Embedding Senses for Efficient Graph-based Word Sense Disambiguation

Luis Nieto Piña and **Richard Johansson**
University of Gothenburg, Sweden
{luis.nieto.pina, richard.johansson}@gu.se

Abstract

We propose a simple graph-based method for word sense disambiguation (WSD) where sense and context embeddings are constructed by applying the Skip-gram method to random walks over the sense graph. We used this method to build a WSD system for Swedish using the SALDO lexicon, and evaluated it on six different annotated test sets. In all cases, our system was several orders of magnitude faster than a state-of-the-art PageRank-based system, while outperforming a random baseline soundly.

1 Introduction

Word sense disambiguation (WSD) is a difficult task for automatic systems (Navigli, 2009). The most accurate WSD systems build on supervised learning models trained on annotated corpora (Taghipour and Ng, 2015), but because of the difficulty of the sense annotation task (Artstein and Poesio, 2008), the luxury of supervised training is available for a few languages only.

An approach that circumvents the lack of annotated corpora is to take advantage of the information available in lexical knowledge bases (LKBs) like WordNet (Miller, 1995; Fellbaum, 1998). This kind of resource encodes word sense lexicons as graphs connecting lexically and semantically related concepts. Several methods are available that use LKBs for WSD (Navigli and Lapata, 2007; Agirre and Soroa, 2009). These approaches usually apply a relatively complex analysis of the underlying graph based on the context of a target word to disambiguate it; e.g., Agirre and Soroa (2009) use the Personalized PageRank algorithm to perform walks on the graph. However, these methods are computationally very costly, which makes them practically useless for large corpora.

In this paper, we investigate a more time-efficient approach to graph-based WSD. We represent the concepts in the LKB by training vector space models on synthetic datasets created using random walks on the LKB's graph. These synthetic datasets are built on the assumption that a random walk starting at a given node in the graph will be composed of interrelated concepts, effectively building a context for it. Training a vector space model on a collection of such data generated for each node in an LKB's graph would result in related concepts being represented near each other in the vector space, according to the distributional hypothesis (Harris, 1954). We then use these representations to perform context-based disambiguation taking advantage of the geometric notions of similarity typical of vector space models. Using simple mechanisms for disambiguation and random walks allows our method to be orders of magnitude faster while keeping its accuracy well above the random-sense baseline.

2 Model

2.1 Word sense vector space model

The Skip-gram model (Mikolov et al., 2013) is a neural network language model (Bengio et al., 2003) intended to produce high-quality word vector representations trained on large collections of text. In its original formulation these representations are limited to a vocabulary of word-forms extracted from the corpus used to train the model. The representations are dense vectors in a high-dimensional space in which it is expected that words with a similar meaning are represented near each other, which allows to associate a similarity measure with a geometrical distance measure. These representations are trained to, given a word, predict its context; the training algorithm, thus, works with two separate vector spaces in which context and target words are represented.

Proceedings of the 2016 Workshop on Graph-based Methods for Natural Language Processing, NAACL-HLT 2016, pages 1–5,
San Diego, California, June 17, 2016.

Skip-gram introduced a highly efficient approach to language modeling using a shallow neural architecture, which has also been extended to handle word *sense* representation (Neelakantan et al., 2014; Chen et al., 2014; Johansson and Nieto Piña, 2015b; Nieto Piña and Johansson, 2015). Our aim in this paper is to build *graph-based* word sense embeddings and apply them to the task of WSD as follows: Given a sentence with an ambiguous word, we can then compare the representation of its context words with each of the ambiguous word's sense representations to decide which of them fits the context better.

For this purpose we use a modified version of the original Skip-gram implementation by Levy and Goldberg (2014), `word2vecf`, which specifies separate target and context vocabularies, making it possible to represent word senses as targets while keeping the context vocabulary restricted to word forms.

2.2 Random walks as contexts

Given a node in a graph G, a random walk generates a random sequence of interconnected nodes by selecting randomly from the edges of the current node at each step. The length of the random walk is controlled by a stop probability p_s. I.e., at each node visited in the walk, the probability of stopping is p_s; if the walk does not stop, one of the node's edges is followed to include another node in the walk. We repeat this process a number of times N_{walk} for each node in G to obtain $|G| \times N_{\text{walk}}$ random walks, where $|G|$ is the number of nodes in G.

The nodes in G are expected to represent word senses, while its edges connect semantically related word senses. Thus, a sequence of nodes generated by a random walk is a set of related word senses. Our assumption is that such a sequence can be considered a context of its starting node: a set of words that are related to, and can appear together in real texts with, the word sense represented by that node, thus emulating real text sentences; to what extent this assumption holds depends of course on the structure of the LKB we are using. Previous efforts in building word embeddings have shown the plausibility of this approach (Goikoetxea et al., 2015).

It can also be argued that different senses of a word appear in different contexts (e.g., it is plausible that the *music* sense of *rock* appears together with *play* and *concert*, while not so much with *mineral* or *throw*). By generating contexts semantically related to a given sense of a word, we expect the resulting vectors trained on them to be effective in the task of word sense disambiguation. At the same time, as the same number of contexts (random walks) are generated for each word sense (node in G), no word sense in the vocabulary contained in G is underrepresented, as can be the case in real text corpora.

In order to conform to the definition of context vocabulary given above, given that nodes in G represent senses, those senses that form part of a context in a random walk will have to be mapped to their word-forms using a dictionary.

2.3 WSD mechanism

Given an ambiguous target word w_i in context $c_{i,j}$, $j = 1, \ldots, n$, our disambiguation mechanism assigns a score to each of its senses $s_{i,k}$, $k = 1, \ldots, K$, based on the dot product of the sense vector $v(s_{i,k})$ with the sum of the context vectors $v(c_{i,j})$:

$$v(s_{i,k})^\intercal \cdot \sum_{j=1}^{n} v(c_{i,j}) \tag{1}$$

Note that all the information used to disambiguate originates from the LKB in the form of co-occurrence of concepts in RWs on the graph; no *external* information, like *a priori* sense probabilities, are used. The scores in Equation 1 are derived from the probability of the context words given a sense, calculated using the softmax function:

$$p(c_{i,1}, \ldots, c_{i,n} | s_{i,k}) = \frac{e^{v(s_{i,k})^\intercal \cdot \sum_{j=1}^{n} v(c_{i,j})}}{\sum_{k'=1}^{K} e^{v(s_{i,k'})^\intercal \cdot \sum_{j=1}^{n} v(c_{i,j})}}.$$

This expression is based on Skip-gram's objective function used to maximize the probability of a context given a target word. In our method, then, each ambiguous word is disambiguated by maximizing its sense scores (Eq. 1) and selecting the highest scoring sense for that instance.

3 Experiments

We built a WSD system for Swedish by applying the random walk-based training described above to the SALDO lexicon (Borin et al., 2013). In the experiments, we then evaluated this system on six different annotated corpora, in which the ambiguous words have been manually disambiguated according

to SALDO, and compared it to random and first-sense baselines and UKB (Agirre and Soroa, 2009), a state-of-the-art graph-based WSD system.

3.1 The SALDO Lexicon

SALDO is the largest freely available lexical resource of this kind available for Swedish: the version used in this paper contains roughly 125,000 entries organized into a single semantic network. Similarly to WordNet (Fellbaum, 1998), SALDO is a large, manually constructed, and general-purpose lexicon that defines the senses in terms of a semantic network. But there are also important differences between WordNet and SALDO, first of all that the sense distinctions in SALDO tend to be more coarse-grained than in WordNet.

The SALDO network is defined in terms of semantic *descriptors*. A descriptor of a sense is another sense used to define its meaning. The most important descriptor is called the *primary* descriptor (PD), and since every sense in SALDO (except an abstract root sense) has a unique PD, the PD subgraph of SALDO forms a tree. A sense can be related to its primary descriptor through hyponymy, synonymy, meronym, antonymy, or some other relationship such as a predicate–argument relationship; this is another contrast with WordNet, where it is the hyponymy subgraph that forms the backbone. In practice, most PDs in SALDO are either synonyms or hypernyms.

To exemplify, Figure 3.1 shows a fragment of the PD tree. In the example, there are some cases where the senses are connected through hyponymy, such as *hard rock* being a type of *rock music*, but there are also other types of relations, such as *to play* being defined in terms of *music*.

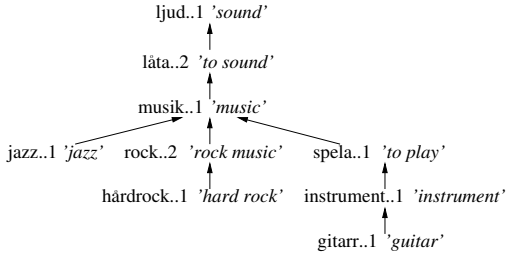

Figure 1: A part of the primary descriptor tree in SALDO.

In this work, we use the PD tree to generate the random walks. For instance, a random walk starting at *rock music* might consist of the senses corresponding to *music*, *play*, *instrument*, *guitar*. As mentioned above, these senses are then mapped back to their corresponding lemmas before being used as context features in `word2vecf`.

3.2 Evaluation Corpora

For development and evaluation, we used six different collections of sense-annotated examples. The first two, the *SALDO examples* (SALDO-ex) and *Swedish FrameNet examples* (SweFN-ex) consist of sentences selected by lexicographers to exemplify the senses (Johansson and Nieto Piña, 2015a). The former is dominated by the most frequent verbs, while the latter has a more even distribution. In our experiments, these two collections were used as a development set to tune the system's parameters.

The additional four collections are taken from an ongoing annotation project (Johansson et al., 2016); each collection corresponds to a domain: blogs, novels, Wikipedia, and Europarl (Koehn, 2005). Unlike the two collections mentioned above, in which the instances have been selected by lexicographers to be prototypical and to have a good coverage of the sense variation, the instances in these four collections are sampled uniformly from running text.

Corpus	Size	$\bar{n}_s$
SALDO-ex	1055	3.1
SweFN-ex	1309	2.9
Blogs	1014	2.9
Europarl	1282	2.7
Novels	1204	3.0
Wikipedia	1311	2.7

Table 1: Evaluation corpus statistics.

We preprocessed the examples in the six collections to tokenize, compound-split, and lemmatize the texts, and to determine the set of possible senses in a given context. We used content words only: nouns, verbs, adjectives, and adverbs. All unambiguous instances were removed from the sets, and we also excluded sentences where the target was a multiword expression or a part of a compound word. We also removed a few instances that could not be lemmatized unambigously.[1] Table 1 shows the number of instances in each collection, as well as the average number of senses per instance ($\bar{n}_s$).

[1]Note that this is done only to facilitate comparison to the UKB model; it is not necessary for our system.

3.3 Evaluation

A model is trained on synthetic datasets compiled from random walks on SALDO. These walks are parameterized by their stop probability p_{stop}, which effectively controls the length of the random walk and has two effects: it impacts the size of training data (a lower p_{stop} will generate longer walks on average, and vice versa); and it controls the level of relatedness between the target sense and the words included in the context—a longer walk will wander away from the initial sense, including increasingly unrelated concepts, while a shorter one will keep its concepts closely related.

We tuned the model by training several versions with different p_{stop} and evaluated their performance on the development datasets. As the best-performing parameterization, we chose $p_{\text{stop}} = 0.25$, which generates random walks with an average length of 3.75 nodes and achieves an accuracy of 51.6% on the development datasets. In all cases, the vector space's dimensionality for senses and contexts is 200, and 10 iterations of the training algorithm are used.

Using this parameterization, we trained models on two different RW datasets: on one, random walks were performed on an unweighted version of SALDO (i.e., all edges are equally probable from any given node); on the other, the graph was weighted favoring the selection of a node's unique PD, with probability 0.5, over inverse (incoming) PD connections, which were uniformly distributed over the remaining probability mass.

The disambiguation mechanism explained in Section 2 is applied to sentences containing one ambiguous word using the sense and context representations that result from training the models: A score is calculated for each of the senses of an ambiguous target word in a context window of size 10 (to each side of the target word) and the highest scoring sense is selected to disambiguate the entry. The accuracy of the method is then obtained by comparing these selections with the annotations of the test datasets.

The results of evaluating this models on each component of the test dataset are shown in Table 2. The performance of the UKB model[2] by Agirre and Soroa (2009) on our datasets is also shown in this table, along with first-sense (S1) and random-sense baselines (Rand). These figures show that the first-sense approach is still a hard baseline. Amongst our two models (RW), the one trained on a weighted graph (w) performs consistently better; both of them outperform by a wide margin the random-sense baseline. The accuracy on the development sets is generally lower, especially in the case of the first-sense baseline, underlying their difference in nature with respect to the test sets (see Section 3.2).

Corpus	RW (uw)	RW (w)	UKB	S1	Rand
SALDO-ex	52.1	51.6	55.5	53.2	39.3
SweFN-ex	51.0	49.5	53.7	54.3	40.3
Blogs	49.8	58.0	70.0	72.4	40.8
Europarl	55.7	59.4	67.6	67.9	42.3
Novels	56.6	59.9	70.1	77.2	40.1
Wikipedia	60.4	59.6	69.5	76.8	41.2

Table 2: WSD accuracies on the development and test sets.

Regarding execution times, the tested models take a few hours to train and, once trained, are able to disambiguate over 8 000 instances per second, significantly surpassing the UKB model's times, which disambiguates approximately 8 instances per second. This is related to the fact that the complexity of our disambiguation mechanism is linear on the context vectors (see Equation 1), while the UKB model's is dependent on the graph size.

4 Conclusion

In this paper we have presented a WSD method trained on a synthetic corpus composed of random walks over an LKB's graph. This method has been shown to be very efficient, disambiguating thousands of words per second. While the accuracy obtained by the method does not beat that of comparable approaches, it is several orders of magnitude faster while outperforming a random-sense baseline. As has been shown in the results, the way in which random walks are generated seems to have an influence on the results; exploring alternative ways of generating training datasets might be a way of improving the model's results while retaining its efficiency.

Acknowledgments

This research was funded by the Swedish Research Council under grant 2013–4944.

[2]We used version 2.0 of UKB, run in the *word-by-word* mode, using an unweighted graph based on the PD tree.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing PageRank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

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

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation*, 47:1191–1211.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035.

Christiane Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado, May–June. Association for Computational Linguistics.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Richard Johansson and Luis Nieto Piña. 2015a. Combining relational and distributional knowledge for word sense disambiguation. In *Proceedings of the 20th Nordic Conference of Computational Linguistics*, pages 69–78, Vilnius, Lithuania.

Richard Johansson and Luis Nieto Piña. 2015b. Embedding a semantic network in a word space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1428–1433, Denver, Colorado, May–June. Association for Computational Linguistics.

Richard Johansson, Yvonne Adesam, Gerlof Bouma, and Karin Hedberg. 2016. A multi-domain corpus of Swedish word sense annotation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.

Phillip Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, pages 79–86.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

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.

Roberto Navigli and Mirella Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *IJCAI*, pages 1683–1688.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69, February.

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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar, October. Association for Computational Linguistics.

Luis Nieto Piña and Richard Johansson. 2015. A simple and efficient method to generate word sense representations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 465–472, Hissar, Bulgaria, September.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 338–344, Beijing, China, July. Association for Computational Linguistics.