A Deeper Look into Dependency-Based Word Embeddings

Sean MacAvaney
Information Retrieval Lab
Department of Computer Science
Georgetown University
Washington, DC, USA
sean@ir.cs.georgetown.edu

Amir Zeldes
Department of Linguistics
Georgetown University
Washington, DC, USA
amir.zeldes@georgetown.edu

Abstract

We investigate the effect of various dependency-based word embeddings on distinguishing between functional and domain similarity, word similarity rankings, and two downstream tasks in English. Variations include word embeddings trained using context windows from Stanford and Universal dependencies at several levels of enhancement (ranging from unlabeled, to Enhanced++ dependencies). Results are compared to basic linear contexts and evaluated on several datasets. We found that embeddings trained with Universal and Stanford dependency contexts excel at different tasks, and that enhanced dependencies often improve performance.

1 Introduction

For many natural language processing applications, it is important to understand word-level semantics. Recently, word embeddings trained with neural networks have gained popularity (Mikolov et al., 2013; Pennington et al., 2014), and have been successfully used for various tasks, such as machine translation (Zou et al., 2013) and information retrieval (Hui et al., 2017).

Word embeddings are usually trained using linear bag-of-words contexts, i.e. tokens positioned around a word are used to learn a dense representation of that word. Levy and Goldberg (2014) challenged the use of linear contexts, proposing instead to use contexts based on dependency parses. (This is akin to prior work that found that dependency contexts are useful for vector models (Pado and Lapata, 2007; Baroni and Lenci, 2010).) They found that embeddings trained this way are better at capturing semantic similarity, rather than relatedness. For instance, embeddings trained using linear contexts place *Hogwarts* (the fictional setting of the Harry Potter series) near *Dumbledore* (a character from the series), whereas embeddings trained with dependency contexts place *Hogwarts* near *Sunnydale* (fictional setting of the series Buffy the Vampire Slayer). The former is *relatedness*, whereas the latter is *similarity*.

Work since Levy and Goldberg (2014) examined the use of dependency contexts and sentence feature representations for sentence classification (Komninos and Manandhar, 2016). Li et al. (2017) filled in research gaps relating to model type (e.g., CBOW, Skip-Gram, GloVe) and dependency labeling. Interestingly, Abnar et al. (2018) recently found that dependency-based word embeddings excel at predicting brain activation patterns. The best model to date for distinguishing between similarity and relatedness combines word embeddings, WordNet, and dictionaries (Recski et al., 2016).

One limitation of existing work is that it has only explored one dependency scheme: the English-tailored Stanford Dependencies (De Marneffe and Manning, 2008b). We provide further analysis using the cross-lingual Universal Dependencies (Nivre et al., 2016). Although we do not compare cross-lingual embeddings in our study, we will address one important question for English: are Universal Dependencies, which are less tailored to English, actually better or worse than the English-specific labels and graphs? Furthermore, we investigate approaches to simplifying and extending dependencies, including Enhanced dependencies and Enhanced++ dependencies (Schuster and Manning, 2016), as well as two levels of relation simplification. We hypothesize that the cross-lingual generalizations from universal dependencies and the additional context from enhanced dependencies should improve the performance of word embeddings at distinguishing between functional and domain

Proceedings of NAACL-HLT 2018: Student Research Workshop, pages 40–45
New Orleans, Louisiana, June 2 - 4, 2018. ©2017 Association for Computational Linguistics

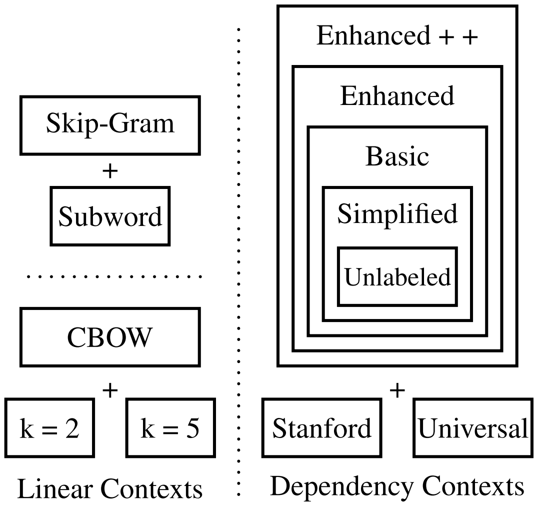

Figure 1: Visual relationship between types of embedding contexts. Each layer of enhancement adds more information to the dependency context (e.g., simplified adds dependency labels to the unlabeled context). We investigate CBOW using both a context window of $k = 2$ and $k = 5$, and we use the SkipGram model both with and without subword information.

similarity. We also investigate how these differences impact word embedding performance at word similarity rankings and two downstream tasks: question-type classification and named entity recognition.

2 Method

In this work, we explore the effect of two dependency annotation schemes on the resulting embeddings. Each scheme is evaluated in five levels of enhancement. These embeddings are compared to embeddings trained with linear contexts using the continuous bag of words (CBOW) with a context window of $k = 2$ and $k = 5$, and Skip-Gram contexts with and without subword information. These configurations are summarized in Figure 1.

Two dependency annotation schemes for English are Stanford dependencies (De Marneffe and Manning, 2008b) and Universal dependencies (Nivre et al., 2016). Stanford dependencies are tailored to English text, including dependencies that are not necessarily relevant cross-lingually (e.g. a label *prt* for particles like *up* in *pick up*). Universal dependencies are more generalized and designed to work cross-lingually. Many structures are similar between the two schemes, but important differences exist. For instance, in Stanford dependencies, prepositions head their phrase and depend on the modified word (*in* is the head of *in Kansas*), whereas in universal dependencies, prepositions depend on the prepositional object (*Kansas* dominates *in*). Intuitively, these differences should have a moderate effect on the resulting embeddings because different words will be in a given word's context.

Simp.	Basic
Stanford dependencies	
mod	poss, prt, predet, det, amod, tmod, npadvmod, possessive, advmod, quantmod, preconj, mark, vmod, nn, num, prep, appos, mwe, mod, number, neg, advcl, rcmod
arg	agent, iobj, dobj, acomp, pcomp, pobj, ccomp, arg, subj, csubj, obj, xcomp, nsubj
aux	aux, cop
sdep	xsubj, sdep
Universal dependencies	
core	iobj, dobj, ccomp, csubj, obj, xcomp, nsubj
ncore	discourse, cop, advmod, dislocated, vocative, aux, advcl, mark, obl, expl
nom	case, nmod, acl, neg, appos, det, amod, nummod
coord	cc, conj
special	goeswith, reparandum, orphan
loose	parataxis, list
mwe	compound, mwe, flat
other	punct, dep, root

Table 1: Simplified Stanford and Universal dependency labels. For simplified dependencies, basic labels are collapsed into the simplified label shown in this table. (Relations not found in this table were left as is.)

We also investigate five levels of enhancement for each dependency scheme. *Basic* dependencies are the core dependency structure provided by the scheme. *Simplified* dependencies are more coarse basic dependencies, collapsing similar labels into rough classes. The categories are based off of the Stanford Typed Dependencies Manual (De Marneffe and Manning, 2008a) and the Universal Dependency Typology (De Marneffe et al., 2014), and are listed in Table 1. Note that the two dependency schemes organize the relations in different ways, and thus the two types of simplified dependencies represent slightly different structures. The *unlabeled* dependency context removes all labels, and just captures syntactically adjacent tokens.

Enhanced and *Enhanced*++ dependencies (Schuster and Manning, 2016) address some practical dependency distance issues by extending basic dependency edges. Enhanced dependencies augment modifiers and conjuncts with their parents' labels, propagate governors and dependents for indirectly governed arguments, and add subjects to controlled verbs.

Enhanced++ dependencies allow for the deletion of edges to better capture English phenomena, including partitives and light noun constructions, multi-word prepositions, conjoined prepositions, and relative pronouns.

3 Experimental Setup

We use the Stanford CoreNLP parser[1] to parse basic, Enhanced, and Enhanced++ dependencies. We use the Stanford english_SD model to parse Stanford dependencies (trained on the Penn Treebank) and english_UD model to parse Universal dependencies (trained on the Universal Dependencies Corpus for English). We acknowledge that differences in both the size of the training data (Penn Treebank is larger than the Universal Dependency Corpus for English), and the accuracy of the parse can have an effect on our overall performance. We used our own converter to generate simple dependencies based on the rules shown in Table 1. We use the modified `word2vecf` software[2] Levy and Goldberg (2014) that works with arbitrary embedding contexts to train dependency-based word embeddings.

As baselines, we train the following linear-context embeddings using the original `word2vec` software:[3] CBOW with $k = 2$, CBOW with $k = 5$, and Skip-Gram. We also train enriched Skip-Gram embeddings including subword information (Bojanowski et al., 2016) using fastText.[4]

For all embeddings, we use a cleaned recent dump of English Wikipedia (November 2017, 4.3B tokens) as training data. We evaluate each on the following tasks:

Similarity over Relatedness Akin to the quantitative analysis done by Levy and Goldberg (2014), we test to see how well each approach ranks similar items above related items. Given pairs of similar and related words, we rank each word pair by the cosine similarity of the corresponding word embeddings, and report the area-under-curve (AUC) of the resulting precision-recall curve. We use the labeled WordSim-353 (Agirre et al., 2009; Finkelstein et al., 2001) and the Chiarello dataset (Chiarello et al., 1990) as a source of similar and related word pairs. For WordSim-353, we only consider pairs with similarity/relatedness scores of at least 5/10, yielding 90 similar pairs and 147 related pairs. For Chiarello, we disregard pairs that are marked as both similar and related, yielding 48 similar pairs and 48 related pairs.

Ranked Similarity This evaluation uses a list of word pairs that are ranked by degree of functional similarity. For each word pair, we calculate the cosine similarity, and compare the ranking to that of the human-annotated list using the Spearman correlation. We use SimLex-999 (Hill et al., 2016) as a ranking of functional similarity. Since this dataset distinguishes between nouns, adjectives, and verbs, we report individual correlations in addition to the overall correlation.

Question-type Classification (QC) We use an existing QC implementation[5] that uses a bidirectional LSTM. We train the model with 20 epochs, and report the average accuracy over 10 runs for each set of embeddings. We train and evaluate using the TREC QC dataset (Li and Roth, 2002). We modified the approach to use fixed (non-trainable) embeddings, allowing us to compare the impact of each embedding type.

Named Entity Recognition (NER) We use the Dernoncourt et al. (2017) NER implementation[6] that uses a bidirectional LSTM. Training consists of a maximum of 100 epochs, with early stopping after 10 consecutive epochs with no improvement to validation performance. We evaluate NER using the F1 score on the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003). Like the QC task, we use a non-trainable embedding layer.

4 Results

4.1 Similarity over Relatedness

The results for the WordSim-353 (WS353) and Chiarello datasets are given in Table 2a. For the WS353 evaluation, notice that the Enhanced dependencies for both Universal and Stanford dependencies outperform the others in each scheme. Even the poorest-performing level of enhancement (unlabeled), however, yields a considerable gain over the linear contexts. Both Skip-Gram variants yield the worst performance, indicating that they

[1] `stanfordnlp.github.io/CoreNLP/`

[2] `bitbucket.org/yoavgo/word2vecf`

[3] `code.google.com/p/word2vec/`

[4] `github.com/facebookresearch/fastText`

[5] `github.com/zhegan27/sentence_classification`

[6] `github.com/Franck-Dernoncourt/NeuroNER`

	(a) Sim/rel (AUC)		(b) Ranked sim (Spearman)				(c) Downstream	
Embeddings	WS353	Chiarello	Overall	Noun	Adj.	Verb	QC (Acc)	NER (F1)
Universal embeddings								
Unlabeled	0.786	0.711	0.370	0.408	0.484	0.252	0.915	0.877
Simplified	0.805	0.774	0.394	0.420	0.475	0.309	0.913	0.870
Basic	0.801	0.761	0.391	0.421	0.451	0.331	0.920	0.876
Enhanced	**0.823**	**0.792**	0.398	0.416	0.473	**0.350**	0.915	0.875
Enhanced++	0.820	0.791	0.396	0.416	0.461	0.348	0.917	0.882
Stanford embeddings								
Unlabeled	0.790	0.741	0.382	0.414	**0.507**	0.256	0.911	0.870
Simplified	0.793	0.748	0.393	0.416	0.501	0.297	**0.923**	0.873
Basic	0.808	0.769	**0.402**	**0.422**	0.494	0.341	0.910	0.865
Enhanced	0.817	0.755	0.399	0.420	0.482	0.338	0.911	0.871
Enhanced++	0.810	0.764	0.398	0.417	0.496	0.346	0.918	0.878
Baselines (linear contexts)								
CBOW, k=2	0.696	0.537	0.311	0.355	0.338	0.252	0.913	0.885
CBOW, k=5	0.701	0.524	0.309	0.353	0.358	0.258	0.899	0.893
Skip-Gram	0.617	0.543	0.264	0.304	0.368	0.135	0.898	0.881
SG + Subword	0.615	0.456	0.324	0.358	0.451	0.166	0.897	**0.887**

Table 2: Results of various dependency-based word embeddings, and baseline linear contexts at (a) similarity over relatedness, (b) ranked similarity, and (c) downstream tasks of question classification and named entity recognition.

capture relatedness better than similarity. For the Chiarello evaluation, the linear contexts perform even worse, while the Enhanced Universal embeddings again outperform the other approaches.

These results reinforce the Levy and Goldberg (2014) findings that dependency-based word embeddings do a better job at distinguishing similarity rather than relatedness because it holds for multiple dependency schemes and levels of enhancement. The Enhanced universal embeddings outperformed the other settings for both datasets. For Chiarello, the margin between the two is statistically significant, whereas for WS353 it is not. This might be due to the fact that the the Chiarello dataset consists of manually-selected pairs that exhibit similarity or relatedness, whereas the settings for WS353 allow for some marginally related or similar terms through (e.g., *size* is related to *prominence*, and *monk* is similar to *oracle*).

4.2 Ranked Similarity

Spearman correlation results for ranked similarity on the SimLex-999 dataset are reported in Table 2b. *Overall* results indicate the performance on the entire collection. In this environment, basic Stanford embeddings outperform all other embeddings explored. This is an interesting result because it shows that the additional dependency labels added for Enhanced embeddings (e.g. for conjunction) do not improve the ranking performance. This trend does not hold for Universal embeddings, with the enhanced versions outperforming the basic embeddings.

All dependency-based word embeddings significantly outperform the baseline methods (10 folds, paired t-test, $p < 0.05$). Furthermore, the unlabeled Universal embeddings performed significantly worse than the simplified Universal, and the simplified, basic, and Enhanced Stanford dependencies, indicating that dependency labels are important for ranking.

Table 2b also includes results for word pairs by part of speech individually. As the majority category, Noun-Noun scores ($n = 666$) mimic the behavior of the overall scores, with basic Stanford embeddings outperforming other approaches. Interestingly, Adjective-Adjective pairs ($n = 111$) performed best with unlabeled Stanford dependencies. Since unlabeled also performs best among universal embeddings, this indicates that dependency labels are not useful for adjective similarity, possibly because adjectives have comparatively few ambiguous functions. Verb-Verb pairs

Embeddings	QC (Acc)	NER (F1)
Universal embeddings		
Unbound	0.921 (+0.007)	0.887 (+0.000)
Simplified	0.929 (+0.016)	0.883 (+0.013)
Basic	0.920 (+0.000)	0.891 (+0.015)
Enhanced	0.923 (+0.008)	0.886 (+0.010)
Enhanced++	0.927 (+0.010)	0.890 (+0.008)
Stanford embeddings		
Unbound	0.926 (+0.015)	0.879 (+0.009)
Simplified	**0.933 (+0.010)**	0.877 (+0.004)
Basic	0.927 (+0.017)	0.885 (+0.020)
Enhanced	0.923 (+0.013)	0.885 (+0.014)
Enhanced++	0.929 (+0.011)	0.884 (+0.006)
Baselines (linear contexts)		
CBOW, k=2	0.921 (+0.008)	0.892 (+0.007)
CBOW, k=5	0.925 (+0.026)	0.892 (+0.001)
Skip-Gram	0.914 (+0.016)	0.887 (+0.006)
SG + Subword	0.919 (+0.022)	**0.896 (+0.009)**

Table 3: Performance results when embeddings are further trained for the particular task. The number in parentheses gives the performance improvement compared to when embeddings are not trainable (Table 2c).

($n = 222$) performed best with Enhanced universal embeddings. This indicates that the augmentation of governors, dependents, and subjects of controlled verbs is particularly useful given the universal dependency scheme, and less so for the English-specific Stanford dependency scheme. Both Stanford and universal unlabeled dependencies performed significantly worse compared to all basic, Enhanced, and Enhanced++ dependencies (5 folds, paired t-test, $p < 0.05$). This indicates that dependency labels are particularly important for verb similarity.

4.3 Downstream Tasks

We present results for question-type classification and named entity recognition in Table 2c. Neither task appears to greatly benefit from embeddings that favor similarity over relatedness or that can rank based on functional similarity effectively without the enhanced sentence feature representations explored by Komninos and Manandhar (2016). We compare the results using to the performance of models with embedding training enabled in Table 3. As expected, this improves the results because the training captures task-specific information in the embeddings. Generally, the worst-performing embeddings gained the most (e.g., CBOW $k = 5$ for QC, and basic Stanford for NER). However, the simplified Stanford embeddings and the embeddings with subword information still outperform the other approaches for QC and NER, respectively. This indicates that the initial state of the embeddings is still important to an extent, and cannot be learned fully for a given task.

5 Conclusion

In this work, we expanded previous work by Levy and Goldberg (2014) by looking into variations of dependency-based word embeddings. We investigated two dependency schemes: Stanford and Universal embeddings. Each scheme was explored at various levels of enhancement, ranging from unlabeled contexts to Enhanced++ dependencies. All variations yielded significant improvements over linear contexts in most circumstances. For certain subtasks (e.g. Verb-Verb similarity), enhanced dependencies improved results more strongly, supporting current trends in the universal dependency community to promote enhanced representations. Given the disparate results across POS tags, future work could also evaluate ways of using a hybrid approach with different contexts for different parts of speech, or using concatenated embeddings.

References

Samira Abnar, Rasyan Ahmed, Max Mijnheer, and Willem Zuidema. 2018. Experiential, distributional and dependency-based word embeddings have complementary roles in decoding brain activity. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*. Salt Lake City, UT, pages 57–66.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Pasca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of HLT-NAACL 2009*. Boulder, CO, pages 19–27.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Comput. Linguist.* 36(4).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606* .

Christine Chiarello, Curt Burgess, Lorie Richards, and Alma Pollock. 1990. Semantic and associative priming in the cerebral hemispheres: Some words do, some words don't sometimes, some places. *Brain and language* 38(1):75–104.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of LREC 2014*. Reykjavik, volume 14, pages 4585–4592.

Marie-Catherine De Marneffe and Christopher D Manning. 2008a. Stanford typed dependencies manual. Technical report, Stanford University.

Marie-Catherine De Marneffe and Christopher D Manning. 2008b. The Stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*. pages 1–8.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. In *Proceedings of EMNLP 2017*. Copenhagen.

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*. pages 406–414.

Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* 41(4).

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A position-aware neural IR model for relevance matching. In *Proceedings of EMNLP 2017*. Copenhagen.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of HLT-NAACL*. San Diego, CA, pages 1490–1500.

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

Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017. Investigating different syntactic context types and context representations for learning word embeddings. In *Proceedings of EMNLP 2017*. Copenhagen, pages 2411–2421.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of COLING 2002*. Taipei, pages 1–7.

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

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of LREC 2016*. Portorož, Slovenia.

Sebastian Pado and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* .

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

Gábor Recski, Eszter Iklódi, Katalin Pajkossy, and Andras Kornai. 2016. Measuring semantic similarity of words using concept networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. Berlin, Germany, pages 193–200.

Sebastian Schuster and Christopher D Manning. 2016. Enhanced English universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of LREC 2016*.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of HLT-NAACL 2003*. volume 4, pages 142–147.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP 2013*. Seattle, WA, pages 1393–1398.