Random Walks and Neural Network Language Models on Knowledge Bases

Josu Goikoetxea, Aitor Soroa and Eneko Agirre
IXA NLP Group
University of the Basque Country
Donostia, Basque Country
{josu.goikoetxea,a.soroa,e.agirre}@ehu.eus

Abstract

Random walks over large knowledge bases like WordNet have been successfully used in word similarity, relatedness and disambiguation tasks. Unfortunately, those algorithms are relatively slow for large repositories, with significant memory footprints. In this paper we present a novel algorithm which encodes the structure of a knowledge base in a continuous vector space, combining random walks and neural net language models in order to produce novel word representations. Evaluation in word relatedness and similarity datasets yields equal or better results than those of a random walk algorithm, using a dense representation (300 dimensions instead of 117K). Furthermore, the word representations are complementary to those of the random walk algorithm and to corpus-based continuous representations, improving the state-of-the-art in the similarity dataset. Our technique opens up exciting opportunities to combine distributional and knowledge-based word representations.

1 Introduction

Graph-based techniques over Knowledge Bases (KB) like WordNet (Fellbaum, 1998) have been widely used in NLP tasks, including word sense disambiguation (Agirre et al., 2014; Moro et al., 2014), semantic similarity and semantic relatedness between terms (Agirre et al., 2009; Agirre et al., 2010; Pilehvar et al., 2013). For instance, Agirre et al. (2009; 2010) apply a random walk algorithm based on Personalized PageRank to WordNet, presenting the best results to date among WordNet-based methods for the well-known WS353 word-similarity dataset (Finkelstein et al., 2001). For each target word, the method performs a personalized random walk on the WordNet graph. At convergence, the target word is represented as a vector in a multi-dimensional conceptual space, with one dimension for each concept in the KB. The good results of the algorithm contrast with the large dimensionality of the vectors that it needs to produce, 117K dimensions (one per synset) for WordNet.

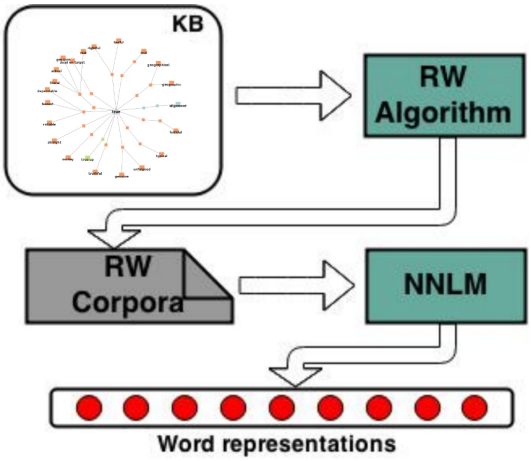

Figure 1: Main architecture for generating KB word embeddings. A random walk algorithm over the KB produces a synthetic corpus, which is fed into a NNLM to produce continuous word representations.

In recent years a wide variety of Neural Network Language Models (NNLM) have been successfully employed in several tasks, including word similarity (Collobert and Weston, 2008; Socher et al., 2011;

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1434–1439,
Denver, Colorado, May 31 – June 5, 2015.

Turian et al., 2010). NNLM extract meaning from unlabeled corpora following the distributional hypothesis (Harris, 1954), where semantic features of a word are related to its co-occurrence patterns. NNLM learn word representations in the form of dense scalar vectors in n-dimensional spaces (e.g. 300 dimensions), in which each dimension is a latent semantic feature. The representations are obtained by optimizing the likelihood of existing unlabeled text. More recently, Mikolov et al. have developed simpler NNLM architectures (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c), which drastically reduced computational complexity by deleting the hidden layer, enabling to compute accurate word representations from very large corpora. The representations obtained by these methods are compact, taking 1.5G for 3M words on 300-dimensional space, and have been shown to outperform other distributional corpus-based methods on several tasks, including the WS353 word similarity dataset (Baroni et al., 2014).

In this work we propose to encode the meaning of words using the structural information in knowledge bases. That is, instead of modeling the meaning based on the co-occurrences of words in corpora, we model the meaning based on random walks over the knowledge base. Each random walk is seen as a context for words in the vocabulary, and fed into the NNLM architecture, which optimizes the likelihood of those contexts (cf. Fig. 1). The resulting word representations are more compact than those produced by regular random walk algorithms (300 vs. tens of thousands), and produce very good results on two well-known benchmarks on word relatedness and similarity: WS353 (Finkelstein et al., 2001) and SL999 (Hill et al., 2014b), respectively. We also show that the obtained representations are complementary to those of random walks alone and to distributional representations obtained by the same NNLM algorithm, improving the results.

Some recent work has explored embedding KBs in low-dimensional continous vector spaces, representing each entity in a k-dimensional vector and characterizing typed relations between entities in the KB (e.g. born-in-city in Freebase or part-of in WordNet) as operations in the k-dimensional space (Wang et al., 2014). The model estimates the parameters which maximize the likelihood of the triples, which can then be used to infer new typed relations which are missing in the KB. In contrast, we use the relations to explicitly model the context of words, in two complementary approaches to embed information in KBs into continuous spaces.

2 NNLM

Neural Network Language Models have become a useful tool in NLP on the last years, specially in semantics. We have used the two models proposed in (Mikolov et al., 2013c) due to their simplicity and effectiveness in word similarity and relatedness tasks (Baroni et al., 2014): Continuous Bag of Words (CBOW) and Skip-gram. The first one is quite similar to the feedforward Neural Network Language Model, but instead of a hidden layer it has a projection layer, and thus all the words are projected in the same position. Word order has thus no influence in the projection. The training criterion is as follows: knowing previous and subsequent words in context, the model maximizes the probability of the predicting the word in the middle. The Skip-gram model uses each current word as an input to a log-linear classifier with a continuous projection layer, and predicts the previous and subsequent words in a context window.

Although the Skip-gram model seems to be more accurate in most of the semantic tasks, we have used both variants in our experiments. We used a publicly available implementation[1].

3 Random Walks and NNLM

Our method performs random walks over KB graphs to create synthetic contexts which are fed into the NNLM architecture, creating novel word representations. The algorithm used for creating the contexts is a Monte Carlo method for computing the PageRank algorithm (Avrachenkov et al., 2007).

We consider a KB as undirected graph $G = (V, E)$, where V is the set of concepts and E represents links among concepts. We also need a dictionary, an association from words to KB concepts. We construct an inverse dictionary that maps graph vertices with the words than can be linked to it.

The inputs of the algorithm are: 1) the graph $G = (V, E)$, 2) the inverse dictionary and 3) the damp-

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

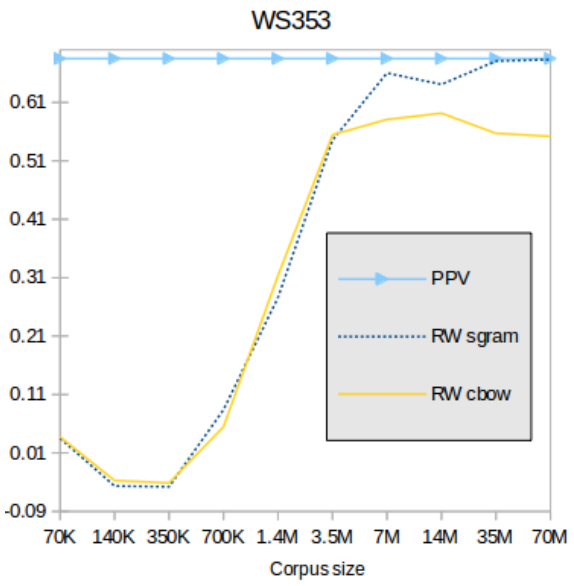

Figure 2: Spearman results on relatedness (WS353) for different corpus sizes (in sentences).

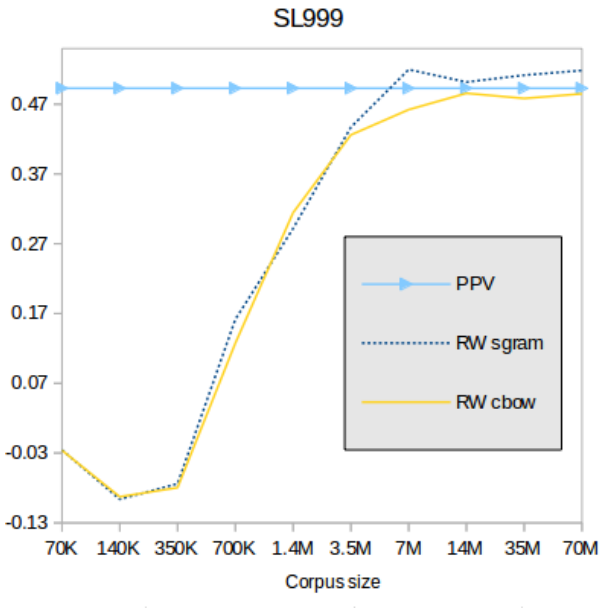

Figure 3: Spearman results on similarity (SL999) for different corpus sizes (in sentences).

ing factor α[2]. In our experiments we used WordNet 3.0 with gloss relations[3], which has 117.522 nodes (synsets) and 525.356 edges (semantic relations). Regarding the dictionary, WordNet already contains links from words to concepts. The dictionary includes the probability of a concept being lexicalized by a specific word, as estimated by the WordNet team from their hand-annotated corpora. Both dictionary and graph are freely available[4].

The method first chooses a vertex at random from the vertex set V, and performs a random walk starting from it. At each step, the random walk might terminate with probability $(1-\alpha)$ or choose a neighbor vertex at random with probability α. Each time the random walk reaches a vertex, a word is emitted at random using the probabilities in the inverse dictionary. When the random walk terminates, the sequence of emitted words forms the pseudo sentence which is fed to the NNLM architecture, and the process starts again choosing a vertex at random until a maximum number of pseudo sentences have been generated.

Our method creates pseudo sentences like the following:

(1) *amphora wine nebuchadnezzar bear retain long*

(2) *graphology writer write scribble scrawler heedlessly in_haste jot note notebook*

These examples give us clues of the kind of the implicit semantic information that is encoded in the generated pseudo-corpus. Example 1 starts with *amphora* following with *wine* (with which amphoras are usually filled with), *nebuchadnezzar* (a particular bottle size) and finishing with words that are related to wine storage, like *bear*,*retain* and *long*. Example 2 shows a similar phenomenom; it starts with *graphology*, follows with the closely related *writer*, then *writer*, finishing with names and adjectives of different variants of writing, such as *scribble*, *scrawler*, *heedlessly*, *in_haste* and *jot*; finally, the context ends with *note* and *notebook*. Note that our method also produces multiword terms like *in_haste*.

4 Experiments

We have trained two Neural Network models, CBOW and Skip-gram, with several iterations of random walks over WordNet. We trained both models with default parameters (Mikolov et al., 2013a): vector size 300, 3 iterations, 5 negative samples, and

[2]The damping factor is the only parameter of PageRank.
[3]http://wordnet.princeton.edu/glosstag.shtml
[4]http://ixa2.si.ehu.eus/ukb

	SL999	WS353
Skip-gram	0.442	**0.686**
RWSGRAM	**0.520**	0.683
RWCBOW	0.486	0.591
PPV	0.493	0.683

Table 1: Spearman correlation results for our methods (RWSGRAM, RWCBOW) on WordNet random walks, compared to just random walks (PPV), and Skip-gram on text corpora.

	SL999	WS353
(a) RWSGRAM	0.518	0.683
(b) PPV	0.493	0.683
(c) Skip-gram	0.442	0.686
(a+b)	0.535	0.700
(a+c)	0.533	0.748
(a+b+c)	**0.552**	0.759
Best	0.520	**0.800**

Table 2: Combinations and best published results: SL999 (Hill et al., 2014a), WS353 (Radinsky et al., 2011).

window size 5. In order to check how many iterations of the random walk algorithm are needed to learn good word representations, we produced up to $70 \cdot 10^6$ contexts. The the damping factor (α) of the random walk algorithm was set to 0.85, a usual value (Agirre et al., 2010). All parameters were thus set to default, and we only explored different corpus sizes.

The word representations were evaluated on WS353 (Finkelstein et al., 2001) and SL999 (Hill et al., 2014b), two datasets on word relatedness and word similarity, respectively. In order to compute the similarity of two words, it suffices to calculate the cosine between the respective word representations. The evaluation measure computes the rank correlation (Spearman) between the human judgments and the system values.

In order to contrast our results with the two related techniques, we used UKB[5], a publicly available implementation of Personalized PageRank (Agirre et al., 2014), and ran it over the same graph as our proposed methods. We used it out-of-the-box with a damping value of 0.85. We also downloaded the embeddings learnt by (Mikolov et al., 2013a) using Skip-gram over a large text corpus [6]. We used the same cosine algorithm to compute similarity with all word representations. To distinguish one word representation from the other, we will call our models RWCBOW and RWSGRAM respectively (RW for random-walk), in contrast to the original Personalized PageRank algorithm (PPV) and the corpus-based embeddings learned using Skip-grams (Skip-gram).

Figures 2 and 3 show the learning curves on the WS353 and SL999 datasets relative to the number of contexts produced by the random walks on WordNet. The results show that WordNet representations grow quickly (around 7 million contexts), converging around 70M, obtaining practically the same results as PPV for WS353, and better results for SL999[7].

The results at convergence are shown in Table 1, together with those of PPV and Skip-gram. Regarding SL999, we can see that the best results are obtained with RWSGRAM, improving over PPV and Skip-gram. Regarding WS353, all methods except RWSGRAM obtain similar results. The results show that our methods are able to effectively capture the information in WordNet, performing on par to the original PPV algorithm, and better than the corpus-based Skip-gram on the SL999 dataset. Note that the best published results for WS353 using WordNet are those of (Agirre et al., 2010) using PPV, which report 0.685.

In order to see if the word representations that we learn are complementary to those of PPV and Skip-gram, we combined the scores produced by each word representation. Given the potentially different scales of the similarity values, we assigned to each item the average of the ranks of the pair in each output. The top part of Table 2 repeats the three relevant systems. The (a+b) row reports an improvement in both datasets, showing that RWSGRAM on WordNet is complementary to PPV in WordNet, and is thus a different representation, even if both use the same knowledge base. The (a+b) and (a+b+c) show that corpus-based Skip-grams are also complemen-

[5] http://ixa2.si.ehu.eus

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

[7] We tried larger context sizes, up to 700M confirming that convergence was around 70M.

tary, yielding incremental improvements. In fact, the combination of all three improves over the best published results on SL999, and approaches the best results for WS353, as shown in the last row of the Table. The state of the art on SL999 corresponds to (Hill et al., 2014a), who training a Recurrent Neural Net model on bilingual text. The best results on WS353 correspond to (Radinsky et al., 2011), who combine a Wikipedia-based algorithm with a corpus-based method which uses date-related information from news to learn word representations.

Note that we have only performed some simple combination to show the complementarity of each information source. More sophisticated combinations (e.g. learning a regression model) could further improve results.

We have performed some qualitative analysis, which indicates that there is a slight tendency for corpus embeddings (with the window size used in the experiments) to group related words (e.g. physics - proton), and not so much similar words (e.g. vodka - gin), while our KB embeddings include both. This analysis agrees with the results in Table 1, where all KB results are better than corpus-based Skip-gram for the semantic similarity dataset (SL999). In passing, note that the best published results to date on similarity (Hill et al., 2014a) use embeddings learnt from bilingual text which suggests that bilingual corpora are better suited to learn embeddings capturing semantic similarity.

5 Conclusions

We have presented a novel algorithm which encodes the structure of a knowledge base in a continuous vector space, combining random walks and neural net language models to produce new word representations. Our evaluation in word relatedness and similarity datasets has shown that these new word representations attain similar results to those of the original random walk algorithm, using 300 dimensions instead of tens of thousands. Furthermore, the word representations are complementary to those of the random walk algorithm and to corpus-based continuous representations, producing better results when combined, and improving the state-of-the-art in the similarity dataset. Hand inspection reinforces the observation that WordNet-based

A promising direction of this research is to leverage multilingual Wordnets to produce cross-lingual embeddings.

On another direction, one of the main limitations of KB approaches is that they produce a relatively small number of embeddings, limited by the size of the dictionary. In the future we want to overcome this sparsity problem by combining both textual and KB based embeddings into a unified model. In fact, we think that our technique opens up exciting opportunities to combine distributional and knowledge-based word representations.

It would also be interesting to investigate the influence of the different semantic relations in WordNet, either by removing certain relations or by assigning different weights to them. This investigation could give us deeper insights about the way our knowledge-based approach codes meaning in vector spaces.

Acknowledgements

This work was partially funded by MINECO (CHIST-ERA READERS project – PCIN-2013-002-C02-01, and SKaTeR project – TIN2012-38584-C06-02), and the European Commission (QTLEAP – FP7-ICT-2013.4.1-610516). The IXA group is funded by the Basque Government (A type Research Group).

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 *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Eneko Agirre, Montse Cuadros, German Rigau, and Aitor Soroa. 2010. Exploring Knowledge Bases for Similarity. In *LREC*.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. 2007. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM J. Numer. Anal.*, 45(2):890–904.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Dont 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.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

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.

Zellig S Harris. 1954. Distributional structure. *Word*.

Felix Hill, KyungHyun Cho, Sébastien Jean, Coline Devin, and Yoshua Bengio. 2014a. Not all neural embeddings are born equal. *CoRR*, abs/1410.0718.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014b. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*.

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 *Advances in Neural Information Processing Systems*, pages 3111–3119.

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

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unied approach. *Transactions of the Association of Computational Linguistics*, 2:231–244, May.

Mohammad Taher Pilehvar, David Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: a Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351, Sofia, Bulgaria.

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*, WWW '11, pages 337–346, New York, NY, USA. ACM.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1591–1601, Doha, Qatar, October. Association for Computational Linguistics.