Negative Sampling Improves Hypernymy Extraction Based on Projection Learning

Dmitry Ustalov[†], **Nikolay Arefyev**[§], **Chris Biemann**[‡], **and Alexander Panchenko**[‡]

[†]Ural Federal University, Institute of Natural Sciences and Mathematics, Russia
[§]Moscow State University, Faculty of Computational Mathematics and Cybernetics, Russia
[‡]University of Hamburg, Deptartment of Informatics, Language Technology Group, Germany
dmitry.ustalov@urfu.ru, narefjev@cs.msu.ru
{biemann,panchenko}@informatik.uni-hamburg.de

Abstract

We present a new approach to extraction of hypernyms based on projection learning and word embeddings. In contrast to classification-based approaches, projection-based methods require no candidate hyponym-hypernym pairs. While it is natural to use both positive and negative training examples in supervised relation extraction, the impact of negative examples on hypernym prediction was not studied so far. In this paper, we show that explicit negative examples used for regularization of the model significantly improve performance compared to the state-of-the-art approach of Fu et al. (2014) on three datasets from different languages.

1 Introduction

Hypernyms are useful in many natural language processing tasks ranging from construction of taxonomies (Snow et al., 2006; Panchenko et al., 2016a) to query expansion (Gong et al., 2005) and question answering (Zhou et al., 2013). Automatic extraction of hypernyms from text has been an active area of research since manually constructed high-quality resources featuring hypernyms, such as WordNet (Miller, 1995), are not available for many domain-language pairs.

The drawback of pattern-based approaches to hypernymy extraction (Hearst, 1992) is their sparsity. Approaches that rely on the classification of pairs of word embeddings (Levy et al., 2015) aim to tackle this shortcoming, but they require candidate hyponym-hypernym pairs. We explore a hypernymy extraction approach that requires no candidate pairs. Instead, the method performs prediction of a hypernym embedding on the basis of a hyponym embedding.

The contribution of this paper is a novel approach for hypernymy extraction based on projection learning. Namely, we present an improved version of the model proposed by Fu et al. (2014), which makes use of both positive and negative training instances enforcing the asymmetry of the projection. The proposed model is generic and could be straightforwardly used in other relation extraction tasks where both positive and negative training samples are available. Finally, we are the first to successfully apply projection learning for hypernymy extraction in a morphologically rich language. An implementation of our approach and the pre-trained models are available online.[1]

2 Related Work

Path-based methods for hypernymy extraction rely on sentences where both hyponym and hypernym co-occur in characteristic contexts, e.g., “such *cars* as *Mercedes* and *Audi*”. Hearst (1992) proposed to use hand-crafted lexical-syntactic patterns to extract hypernyms from such contexts. Snow et al. (2004) introduced a method for learning patterns automatically based on a set of seed hyponym-hypernym pairs. Further examples of path-based approaches include (Tjong Kim Sang and Hofmann, 2009) and (Navigli and Velardi, 2010). The inherent limitation of the path-based methods leading to sparsity issues is that hyponym and hypernym have to co-occur in the same sentence.

Methods based on distributional vectors, such as those generated using the *word2vec* toolkit (Mikolov et al., 2013b), aim to overcome this sparsity issue as they require no hyponym-hypernym co-occurrence in a sentence. Such methods take representations of individual words as an input to predict relations between them.

[1] http://github.com/nlpub/projlearn

Two branches of methods relying on distributional representations emerged so far.

Methods based on word pair classification take an ordered pair of word embeddings (a candidate hyponym-hypernym pair) as an input and output a binary label indicating a presence of the hypernymy relation between the words. Typically, a binary classifier is trained on concatenation or subtraction of the input embeddings, cf. (Roller et al., 2014). Further examples of such methods include (Lenci and Benotto, 2012; Weeds et al., 2014; Levy et al., 2015; Vylomova et al., 2016).

HypeNET (Shwartz et al., 2016) is a hybrid approach which is also based on a classifier, but in addition to two word embeddings a third vector is used. It represents path-based syntactic information encoded using an LSTM model (Hochreiter and Schmidhuber, 1997). Their results significantly outperform the ones from previous path-based work of Snow et al. (2004).

An inherent limitation of classification-based approaches is that they require a list of candidate words pairs. While these are given in evaluation datasets such as BLESS (Baroni and Lenci, 2011), a corpus-wide classification of relations would need to classify all possible word pairs, which is computationally expensive for large vocabularies. Besides, Levy et al. (2015) discovered a tendency to lexical memorization of such approaches hampering the generalization.

Methods based on projection learning take one hyponym word vector as an input and output a word vector in a topological vicinity of hypernym word vectors. Scaling this to the vocabulary, there is only one such operation per word. Mikolov et al. (2013a) used projection learning for bilingual word translation. Vulić and Korhonen (2016) presented a systematic study of four classes of methods for learning bilingual embeddings including those based on projection learning.

Fu et al. (2014) were first to apply projection learning for hypernym extraction. Their approach is to learn an affine transformation of a hyponym into a hypernym word vector. The training of their model is performed with stochastic gradient descent. The k-means clustering algorithm is used to split the training relations into several groups. One transformation is learned for each group, which can account for the possibility that the projection of the relation depends on a subspace. This state-of-the-art approach serves as the baseline in our experiments.

Nayak (2015) performed evaluations of distributional hypernym extractors based on classification and projection methods (yet on different datasets, so these approaches are not directly comparable). The best performing projection-based architecture proposed in this experiment is a four-layered feed-forward neural network. No clustering of relations was used. The author used negative samples in the model by adding a regularization term in the loss function. However, drawing negative examples uniformly from the vocabulary turned out to hamper performance. In contrast, our approach shows significant improvements using manually created synonyms and hyponyms as negative samples.

Yamane et al. (2016) introduced several improvements of the model of Fu et al. (2014). Their model jointly learns projections and clusters by dynamically adding new clusters during training. They also used automatically generated negative instances via a regularization term in the loss function. In contrast to Nayak (2015), negative samples are selected not randomly, but among nearest neighbors of the predicted hypernym. Their approach compares favorably to (Fu et al., 2014), yet the contribution of the negative samples was not studied. Key differences of our approach from (Yamane et al., 2016) are (1) use of explicit as opposed to automatically generated negative samples, (2) enforcement of asymmetry of the projection matrix via re-projection. While our experiments are based on the model of Fu et al. (2014), our regularizers can be straightforwardly integrated into the model of Yamane et al. (2016).

3 Hypernymy Extraction via Regularized Projection Learning

3.1 Baseline Approach

In our experiments, we use the model of Fu et al. (2014) as the baseline. In this approach, the projection matrix $\mathbf{\Phi}^*$ is obtained similarly to the linear regression problem, i.e., for the given row word vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ representing correspondingly hyponym and hypernym, the square matrix $\mathbf{\Phi}^*$ is fit on the training set of positive pairs $\mathcal{P}$:

$$\mathbf{\Phi}^* = \arg\min_{\mathbf{\Phi}} \frac{1}{|\mathcal{P}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{P}} \|\boldsymbol{x}\mathbf{\Phi} - \boldsymbol{y}\|^2 ,$$

where $|\mathcal{P}|$ is the number of training examples and $\|\boldsymbol{x}\mathbf{\Phi} - \boldsymbol{y}\|$ is the distance between a pair of row vectors $\boldsymbol{x}\boldsymbol{\Phi}$ and $\boldsymbol{y}$. In the original method, the L^2 distance is used. To improve performance, k projection matrices $\boldsymbol{\Phi}$ are learned one for each cluster of relations in the training set. One example is represented by a hyponym-hypernym offset. Clustering is performed using the k-means algorithm (MacQueen, 1967).

3.2 Linguistic Constraints via Regularization

The nearest neighbors generated using distributional word vectors tend to contain a mixture of synonyms, hypernyms, co-hyponyms and other related words (Wandmacher, 2005; Heylen et al., 2008; Panchenko, 2011). In order to explicitly provide examples of undesired relations to the model, we propose two improved versions of the baseline model: *asymmetric regularization* that uses inverted relations as negative examples, and *neighbor regularization* that uses relations of other types as negative examples. For that, we add a regularization term to the loss function:

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}} \frac{1}{|\mathcal{P}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{P}} \|\boldsymbol{x}\boldsymbol{\Phi} - \boldsymbol{y}\|^2 + \lambda R,$$

where λ is the constant controlling the importance of the regularization term R.

Asymmetric Regularization. As hypernymy is an asymmetric relation, our first method enforces the asymmetry of the projection matrix. Applying the same transformation to the predicted hypernym vector $\boldsymbol{x}\boldsymbol{\Phi}$ should not provide a vector similar $(\cdot)$ to the initial hyponym vector $\boldsymbol{x}$. Note that, this regularizer requires only positive examples $\mathcal{P}$:

$$R = \frac{1}{|\mathcal{P}|} \sum_{(\boldsymbol{x},\_)\in\mathcal{P}} (\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi} \cdot \boldsymbol{x})^2.$$

Neighbor Regularization. This approach relies on the negative sampling by explicitly providing the examples of semantically related words $\boldsymbol{z}$ of the hyponym $\boldsymbol{x}$ that penalizes the matrix to produce the vectors similar to them:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{N}} (\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi} \cdot \boldsymbol{z})^2.$$

Note that this regularizer requires negative samples $\mathcal{N}$. In our experiments, we use synonyms of hyponyms as $\mathcal{N}$, but other types of relations can be also used such as antonyms, meronyms or co-hyponyms. Certain words might have no synonyms in the training set. In such cases, we substitute $\boldsymbol{z}$ with $\boldsymbol{x}$, gracefully reducing to the previous variation. Otherwise, on each training epoch, we sample a random synonym of the given word.

Regularizers without Re-Projection. In addition to the two regularizers described above, that rely on re-projection of the hyponym vector ($\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$), we also tested two regularizers without re-projection, denoted as $\boldsymbol{x}\boldsymbol{\Phi}$. The neighbor regularizer in this variation is defined as follows:

$$R = \frac{1}{|\mathcal{N}|} \sum_{(\boldsymbol{x},\boldsymbol{z})\in\mathcal{N}} (\boldsymbol{x}\boldsymbol{\Phi} \cdot \boldsymbol{z})^2.$$

In our case, this regularizer penalizes relatedness of the predicted hypernym $\boldsymbol{x}\boldsymbol{\Phi}$ to the synonym $\boldsymbol{z}$. The asymmetric regularizer without re-projection is defined in a similar way.

3.3 Training of the Models

To learn parameters of the considered models we used the Adam method (Kingma and Ba, 2014) with the default meta-parameters as implemented in the TensorFlow framework (Abadi et al., 2016).[2] We ran 700 training epochs passing a batch of 1024 examples to the optimizer. We initialized elements of each projection matrix using the normal distribution $\mathcal{N}(0, 0.1)$.

4 Results

4.1 Evaluation Metrics

In order to assess the quality of the model, we adopted the hit@l measure proposed by Frome et al. (2013) which was originally used for image tagging. For each subsumption pair $(\boldsymbol{x}, \boldsymbol{y})$ composed of the hyponym $\boldsymbol{x}$ and the hypernym $\boldsymbol{y}$ in the test set $\mathcal{P}$, we compute l nearest neighbors for the projected hypernym $\boldsymbol{x}\boldsymbol{\Phi}^*$. The pair is considered matched if the gold hypernym $\boldsymbol{y}$ appears in the computed list of the l nearest neighbors $\mathrm{NN}_l(\boldsymbol{x}\boldsymbol{\Phi}^*)$. To obtain the quality score, we average the matches in the test set $\mathcal{P}$:

$$\text{hit@}l = \frac{1}{|\mathcal{P}|} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{P}} \mathbb{1}\big(\boldsymbol{y} \in \mathrm{NN}_l(\boldsymbol{x}\boldsymbol{\Phi}^*)\big),$$

where $\mathbb{1}(\cdot)$ is the indicator function. To consider also the rank of the correct answer, we compute the area under curve measure as the area under the $l-1$ trapezoids:

$$\text{AUC} = \frac{1}{2}\sum_{i=1}^{l-1}(\text{hit@}(i) + \text{hit@}(i+1)).$$

4.2 Experiment 1: The Russian Language

Dataset. In this experiment, we use word embeddings published as a part of the Russian Dis-

[2] `https://www.tensorflow.org`

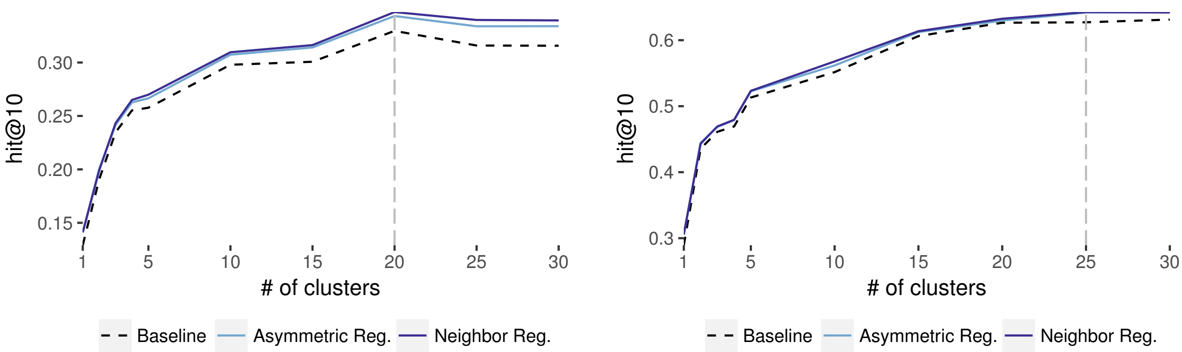

Figure 1: Performance of our models with re-projection as compared to the baseline approach of (Fu et al., 2014) according to the hit@10 measure for Russian (left) and English (right) on the validation set.

Model		hit@1	hit@5	hit@10	AUC
Baseline		0.209	0.303	0.323	2.665
Asym. Reg.	$\boldsymbol{x\Phi}$	0.213	0.300	0.322	2.659
Asym. Reg.	$\boldsymbol{x\Phi\Phi}$	0.212	0.312	0.334	2.743
Neig. Reg.	$\boldsymbol{x\Phi}$	**0.214**	0.304	0.325	2.685
Neig. Reg.	$\boldsymbol{x\Phi\Phi}$	0.211	**0.315**	**0.338**	**2.768**

Table 1: Performance of our approach for Russian for $k = 20$ clusters compared to (Fu et al., 2014).

tributional Thesaurus (Panchenko et al., 2016b) trained on 12.9 billion token collection of Russian books. The embeddings were trained using the skip-gram model (Mikolov et al., 2013b) with 500 dimensions and a context window of 10 words.

The dataset used in our experiments has been composed of two sources. We extracted synonyms and hypernyms from the Wiktionary[3] using the Wikokit toolkit (Krizhanovsky and Smirnov, 2013). To enrich the lexical coverage of the dataset, we extracted additional hypernyms from the same corpus using Hearst patterns for Russian using the PatternSim toolkit (Panchenko et al., 2012).[4] To filter noisy extractions, we used only relations extracted more than 100 times.

As suggested by Levy et al. (2015), we split the train and test sets such that each contains a distinct vocabulary to avoid the lexical overfitting. This results in 25 067 training, 8 192 validation, and 8 310 test examples. The validation and test sets contain hypernyms from Wiktionary, while the training set is composed of hypernyms and synonyms coming from both sources.

Discussion of Results. Figure 1 (left) shows performance of the three projection learning setups on the validation set: the baseline approach, the asymmetric regularization approach, and the neighbor regularization approach. Both regularization strategies lead to consistent improvements over the non-regularized baseline of (Fu et al., 2014) across various cluster sizes. The method reaches optimal performance for $k = 20$ clusters. Table 1 provides a detailed comparison of the performance metrics for this setting. Our approach based on the regularization using synonyms as negative samples outperform the baseline (all differences between the baseline and our models are significant with respect to the t-test). According to all metrics, but hit@1 for which results are comparable to $\boldsymbol{x\Phi}$, the re-projection ($\boldsymbol{x\Phi\Phi}$) improves results.

4.3 Experiment 2: The English Language

We performed the evaluation on two datasets.

EVALution Dataset. In this evaluation, word embeddings were trained on a 6.3 billion token text collection composed of Wikipedia, ukWaC (Ferraresi et al., 2008), Gigaword (Graff, 2003), and news corpora from the Leipzig Collection (Goldhahn et al., 2012). We used the skip-gram model with the context window size of 8 tokens and 300-dimensional vectors.

We use the EVALution dataset (Santus et al., 2015) for training and testing the model, composed of 1 449 hypernyms and 520 synonyms, where hypernyms are split into 944 training, 65 validation and 440 test pairs. Similarly to the first experiment, we extracted extra training hypernyms using the Hearst patterns, but in contrast to Russian, they did not improve the results significantly, so we left them out for English. A reason for such difference could be the more complex morphological system of Russian, where each word has more morphological variants compared

[3] http://www.wiktionary.org

[4] https://github.com/cental/patternsim

		EVALution					EVALution, BLESS, K&H+N, ROOT09				
Model		k	**hit@1**	**hit@5**	**hit@10**	**AUC**	k	**hit@1**	**hit@5**	**hit@10**	**AUC**
Baseline		1	0.109	0.118	0.120	1.052	1	0.104	0.247	0.290	2.115
Asymmetric Reg.	$\boldsymbol{x}\boldsymbol{\Phi}$	1	0.116	0.125	0.132	1.140	1	0.132	0.256	0.292	2.204
Asymmetric Reg.	$\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$	1	0.145	0.166	0.173	1.466	1	0.112	**0.266**	0.314	2.267
Neighbor Reg.	$\boldsymbol{x}\boldsymbol{\Phi}$	1	0.134	0.141	0.150	1.280	1	**0.134**	0.255	0.306	2.267
Neighbor Reg.	$\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$	1	**0.148**	**0.168**	**0.177**	**1.494**	1	0.111	0.264	**0.316**	**2.273**
Baseline		30	0.327	0.339	0.350	3.080	25	0.546	0.614	0.634	5.481
Asymmetric Reg.	$\boldsymbol{x}\boldsymbol{\Phi}$	30	0.336	0.354	0.366	3.201	25	0.547	0.616	0.632	5.492
Asymmetric Reg.	$\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$	30	0.341	0.364	0.368	3.255	25	**0.553**	0.621	**0.642**	5.543
Neighbor Reg.	$\boldsymbol{x}\boldsymbol{\Phi}$	30	0.339	0.357	0.364	3.210	25	0.547	0.617	0.634	5.494
Neighbor Reg.	$\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$	30	**0.345**	**0.366**	**0.370**	**3.276**	25	**0.553**	**0.623**	0.641	**5.547**

Table 2: Performance of our approach for English without clustering ($k = 1$) and with the optimal number of cluster on the EVALution datasets ($k = 30$) and on the combined datasets ($k = 25$).

to English. Therefore, extra training samples are needed for Russian (embeddings of Russian were trained on a non-lemmatized corpus).

Combined Dataset. To show the robustness of our approach across configurations, this dataset has more training instances, different embeddings, and both synonyms and co-hyponyms as negative samples. We used hypernyms, synonyms and co-hyponyms from the four commonly used datasets: EVALution, BLESS (Baroni and Lenci, 2011), ROOT09 (Santus et al., 2016) and K&H+N (Necsulescu et al., 2015).The obtained 14 528 relations were split into 9 959 training, 1 631 validation and 1 625 test hypernyms; 1 313 synonyms and co-hyponyms were used as negative samples. We used the standard 300-dimensional embeddings trained on the 100 billion tokens Google News corpus (Mikolov et al., 2013b).

Discussion of Results. Figure 1 (right) shows that similarly to Russian, both regularization strategies lead to consistent improvements over the non-regularized baseline. Table 2 presents detailed results for both English datasets. Similarly to the first experiment, our approach consistently improves results robustly across various configurations. As we change the number of clusters, types of embeddings, the size of the training data and type of relations used for negative sampling, results using our method stay superior to those of the baseline. The regularizers without re-projection ($\boldsymbol{x}\boldsymbol{\Phi}$) obtain lower results in most configurations as compared to re-projected versions ($\boldsymbol{x}\boldsymbol{\Phi}\boldsymbol{\Phi}$). Overall, the neighbor regularization yields slightly better results in comparison to the asymmetric regularization. We attribute this to the fact that some synonyms $\boldsymbol{z}$ are close to the original hyponym $\boldsymbol{x}$, while others can be distant. Thus, neighbor regularization is able to safeguard the model during training from more errors. This is also a likely reason why the performance of both regularizers is similar: the asymmetric regularization makes sure that a re-projected vector does not belong to a semantic neighborhood of the hyponym. Yet, this is exactly what neighbor regularization achieves. Note, however that neighbor regularization requires explicit negative examples, while asymmetric regularization does not.

5 Conclusion

In this study, we presented a new model for extraction of hypernymy relations based on the projection of distributional word vectors. The model incorporates information about explicit negative training instances represented by relations of other types, such as synonyms and co-hyponyms, and enforces asymmetry of the projection operation. Our experiments in the context of the hypernymy prediction task for English and Russian languages show significant improvements of the proposed approach over the state-of-the-art model without negative sampling.

Acknowledgments

We acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) foundation under the "JOIN-T" project, the Deutscher Akademischer Austauschdienst (DAAD), the Russian Foundation for Basic Research (RFBR) under the project no. 16-37-00354 mol_a, and the Russian Foundation for Humanities under the project no. 16-04-12019 "RussNet and YARN thesauri integration". We also thank Microsoft for providing computational resources under the Microsoft Azure for Research award. Finally, we are grateful to Benjamin Milde, Andrey Kutuzov, Andrew Krizhanovsky, and Martin Riedl for discussions and suggestions related to this study.

References

Martín Abadi et al. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, abs/1603.04467.

Marco Baroni and Alessandro Lenci. 2011. How We BLESSed Distributional Semantic Evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10, Edinburgh, Scotland. Association for Computational Linguistics.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukWaC, a very large Web-derived corpus of English. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4): Can we beat Google?*, pages 47–54, Marakech, Morocco.

Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., Harrahs and Harveys, NV, USA.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning Semantic Hierarchies via Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, MD, USA. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Zhiguo Gong, Chan Wa Cheang, and U. Leong Hou. 2005. Web Query Expansion by WordNet. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications - DEXA '05*, pages 166–175. Springer Berlin Heidelberg, Copenhagen, Denmark.

David Graff. 2003. English Gigaword. Technical Report LDC2003T05, Linguistic Data Consortium, Philadelphia, PA, USA.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING'92, pages 539–545, Nantes, France. Association for Computational Linguistics.

Kris Heylen, Yves Peirsman, Dirk Geeraerts, and Dirk Speelman. 2008. Modelling Word Similarity: an Evaluation of Automatic Synonymy Extraction Algorithms. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 3243–3249, Marrakech, Morocco. European Language Resources Association (ELRA).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Andrew A. Krizhanovsky and Alexander V. Smirnov. 2013. An approach to automated construction of a general-purpose lexical ontology based on Wiktionary. *Journal of Computer and Systems Sciences International*, 52(2):215–225.

Alessandro Lenci and Giulia Benotto. 2012. Identifying Hypernyms in Distributional Semantic Spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 75–79, Montréal, Canada. Association for Computational Linguistics.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, USA. Association for Computational Linguistics.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, California, USA. University of California Press.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013a. Exploiting Similarities among Languages for Machine Translation. *CoRR*, abs/1309.4168.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., Harrahs and Harveys, NV, USA.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for Definition and Hypernym Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1318–1327, Uppsala, Sweden. Association for Computational Linguistics.

Neha Nayak. 2015. Learning Hypernymy over Word Embeddings. Technical report, Stanford University.

Silvia Necsulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading Between the Lines: Overcoming Data Sparsity for Accurate Classification of Lexical Relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 182–192, Denver, CO, USA. Association for Computational Linguistics.

Alexander Panchenko, Olga Morozova, and Hubert Naets. 2012. A Semantic Similarity Measure Based on Lexico-Syntactic Patterns. In *Proceedings of KONVENS 2012*, pages 174–178, Vienna, Austria. ÖGAI.

Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cedrick Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016a. TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327, San Diego, CA, USA. Association for Computational Linguistics.

Alexander Panchenko, Dmitry Ustalov, Nikolay Arefyev, Denis Paperno, Natalia Konstantinova, Natalia Loukachevitch, and Chris Biemann. 2016b. Human and Machine Judgements for Russian Semantic Relatedness. In *Proceedings of the 5th Conference on Analysis of Images, Social Networks and Texts (AIST'2016)*, volume 661 of *Communications in Computer and Information Science*, pages 303–317, Yekaterinburg, Russia. Springer-Verlag Berlin Heidelberg.

Alexander Panchenko. 2011. Comparison of the Baseline Knowledge-, Corpus-, and Web-based Similarity Measures for Semantic Relations Extraction. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 11–21, Edinburgh, UK. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1025–1036, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China. Association for Computational Linguistics.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Nine Features in a Random Forest to Learn Taxonomical Semantic Relations. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4557–4564, Portorož, Slovenia. European Language Resources Association (ELRA).

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving Hypernymy Detection with an Integrated Path-based and Distributional Method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2389–2398, Berlin, Germany. Association for Computational Linguistics.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 1297–1304, Vancouver, British Columbia, Canada. MIT Press.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 801–808, Sydney, Australia. Association for Computational Linguistics.

Erik Tjong Kim Sang and Katja Hofmann. 2009. Lexical Patterns or Dependency Patterns: Which Is Better for Hypernym Extraction? In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 174–182, Boulder, Colorado, USA. Association for Computational Linguistics.

Ivan Vulić and Anna Korhonen. 2016. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 247–257, Berlin, Germany. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and Took, Gaggle and Goose, Book and Read: Evaluating the Utility of Vector Differences for Lexical Relation Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1671–1682, Berlin, Germany. Association for Computational Linguistics.

Tonio Wandmacher. 2005. How semantic is Latent Semantic Analysis? In *Proceedings of RÉCITAL 2005*, pages 525–534, Dourdan, France.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to Distinguish Hypernyms and Co-Hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Josuke Yamane, Tomoya Takatani, Hitoshi Yamada, Makoto Miwa, and Yutaka Sasaki. 2016. Distributional Hypernym Generation by Jointly Learning Clusters and Projections. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1871–1879, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. 2013. Improving Question Retrieval in Community Question Answering Using World Knowledge. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2239–2245, Beijing, China. AAAI Press.