Joint Semantic and Distributional Word Representations with Multi-Graph Embeddings

Pierre-Daix Moreux[*]
Ubisoft Entertainment SA
pierre.daix-moreux@ubisoft.com

Matthias Gallé
Naver Labs Europe
matthias.galle@naverlabs.com

Abstract

Word embeddings continue to be of great use for NLP researchers and practitioners due to their training speed and easiness of use and distribution. Prior work has shown that the representation of those words can be improved by the use of semantic knowledge-bases. In this paper we propose a novel way of combining those knowledge-bases while the lexical information of co-occurrences of words remains. It is conceptually clear, as it consists in mapping both distributional and semantic information into a multi-graph and modifying existing node embeddings techniques to compute word representations. Our experiments show improved results compared to vanilla word embeddings, retrofitting and concatenation techniques using the same information, on a variety of data-sets of word similarities.

1 Motivation

Word embeddings revolutionized NLP through the use of lookup dictionaries that provided continuous representations of words. While surpassed recently by token-based (contextual) embeddings, word embeddings continue to be popular because they are faster to train, can be used plug-and-play by a multitude of machine learning systems, with storing a database of embeddings for sole requirement. This makes them particularly attractive for domain-specific embeddings (e.g., privacy policies (Harkous et al., 2018), oil and gas (Nooralahzadeh et al., 2018) or sentiment analysis (Sarma et al., 2018).

The most popular word embeddings are trained purely with a distributional prior: words occurring in a similar context should have a similar representation. It is well known that the quality of word embeddings can be improved by injecting semantic knowledge in the form of curated databases of relationships between words. However, it is less clear how these two types of knowledge can be mixed. Existing approaches work mostly by fine-tuning them afterwards (Mrkšic et al., 2016; Faruqui et al., 2014) through additional semantic constraints from lexical databases, such as WordNet (Miller, 1995). Although some joint learning approaches have been proposed, the way that the semantic knowledge is injected is not straightforward as the original data-structure is very different (sequences and graphs).

In this paper we propose to represent the co-occurrence relationship of words as a graph. Such a representation opens up natural ways of merging this lexical graph with the semantic graph incorporating new edges with different types. Our experiments show that graph embeddings of the resulting nodes (words) outperform not only pure distributional-based embeddings, but also retrofitted and concatenated ones, on standard word similarity tasks.

The main contributions of this paper are:

- Combining two types of knowledge into one structure represented as a multi-graph.
- Tailoring optimization methods from graph embeddings to include edge types.
- Experimental results showing that they outperform existing methods on a standard word similarity task. The gap is higher when less lexical training data is available.

2 Related Work

Continuous embeddings of words rely on two representations per word w: one considering it as token ($\vec{v}_w$), and another one considering it as context of another token ($\vec{v}'_c$). When using pointwise mutual information of the co-ocurrences matrix (Levy and Goldberg, 2014), this happens implicitly as

[*] Work done while at Naver Labs Europe

Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13), pages 118–123

the final matrix is square. Alternatively, when using a matrix reduction technique (eg: SVD), the second non-diagonal matrix (which is discarded most of the time) can be considered as a contextual view of the tokens.

In more modern word embeddings, like word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014), such a representation is explicit, and the final representation is either one of the two or a combination of them (for a discussion on the impact of different combinations see (Duong et al., 2016)).

2.1 Node embeddings

This dual representation becomes even more important when considering graph embeddings. To find a self-supervised optimization function that induces a representation of nodes, two different goals are formalized (Tang et al., 2015): *homophily*, stating that close nodes should have a similar representation (McPherson et al., 2001), and *structural similarity*, aiming to have similar representations for words that have a similar neighbourhood (Fortunato, 2010). The LINE algorithm (Tang et al., 2015) creates node embeddings optimized for either of those two. When optimizing for homophily, the loss function consists in maximising their first order similarity:

$$\text{sim}(\vec{v}_i, \vec{v}_j) \tag{1}$$

As in previous work, we will define the similarity as the logistic function, work in log-space and define for simplicity :

$$\text{sim}(\vec{v}_i, \vec{v}_j) = \log \frac{1}{1 + e^{-\vec{v}_i \cdot \vec{v}_j}} \tag{2}$$

Optimizing for structural similarity is achieved by focusing on the second order similarity, going through the contextual embedding, by maximising:

$$\text{sim}(\vec{v}_i, \vec{v}'_j) \tag{3}$$

For two nodes with shared neighbourhoods, optimizing this will force their representations to be similar. While LINE uses the alias table method (Li et al., 2014) to optimize either Eq. 1 or Eq. 3, word2vec uses a context window of fixed-size c to maximize Eq. 3.

2.2 Incorporating semantic knowledge

Combining lexical and semantic information from a knowledge-graph – for word embeddings – is not straightforward as they consist in two different representations.

One line of research uses knowledge-graphs to modify word embeddings obtained through pure distributional, lexical approaches *afterwards*. Faruqui et al. (2014) do so by maximising first order similarity (Eq. 1) of two words marked as synonyms in a semantic graph. In order to not disrupt the embedding space, this is regularized with a term insisting that the new embeddings should not be too far apart from the original ones. On top of this Mrkšic et al. (2016) add also antonyms, pushing the representations of two antonyms apart.

Another line of research, closer to our proposition, is to incorporate semantic knowledge at training time. Liu et al. (2015) do so through using ordinal constraints (similarity of synonyms should be higher than of non-synonyms). Many other works trained co-occurrences together with synonyms (Yu and Dredze, 2014; Bian et al., 2014; Kiela et al., 2015) or even other terms. However, those methods treat synonyms equally to context words and do not modify the similarity between them. They are therefore optimized through second order similarity as well.

Our main contribution consists in (i) defining this problem through a conceptually simple multi-graph data structure and (ii) treating different edges types with different similarity (first or second order).

3 Joint Learning of Semantic and Lexical Embeddings

Our proposal is to construct a graph that contains both the lexical information of co-occurrence of words, as well as the semantic information contained in knowledge graphs. We construct a multi-edge graph, where each edge belongs to one of a predefined class. Here, we report results using three classes:

- lexical: we add an edge or increment its weight between node v_i and v_j every time word i occurs in the same window (of predefined size c) than word j
- synonym: words i and j are connected whenever any of their senses belongs to the same synset from WordNet.

- antonym: words i and j are connected whenever any of their senses are antonyms according to WordNet.

In this paper, we will uniformly sample a synonym from the set of synonyms of a word. We define $\mathcal{S}_i$ and $\mathcal{T}_i$ to respectively be the set of all synonyms and antonyms of word i.

For a node (word) v_i, we will model its relation with one of its synonyms using first order proximity:

$$\text{sim}(\vec{v}_s, \vec{v}_i) \tag{4}$$

where $s \in \mathcal{S}_i$.

Using the multi-edge graph setting, we run an experiment where, when possible, we also include an antonym as an additional first order negative example, uniformly sampled from the antonyms set $\mathcal{T}_i$ of a word. For training, we use negative sampling ($N(v)$) and end up with the following objective which we train with stochastic gradient descent:

$$\text{sim}(\vec{v}'_j \cdot \vec{v}_i) + \sum_{n=1}^{k} \mathop{\mathbb{E}}_{v_n \sim N(v)} \left(\text{sim}\left(-\vec{v}'_n \cdot \vec{v}_i\right)\right)$$
$$+ \text{sim}(\vec{v}_s \cdot \vec{v}_i) + \sum_{n=1}^{k} \mathop{\mathbb{E}}_{v_n \sim N(v)} \left(\text{sim}\left(-\vec{v}_n \cdot \vec{v}_i\right)\right)$$
$$+ \text{sim}(-\vec{v}_a \cdot \vec{v}_i) + \sum_{n=1}^{k} \mathop{\mathbb{E}}_{v_n \sim N(v)} \left(\text{sim}\left(-\vec{v}_n \cdot \vec{v}_i\right)\right)$$

where $s \in \mathcal{S}_i$ and $a \in \mathcal{T}_i$.

This objective function accounts for both types of similarity during learning.

4 Results

We trained our embeddings on 25 millions of lines from the english One Billion Word Corpus (Chelba et al., 2013). For any word, we used WordNet to include its set of synonyms and antonyms when needed. As usual, we compute the cosine similarity between the embeddings for each word and compare the Spearman correlation of that similarity with human scores evaluating the extent to which those two words are similar (synonyms) or related.

Datasets compiling scores for explicitly evaluating similarity include SimLex-999 (Hill et al., 2015), RG-65 (Rubenstein and Goodenough, 1965) and MC-30 (Miller and Charles, 1991). For comparison purposes, we include also EN-MTURK-771 (Guy Halawi, 2012), which rather deals with evaluating word pairs' relatedness. WordSimilarity-353 (Finkelstein et al., 2001) (EN-WS-353-SIM) is less clear on whether it evaluates similarity or relatedness, as in contrast to its title, human participants were asked "to estimate the relatedness of the words". Lofi (2015) or Asr et al. (2018) provide good introductions to the difference between evaluating similarity versus relatedness.

Table 1 summarizes the different results obtained with our joint learning approach (with and without antonyms), separate results for first order and second order representations, and word2vec (skip-gram with negative sampling) with and without retrofitting. It also includes results obtained with concatenated representations learned from optimizing first order proximity of synonyms with second order embeddings (as proposed in Tang et al. (2015)).

Concatenation surpasses the retrofitting technique in terms of Spearman correlation scores. It requires however much more training time, as an additional embedding is required for each new semantic relationship.

In any case, the joint learning approach we propose outperforms any kind of method on datasets evaluating similarity. For instance, the Spearman correlations we obtain on SimLex-999, when solely including synonyms, improve on the word2vec baseline by over 11%. Including antonyms increases this difference in performance up to 17% on this particular dataset.

Our attempt at shifting our vector space towards a similarity nudged one seems confirmed by our performance on EN-MTURK-771. Indeed, purely distributional vectors obtain here better Spearman correlation scores.

We also benchmarked the learning curve with an increasing amount of lexical data. In Figure 1, we plot the Spearman correlations obtained when training with an increasing chunk of the 1 Billion Word Corpus, comparing jointly-learned vectors with concatenated, LINE second order and word2vec (vanilla and retrofitted) embeddings.

Figure 1 illustrates that provided with WordNet, the joint learning approach is better equipped to learn representations when less lexical training data is available. Indeed, higher Spearman correlation scores are obtained from the beginning. In addition to this, including antonyms further increases the observed gap.

Method	en-simlex-999	en-rg-65	en-mc-30	en-mturk-771	en-ws-353-sim
w2v	0.402	0.603	0.605	0.629	0.713
w2v retrofitted	0.444	0.667	0.653	0.620	0.689
LINE second order	0.378	0.562	0.604	0.571	0.685
LINE first order	0.304	0.533	0.516	0.536	0.642
Concat. Syn.	0.471	0.643	0.673	0.551	0.703
Joint Learning (Syn.)	0.516	**0.726**	**0.771**	**0.573**	**0.742**
Joint Learning (Syn. + Ant.)	**0.580**	0.705	0.749	0.570	0.675

Table 1: Spearman correlations between human-based judgements and similarity obtained using different embeddings learned on more than 634 millions of tokens. "Concat. Syn." stands for results obtained when concatenating to second order representations, first order embeddings of synonyms. "Syn." and "Syn. + Ant." stand for the inclusion of synonyms and antonyms during joint learning

Word pair - relation	GloVe	w2v	Joint Learning (Syn. + Ant.)	LINE second order
(coffee, cup) - relatedness	0.335	0.333	**0.133**	0.207
(cheap, expensive) - antonymy	0.545	0.512	**-0.504**	0.556
(period, epoch) - meronymy	0.199	0.274	**0.272**	0.348
(torso, trunk) - synonymy	0.257	0.481	**0.766**	0.447

Table 2: Cosine similarity measures for different word pairs.

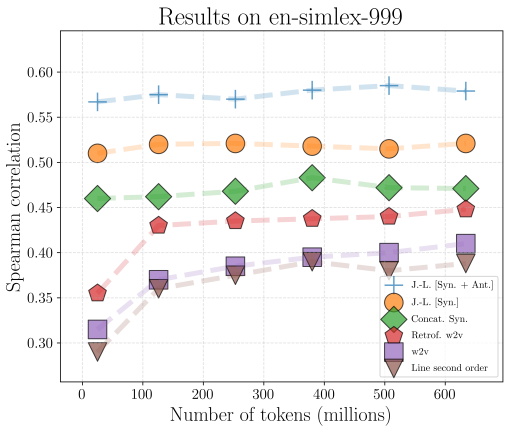

Figure 1: Spearman correlation scores over SimLex-999 obtained for different embeddings on an increasing amount of lexical data. "J.-L." stands for joint-learning.

To illustrate the impact the joint learning approach has on the embeddings space, Table 2 provides examples showing the impact our approach has on the cosine similarity of different kinds of word pairs.

We observe that while the cosine similarity of the related word pair decreases, the similarity of two synonyms greatly increases in comparison to the antonym pair's similarity, which turns negative. Interestingly, the similarity of the two provided meronyms does not show any great difference with respect to the one provided by distributional methods.

5 Conclusion

We proposed a novel way of combining the lexical information of co-occurrences with that of semantic knowledge bases. Our method maps all those sources of information into a multi-graph and modifies existing node embeddings technique so that they treat edges of different types differently. We claim that our proposal is conceptually simpler than existing proposals which either use the information from the semantic graph to finetune the word embeddings obtained through lexical information, or combine the information in some other indirect way. Instead of this, our method formalizes an objective to include novel edge types. In our experiments we presented results using three types of edges. In addition to obtaining better results when measured on a standard word similarity task, our method is less data-greedy: it obtains better results with much less training data than other methods. This can be interesting in particular for the creation of in-domain word embeddings, where curated knowledge graph exists. Thus, using a multi-graph allows for an easy way of incorporating additional types of information. In particular, we are consid-

ering multi-lingual embeddings through the inclusion of bilingual dictionaries which connect nodes (words) of different languages.

References

Fatemeh Torabi Asr, Robert Zinkov, and Michael Jones. 2018. Querying word embeddings for similarity and relatedness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 675–684.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. 2016. Learning crosslingual word embeddings without bilingual corpora. *arXiv preprint arXiv:1606.09403*.

Manaal Faruqui, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2014. Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

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. ACM.

Santo Fortunato. 2010. Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Evgeniy Gabrilovich Yehuda Koren Guy Halawi, Gideon Dror. 2012. Large-scale learning of word relatedness with constraints. *KDD*, pages 1406–1414.

Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 531–548.

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

Douwe Kiela, Felix Hill, and Stephen Clark. 2015. Specializing word embeddings for similarity or relatedness. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2044–2048.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900. ACM.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning semantic word embeddings based on ordinal knowledge constraints. 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 1501–1511.

Christoph Lofi. 2015. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. *Information and Media Technologies*, 10(3):493–501.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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

Nikola Mrkšic, Diarmuid OSéaghdha, Blaise Thomson, Milica Gašic, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of NAACL-HLT*, pages 142–148.

Farhad Nooralahzadeh, Lilja Øvrelid, and Jan Tore Lønning. 2018. Evaluation of domain-specific word embeddings using knowledge resources. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Prathusha K Sarma, Yingyu Liang, and William A Sethares. 2018. Domain adapted word embeddings for improved sentiment classification. In *ACL*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550.