state-of-the-art, are illustrated in Table 3. First, we see that all of our methods yield significant performance gains over the context-insensitive Base baseline. Similarly to the behavior reported in (Kremer et al., 2014), this gain is much more substantial in LS-SE than in LS-CIC, which seems to be due to the target words in LS-SE being more ambiguous by construction. Next, we see that the non-balanced methods, Mult and Add, perform a little better on the LS-SE dataset. This suggests that giving more weight to context compatibility at the expense of word similarity is beneficial when ranking substitute candidates of ambiguous words. This can be justified considering that all the substitute candidates already bear some semantic similarity with the target by way of construction. Finally, the multiplicative combinations seem to perform slightly better than the additive ones on LS-SE.

In comparison to previous works our results are slightly better than state-of-the-art, with the exception of Szarvas et al. (2013). However, we note that Szarvas et al. (2013) is a supervised model, evaluated on the LS-SE gold standard with 10-fold cross validation and therefore is not directly comparable with unsupervised models, such as our own.

4.4 Substitute prediction task

In the original lexical substitution task of SemEval-2007, the organizers evaluated participant systems on their ability to predict the substitutes in the gold standard of the LS-SE test-set in a few subtasks (1) *best* and *best-mode* - evaluate the quality of the best predictions (2) *oot* and *oot-mode* (out of ten) - evaluate the coverage of the gold substitute list by the top ten best predictions.[8] We performed this evaluation on both the LS-SE and LS-CIC datasets, using our measures to predict the most suitable substitutes. We note that this task is a lot more challenging than the candidate ranking task, as it requires to identify the best substitutes out of the entire word vocabulary. To the best of our knowledge, Biemann and Riedl (2013), denoted here BR-2013, is the only prior work that reported such results on the LS-SE dataset, learning only from corpus data like we do. They used a syntax-based distributional thesaurus to generate a list of substitute candidates and then ranked the candidates according to their compatibility with the given context. As their learning corpus, they merged Gigaword (Parker et al., 2011) and LLC (Richter et al., 2006), which is similar in size to ours. Both Biemann and Riedl (2013) and our model do not attempt to identify and therefore always fail to predict the multi-word expression substitutes in the gold standard. There is no previously reported result

[6] A target type is defined as the pair (word lemma, pos), where pos $\in$ {noun, verb, adjective, adverb}.

[7] In cases where this procedure was not clearly described in the paper, we verified it with the authors.

[8] For brevity we do not describe the details of these subtasks. We report only *recall* scores as in this task *recall*=*precision* for all methods that predict substitutes to all of the instances in the dataset as we did.

Method	best	best-mode	oot	oot-mode
LS-SE test-set				
Mult	6.64	10.89	23.16	33.58
BalMult	**8.09**	**13.41**	**27.65**	**39.19**
Add	7.37	12.11	25.52	36.59
BalAdd	**8.14**	**13.41**	**27.42**	**39.11**
Base	7.81	13.41	23.38	33.98
BR-2013	n/a	n/a	27.48	37.19
LS-CIC				
Mult	4.08	8.36	15.07	26.02
BalMult	**5.51**	**11.72**	**19.59**	**33.32**
Add	4.82	9.97	17.11	29.48
BalAdd	**5.62**	**11.89**	**20.03**	**33.75**
Base	5.17	10.93	18.01	30.29

Table 4: *best* and *oot* subtasks scores for all compared methods on the substitute prediction task.

for this task on LS-CIC.

The results are shown in Table 4. In contrast to the candidate ranking task, we see that in the prediction task the balanced methods perform significantly better than the non-balanced ones. This suggests that in the absence of a substitute candidate 'oracle' it is important for the models to balance both word similarity and context compatibility. The balanced methods, BalAdd and BalMult, perform similarly, and show significant advantage over the context-insensitive Base baseline in the *oot* subtasks. On the *best* sub-tasks they show very little improvement. Finally, our results are on par with the results reported by Biemann and Riedl (2013).

5 Conclusions

In this paper we showed how the skip-gram model can be utilized directly to perform context-sensitive lexical substitution. This is achieved by exploiting its internally-learned context embeddings in conjunction with the 'standard' target word embeddings, to weigh context compatibility together with word similarity. Despite its simplicity, our model achieves state-of-the-art results on lexical substitution tasks using two different datasets.

Word embeddings in general, and skip-gram embeddings in particular, have recently become very popular in many NLP tasks since they achieve state-of-the-art performance, and at the same time are easy to use and efficient both in learning and inference time. Our work shows how these attractive properties can be easily carried over when addressing context-sensitive lexical substitution.

In future work, we hypothesize that our simple model may be further optimized. One reason to believe so is that although our balanced weighting methods showed robust performance across all the tasks in our evaluations, we did see that other strategies, which put more weight on context compatibility, achieve the best results in a substitute candidate ranking setting. This suggests that applications may benefit from adapting our model to the task at hand. For example, a possible direction is using a tunable weighting parameter for interpolating between the components of our substitutability measure.

Finally, while focusing on skip-gram embeddings in this work, it would be interesting to explore how well our approach generalizes to other types of embeddings that can represent both target words and contexts (Pennington et al., 2014; Levy et al., 2015).

Acknowledgments

This work was partially supported by the Israel Science Foundation grant 880/12 and the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1).

References

Guy Aston and Lou Burnard. 1998. *The BNC handbook: exploring the British National Corpus with SARA*. Capstone.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! A framework for lexical expansion with contextual similarity. *Journal of Language Modelling*, 1(1):55–95.

Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*.

Christiane Fellbaum. 2010. *WordNet*. Springer.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4)*.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of ACL*.

Kazuaki Kishida. 2005. *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. National Institute of Informatics Tokyo, Japan.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us-analysis of an all-words lexical substitution corpus. In *Proceedings of EACL*.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of ACL*.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL-2014*.

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.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of SemEval*.

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

Taesun Moon and Katrin Erk. 2013. An inference-based model of word meaning in context as a paraphrase distribution. *ACM Trans. Intell. Syst. Technol.*, 4(3):42:1–42:28, July.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of EMNLP*.

Diarmuid Ó Séaghdha and Anna Korhonen. 2014. Probabilistic distributional semantics with latent variable models. *Computational Linguistics*, 40(3):587–631.

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

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, june. *Linguistic Data Consortium, LDC2011T07*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the EMNLP*.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the Leipzig corpora collection. *Proceesings of the IS-LTC*.

György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. 2013. Learning to rank lexical substitutions. In *Proceedings of EMNLP*.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word meaning in context: A simple and effective vector model. In *Proceedings of IJCNLP*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.