

# Towards Lexical Chains for Knowledge-Graph-based Word Embeddings

Kiril Simov

Svetla Boytcheva

Petya Osenova

Linguistic Modeling and Knowledge Processing Department

Institute of Information and Communication Technology

Bulgarian Academy of Sciences

kivs@bultreebank.org, svetla.boytcheva@gmail.com,

petya@bultreebank.org

## Abstract

Word vectors with varying dimensionalities and produced by different algorithms have been extensively used in NLP. The corpora that the algorithms are trained on can contain either natural language text (e.g. Wikipedia or newswire articles) or artificially-generated pseudo corpora due to natural data sparseness.

We exploit Lexical Chain based templates over Knowledge Graph for generating pseudo-corpora with controlled linguistic value. These corpora are then used for learning word embeddings. A number of experiments have been conducted over the following test sets: WordSim353 Similarity, WordSim353 Relatedness and SimLex-999.

The results show that, on the one hand, the incorporation of many-relation lexical chains improves results, but on the other hand, unrestricted-length chains remain difficult to handle with respect to their huge quantity.

## 1 Introduction

Recent research in NLP has focused on distributional semantic models that incorporate linguistic information from various resources. Such models are trained with different algorithms, among which: Network Language Model (NNLM), Latent Semantic Analysis (LSA), etc. Word embeddings have become lately a popular distributed representation. Word vectors with varying dimensionalities and produced by different algorithms have been extensively put forward in the literature. The corpora that the algorithms are trained on can contain either natural language text (e.g.

Wikipedia or newswire articles) or artificially-generated pseudo corpora, such as the output of the Random Walk on Graphs algorithm, when run to select sequences of nodes from a knowledge graph (KG) — see (Goikoetxea et al., 2015) and (Ristoski and Paulheim, 2016). We denote the pseudo corpus generated via Random Walk on Graphs algorithm as *Pseudo Corpus RWG*.

In this paper we present the results from our initial experiments of training word embeddings on the basis of generated pseudo corpora over a knowledge graph via lexical chain patterns. We name the pseudo corpus generated via lexical chain patterns *Pseudo Corpus LC*. The main knowledge graph in the experiments is the English WordNet (WN) (Fellbaum, 1998). It is represented as nodes corresponding to synsets in WN and arcs corresponding to relations encoded in WN, such as hypernymy, meronymy, entailment, etc. We have extended the graph with new relations between synsets (i.e. additional arcs in the graph). These new relations come from sources outside WordNet, such as relations extracted from semantically annotated corpora or explication of implicit knowledge in WordNet on the basis of inference procedures (e.g. transitive closure over the relations). In our previous work we showed that enriching the knowledge graph with more relations can lead to accuracy improvement in the Word Sense Disambiguation (WSD) task — (Simov et al., 2016b), (Simov et al., 2016a).

However, the relevant evaluation of such distributed semantic models still faces some issues. Here a few of the typical problems in this line of work are listed: a) the evaluation datasets merge similarity and association relations, while the models typically favor one subset and suppress the role of the other; b) uneven frequency of the words in the corpus; c) the impossibility to cover all words with all their meanings in all

possible contexts. In this paper we aim to evaluate the obtained word embeddings for the similarity and association (relatedness) tasks. We report our initial experiments on training word embeddings over pseudo corpora generated with the help of lexical chain templates as measured on the following datasets: WordSim353 Similarity, WordSim353 Relatedness (see (Finkelstein et al., 2001) for introducing WordSim353) and SimLex-999 (Hill et al., 2015). We compare these results to the results over a pseudo corpora generated over the same knowledge graph with the help of Random Walk over Graphs as presented within (Goikoetxea et al., 2015). Our currently generated *Pseudo Corpora LC* do not lead to better word embeddings (with respect to the above mentioned evaluation) in comparison to the *Pseudo Corpora RWG*. However, they provide valuable additional knowledge which improves the performance of joint corpora.

The structure of the paper is as follows: the next section discusses related work; Section 3 describes the experimental setups including the used knowledge graphs, tools and evaluation datasets; Section 4 presents the results from the experiments; Section 5 concludes the paper.

## 2 Related Work

The content as well as the relation distribution within the knowledge graph play an important role in the successful performance of the various NLP tasks. One of the most approached tasks is Knowledge-based Word Sense Disambiguation (KWSD) — see (Agirre et al., 2014). Our own experiments showed that the manipulation of the knowledge graph with respect to adding new relations, combining various lexical relations or combining syntagmatic (corpus-based) and paradigmatic (lexicon-based) relations can improve the results in this task. For example, in (Simov et al., 2016b) we experimented with lexical relations from WN, glosses from eXtended WN as well as semantic relations, extracted from the syntactic annotation of SemCor (Miller et al., 1993). We reuse the sets of relations developed in these works to generate our *Pseudo Corpora LC*.

Goikoetxea et al. 2015 (Goikoetxea et al., 2015) describe an architecture in which a run of the Random Walk algorithm (Agirre et al., 2014) produces an artificial corpus from WordNet. The graph that is fed to the algorithm is composed of WordNet

synsets (the graph nodes) and of different types of relations between them (the graph arcs; some relation types are antonymy, hypernymy, derivation, etc.). This corpus is then fed into a shallow neural net which creates distributed word representations. The authors use the Continuous Bag of Words (CBOW) and Skip-gram algorithms as introduced in (Mikolov et al., 2013). They show that training on the artificial corpus gives improvements over text-trained models on some datasets (WordSim353 Similarity, WordSim353 Relatedness and SimLex-999). They also conclude that all explored methods are complementary to each other. In Goikoetxea et al. 2016 (Goikoetxea et al., 2016), the idea described above is further developed with the inclusion of a text corpus in addition to the corpus generated from WordNet. Learning is first performed on each of the two resources, and then various combination methods are introduced. The combined system outperforms the other systems on similarity, equals in relatedness, showing the advantages of simple combinations (like concatenation of independently learned embeddings) to the more complex ones.<sup>1</sup>

In addition to the usage of Random Walks algorithm, we promote the idea of using the lexical chains for handling graph contexts. In (Hirst and St-Onge, 1996) this idea was exploited for detecting and correcting malapropisms. Lexical chains introduce light context in a meaningful, i.e. not random way. WordNet was used as source for word similarity in order to form various types of lexical chains. Three types of relations have been constituted: extra-strong (literal word repetition), strong (between words in the same synsets, direct horizontal links<sup>2</sup> - similarity, antonymy, etc.) and medium-strong (defined by allowable paths in the knowledge graph, like between an apple and a carrot). The authors investigate only noun chains. The procedure is monotonic. First, an extra-strong relation is sought for a word. If found, it is added to the corresponding chain. If not, a strong relation is sought with search scope limitation. Similarly, medium-strong relation is sought with even stricter context window. If nothing is found, a new chain is started. The lexical chainer has been tested intrinsically (omissions and misplacements in chains) as well as extrinsically (in a spelling correction task).

<sup>1</sup>Due to space limitations we do not present an extensive overview of the literature on word embeddings.

<sup>2</sup>For the directions of the relations see next section.

### 3 Experimental Setup

Our experimental setup adopts the idea of the lexical chains as a mechanism for generation of *Pseudo Corpora LC*. There are some important differences from (Hirst and St-Onge, 1996). First, we use relations not only between nouns, but also among other parts-of-speech (POS)— noun (N), verb (V), adjective (A), and adverb (R). Second, we construct lexical chains over a knowledge graph, instead of constructing lexical chains over texts. Our experience in Word Sense Disambiguation task showed that usually two types of context play important role for the task (Mihalcea et al., 2004): (1) **Local context** — where the semantic measures are used for disambiguating words additionally connected by syntactic relations; and (2) **Global context** — where the semantic measures are employed to derive lexical chains, which are viewed as threads of meaning throughout an entire text. In usual text-based word embeddings only the local context is mainly considered. The creation of pseudo corpora on the basis of knowledge graphs additionally provides information about words that are usually distributed in the global context.

In order to simulate lexical chains in a *Pseudo Corpus LC* we exploited the formula of (Hirst and St-Onge, 1996) for calculation of the word similarity within the medium-strong relations:

$$weight = C - pathLen - k * numberChDir$$

$pathLen$  is the length of the path in the graph,  $numberChDir$  is the number of the changes in the relation path directions,  $C$  and  $k$  are parameters. In our first experiment, reported here, we consider as pseudo sentences in the *Pseudo Corpus LC* the paths starting from a given synset in WordNet and having weight above some threshold. We called such a path *lexical chain path*. Unfortunately, the number of paths starting at a given synset with appropriate weights has become exponentially huge. This prevented the practical usage of this set of paths. Thus, reduction of the number of paths generated for each synset was needed.

First, we define the number of occurrences of each relation in the lexical chain path. In such a way, the allowable paths are selected that provide bigger diversity of relations in the pseudo sentences. Even with these restrictions the number of the generated paths remained huge. Thus, we had to impose additional restrictions on the basis of the relation types. Two types of relations were

considered: *paradigmatic* and *syntagmatic*. We alternated the relations according to their types. For example, after up to three paradigmatic relations we allowed up to three syntagmatic ones.

#### 3.1 Knowledge Graph Relations

The WordNet-based KG (WN) has been constructed out of the relations in the Princeton WordNet (PWN3.0) and additional relations added by us from different sources. PWN3.0 groups together words in synsets, which in the knowledge graph are represented as nodes. The relation types that are possible between the different synsets are 16 original relations from PWN3.0 and 7 additional relations (Table 1<sup>3</sup>). We also consider their reverse relations which allows us to have more freedom for navigating over the knowledge graph.

The relation `defby` is taken from set of relations **WNG**<sup>4</sup>. It contains relations extracted from WN3.0 glosses, as the glosses are annotated with synset ids in eXtended WordNet (XWN) — (Mihalcea and Moldovan, 2001). The relations in **WNG** are constructed as co-occurrences of a synset and the synsets for open class words in its gloss. For example, from the gloss: “a set of data arranged in rows and columns” for the synset {table, tabular array} the following relations are extracted: {table, tabular array} `defby` {set ...}; {table, tabular array} `defby` {data ...}; {table, tabular array} `defby` {arrange ...}; etc.

The relations `coinc`, `moda`, `modn`, `modv`, `hpart`, and `condto` are extracted from two sources: logical representation of glosses in XWN and dependency analyses of sentences in SemCor corpus. For example, the synset {disyllable, dissyllable} is defined by “a word having two syllables.” The logical form for this gloss in XWN is the following

```
disyllable:NN(x1) ->
word:NN(x1) have:VB(e1, x1, x2)
two:JJ(x2) syllable:NN(x2)
```

In our opinion, each predicate that originates from a verbal, adjectival, adverbial, or prepositional lemma expresses an event. In the example, `have:VB(e1, x1, x2)` denote the event of “holding” of object denoted by  $x_2$  by the object denoted by  $x_1$ . Both of these objects are participants of the event of “holding”  $e_1$ . From this

<sup>3</sup>Each relation is represented by its acronym, name, number of instances, POS combination (X-X denote all combinations of POS.), type (Syntagmatic and Paradigmatic) and direction ( $\uparrow$  (UP),  $\downarrow$  (Down), and  $\leftrightarrow$  (Horizontal)).

<sup>4</sup>This set of relations is downloaded from the web page of UKB System: <http://ixa2.si.ehu.es/ukb/>

Acronym	Name	Number	POS	Type	Direction
WordNet Original Relations					
hyp	<i>hypernymy</i>	<b>89 089</b>	N-N, V-V	P	↑
ant	<i>antonymy</i>	<b>8 689</b>	A-A, N-N, R-R, V-V	P	↔
at	<i>attribute relation</i>	<b>886</b>	N-A, A-N	S	↔
cls	<i>a member of a class</i>	<b>9 420</b>	A-N, N-N, R-N, V-N	S	↑
cs	<i>cause</i>	<b>192</b>	V-V	S	↓
der	<i>derivational morphology</i>	<b>74 644</b>	A-N, N-A, N-N, N-V	S	↔
ent	<i>entailment</i>	<b>408</b>	V-V	S	↓
ins	<i>instance</i>	<b>8 576</b>	N-N	P	↑
mm	<i>member meronym</i>	<b>12 293</b>	N-N	P	↑
mp	<i>part meronym</i>	<b>9 097</b>	N-N	P	↑
ms	<i>substance meronym</i>	<b>797</b>	N-N	P	↑
per	<i>pertains</i>	<b>8 505</b>	A-N, R-A	S	↔
ppl	<i>participle</i>	<b>79</b>	A-V	S	↔
sa	<i>additional information</i>	<b>3 269</b>	A-A, V-V	P	↔
sim	<i>similar in meaning</i>	<b>21 386</b>	A-A	P	↔
vgp	<i>similar in meaning verb</i>	<b>1 725</b>	V-V	S	↔
Additional Relations					
defby	<i>defined by</i>	<b>419 387</b>	X-X	P	↑
coinc	<i>cooccurrences</i>	<b>93 221</b>	A-A, N-N, V-V, R-R	S	↔
moda	<i>adverb modifier</i>	<b>1 970</b>	R-A, R-V	S	↓
modn	<i>adjective modifier</i>	<b>69 857</b>	A-N, V-N	S	↓
modv	<i>adverb modifying verb</i>	<b>10 653</b>	R-V	S	↓
hpart	<i>has participant</i>	<b>128 444</b>	V-N	S	↓
condto	<i>condition to</i>	<b>2 494</b>	V-V	S	↑

Table 1: Relations in Knowledge Graph.

we extract the following relations: {have} hpart {word} — “word” is a participant in the event denoted by “have”; {have} hpart {syllable} — “syllable” is a participant in the event denoted by “have”; {word} coinc {syllable} — co-participants in the same event; and {two} moda {syllable} — “two” modifies “syllable”.

Similarly from {ice-cream cone} defined by “ice cream in a crisp conical wafer” the following logical form is presented:

```
ice-cream_cone:NN(x1) ->
  ice_cream:NN(x1) in:IN(x1, x2)
  crisp:JJ(x2) conical:JJ(x2)
  wafer:NN(x2)
```

From it we extracted the following relation: {crisp} coinc {conical} in the appropriate senses, because they modified the same noun.

The reverse relations have the same characteristics as the original ones, except for the direction which is changed to the opposite one. For some relations the reverse ones are not added because they are symmetric like ant relation. The knowledge graph used in the experiments is defined as

a set of nodes formed by the synsets of WN, and the arcs corresponds to all above relations. This knowledge graph is called **WNUnion**.

### 3.2 Pseudo-corpus Generation

For the pseudo corpus generation on the basis of the knowledge graph **WNUnion** we used a cascade vertical approach (fig. 1). We started with the generation of relation templates — combinations of relations without concrete synset ids. Here is an example of a path consisting of three relations: ○-modn-○-mp-○-hpart-1-○<sup>5</sup>. This determines a set of lexical chains that contain an adjective modifying a noun (local context), then part-of relation between two nouns (global context), and then connection between a noun and a verb where the noun refers to a participant predicated by the verb (local context). For each template we used a mapping to the knowledge graph and identified all possible

<sup>5</sup>In this case “-1” in hpart-1 denotes the reverse relation from a noun to a verb.

paths — the empty nodes (○) in templates were filled by synset identifiers in KG. For the template above we generated 8 866 possible combinations of synset ids. Here is one example: {02176178-a}-modn-{03614007-n}-mp-{03928116-n}-hpart-1-{00599992-v}. Finally we applied lemma substitution to each path and generated pseudo sentences. From the above example we generated the following pseudo sentence: `complex keyboard piano learn`. Keep in mind that the relation `hpart` (and thus `hpart-1`) does not distinguish between the syntactic roles of the event participants. Thus, in the example `piano` could be a direct object in a real sentence.

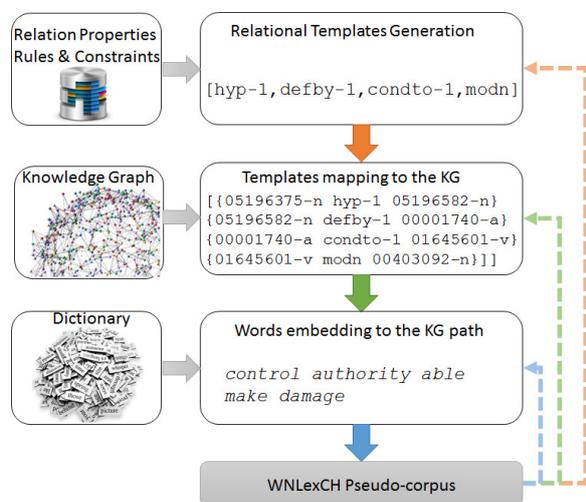


Figure 1: WNLexCH pseudo-corpus generation

The method was applied several times iterating the length of the generated templates from 2 up to the desired maximal pseudo sentence length. At each step the method reused the generated templates, paths and pseudo-sentences by joining them into a longer one. The combination of relations in the templates reflects their properties like direction, transitivity, allowed maximal number of repetitions, scope, relation type - paradigmatic and syntagmatic. In addition, some constraints for templates generation were used, like penalties for each direction change; consecutive relation and its opposite were allowed only for transitive relations etc. Note that in these experiments we used only pseudo-sentences generated on the first 3 steps because of the exploding size of the pseudo corpus. In the experiments we used subcorpora of **WNLexCH**, depending on the length of the paths involved in the corresponding subcorpora: **WN-**

**LexCH 2R** — paths of length 2 (466 285 957 pseudo sentences); **WNLexCH 3R** — paths of length 3 (733 998 728 pseudo sentences); **WNLexCH 4R** — paths of length 4 (143 416 212 pseudo sentences); and the whole corpus **WNLexCH**.

## 4 Experiment Results

The generated corpora have been fed into the Word2Vec tool<sup>6</sup> in order for the models to be trained. Initially, we performed experiments with different settings of the system parameters: context window size varying from 1 to 19 words, with the best results in most cases being for context window of 5 and context window of 15 words (reported below); iterations from 1 to 9, with best results for 7 iterations; negative examples set to 5; and frequency cut sampling was set to 7.

The evaluation of the word embeddings was done over the following datasets: *WordSim353 Similarity*, *WordSim353 Relatedness*<sup>7</sup>, and *SimLex-999*<sup>8</sup>. Each of the datasets consists of pair of words and numeric value of their similarity (*WordSim353 Similarity* and *SimLex-999*) or their relatedness (*WordSim353 Relatedness*). The numerical values were established on the basis of consultation with a number of human subjects. The evaluation was done in the following manner: first, the distance between the words in each pair was calculated on the basis of the corresponding word embedding; then, Spearman’s rank correlation between the predicted distances and the gold standard values was calculated.

We have performed a number of experiments through the different subcorpora as described above. As baselines, we evaluated two text-corpus-based word embeddings that are freely available on the web, as well as the best result of Goikoetxea et al. (Goikoetxea et al., 2015), available from the UKB web page<sup>9</sup>. Thus, the pseudo-corpus-based embeddings have been compared with text-based embeddings. We have selected two text-based sets of word vectors<sup>10</sup>: Google News trained over 100 billion running words — named **GoogleNews**; and Wikipedia dependency trained over context extracted from a dependency analysis of Wikipedia articles — named **Depen-**

<sup>6</sup><https://code.google.com/archive/p/word2vec/>

<sup>7</sup><http://alfonseca.org/eng/research/wordsim353.html>.

<sup>8</sup><https://www.cl.cam.ac.uk/fh295/simlex.html>

<sup>9</sup><http://ixa2.si.ehu.es/ukb/>

<sup>10</sup><https://github.com/3Top/word2vec-api>

<i>Embedding</i>	<i>WordSim353 Similarity</i>	<i>WordSim353 Relatedness</i>	<i>SimLex-999</i>
<b>GoogleNews</b>	0.77145	<b>0.61988</b>	0.44196
<b>Dependency</b>	0.76699	0.46764	0.44730
<b>WN+WNG<sub>best</sub></b>	<b>0.78670</b>	<b>0.61316</b>	<b>0.52479</b>
<b>WNLexCH 2R</b>	0.45172	0.29896	0.33593
<b>WNLexCH 3R</b>	0.66728	0.55904	0.48925
<b>WNLexCH 4R</b>	0.59042	0.41102	0.47418
<b>WNLexCH</b>	0.70038	0.52050	0.47907
<b>WNUnion Random</b>	0.74378	0.59988	0.51650
<b>WNUnion Random + WNLexCH 3R</b>	0.74998	<b>0.63375</b>	<b>0.52302</b>
<b>WNUnion Random + WNLexCH</b>	<b>0.75916</b>	0.62848	0.52125

Table 2: Results from experiments with different extensions of the WordNet knowledge graph. All our embeddings were trained with the same options for word2vec.

**dency.** The results for these baselines are presented in the first three rows of Table 2.

The experiments were divided into two groups: (1) experiments by our algorithm with corpora generated with different numbers of relations; and (2) experiments by the UKB system with a pseudo corpus generated over the same knowledge graph, and its combinations with the corpora generated by our algorithm. In Table 2 we present some results for each type of the experiments.

The results from the experiments show that increasing the length of the lexical chain path improves the results of the word embeddings with respect to both similarity and relatedness measures. In order to compare our approach to the *Pseudo Corpus RWG* we have generated a pseudo corpus over the same knowledge graph **WNUnion**. The result shows that the *Pseudo Corpus RWG* is better in comparison to the *Pseudo Corpus LC* with its length of the paths up to four. But on the other hand, the combination of the *Pseudo Corpus RWG* and the *Pseudo Corpus LC* produces better results than each of them separately — the last two rows in Table 2.

## 5 Conclusion

The paper presents an approach to the generation of pseudo corpora on the basis of lexical chain templates over knowledge graphs. We showed that the knowledge within the closer neighborhood of the synsets plays an important role for the similarity and relatedness evaluation. In some cases they outperformed the embeddings trained on text corpora. The improvement over Pseudo Corpora RWG, generated on the same KG, shows also that

longer lexical chain paths are necessary. In our view the main reason for the current results is the fact that the pseudo corpus **WNLexCH** reflects all the relations in the local context of the synsets in the KG. In our work (Simov et al., 2017) we demonstrated that different combinations of relations in the KG generally improved performance of similarity and relatedness. Our hope is that knowledge learned by the embeddings over *Pseudo Corpus LC* is more homogeneous comparing to embeddings over text corpora where the problem is with the infrequent words. In these cases the embeddings learned less features than the ones for the frequent ones. The problem with the embeddings over *Pseudo Corpus RWG* is that there is no way to control the coverage.

Our research is in-line with the current tendency of building artificial corpora for various NLP tasks due to sparseness and bias of the available data. In future, we envisage to do experiments with selections of longer lexical chain paths. Also we will perform extrinsic evaluation incorporating the various embeddings in Neural Network systems for different NLP tasks. On the basis of the results we plan to do experiments with different lexical chain templates in order to determine the most appropriate models for each target NLP task.

## Acknowledgements

This research has received partial support by the grant 02/12 — *Deep Models of Semantic Knowledge (DemoSem)*, funded by the Bulgarian National Science Fund in 2017–2019. We are grateful to the anonymous reviewers for their remarks, comments, and suggestions.

## References

- Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.* 40(1):57–84.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London.
- 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*. ACM, New York, NY, USA, WWW '01, pages 406–414.
- Josu Goikoetxea, Eneko Agirre, and Aitor Soroa. 2016. Single or multiple? combining word representations independently learned from text and wordnet. In *AAAI*. AAAI Press, pages 2608–2614.
- Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *HLT-NAACL*. The Association for Computational Linguistics, pages 1434–1439.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.
- Graeme Hirst and David St-Onge. 1996. *Lexical chains as representations of context for the detection and correction of malapropisms*, The MIT Press.
- Rada Mihalcea and Dan I. Moldovan. 2001. extended wordnet: progress report. In *in Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*. pages 95–100.
- Rada Mihalcea, Paul Tarau, and Elizabeth Figa. 2004. Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of Coling 2004*. COLING, Geneva, Switzerland, pages 1126–1132.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.
- George A. Miller, Claudia Leacock, Randee Teng, and Ross T. Bunker. 1993. A semantic concordance. In *Proc. of HLT '93*. pages 303–308.
- Petar Ristoski and Heiko Paulheim. 2016. Rdf2vec: Rdf graph embeddings for data mining. In *International Semantic Web Conference*. Springer, pages 498–514.
- Kiril Simov, Petya Osenova, and Alexander Popov. 2016a. Using context information for knowledge-based word sense disambiguation. In Christo Dichev and Gennady Agre, editors, *Proceedings of Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2016)*. Springer International Publishing, Cham, pages 130–139.
- Kiril Simov, Petya Osenova, and Alexander Popov. 2017. Comparison of word embeddings from different knowledge graphs. In *Proceedings of Language, Data, and Knowledge Conference*. Springer International Publishing, pages 213–221.
- Kiril Simov, Alexander Popov, and Petya Osenova. 2016b. The role of the wordnet relations in the knowledge-based word sense disambiguation task. In *Proceedings of Eighth Global WordNet Conference*. pages 391–398.