

Using Distributional Thesaurus Embedding for Co-hyponymy Detection

Abhik Jana, Nikhil Reddy Varimalla and Pawan Goyal

Universität Hamburg, Indian Institute of Technology Kharagpur, Indian Institute of Technology Kharagpur
jana@informatik.uni-hamburg.de, nikhil.varimala@gmail.com, pawang@cse.iitkgp.ac.in

Abstract

Discriminating lexical relations among distributionally similar words has always been a challenge for natural language processing (NLP) community. In this paper, we investigate whether the network embedding of distributional thesaurus can be effectively utilized to detect co-hyponymy relations. By extensive experiments over three benchmark datasets, we show that the vector representation obtained by applying node2vec on distributional thesaurus outperforms the state-of-the-art models for binary classification of co-hyponymy vs. hypernymy, as well as co-hyponymy vs. meronymy, by huge margins.

Keywords: Co-hyponymy detection, Distributional Thesaurus, Network Embedding

1. Introduction

Distributional semantic models are used in a wide variety of tasks like sentiment analysis, word sense disambiguation, predicting semantic compositionality, etc. Automatic detection of lexical relations is one such fundamental task which can be leveraged in applications like paraphrasing, ontology building, metaphor detection etc. Both supervised and unsupervised methods have been proposed by the researchers to identify lexical relations like hypernymy, co-hyponymy, meronymy etc. over the years. Recent attempts to solve this task deal with proposing similarity measures based on distributional semantic models (Roller et al., 2014; Weeds et al., 2014; Santus et al., 2016; Shwartz et al., 2017; Roller and Erk, 2016). For hypernymy detection, several works use distributional inclusion hypothesis (Geffet and Dagan, 2005), entropy-based distributional measure (Santus et al., 2014) as well as several embedding schemes (Fu et al., 2014; Yu et al., 2015; Nguyen et al., 2017). Image generality for lexical entailment detection (Kiela et al., 2015) has also been tried out for the same purpose. As far as meronymy detection is concerned, most of the attempts are pattern based (Berland and Charniak, 1999; Girju et al., 2006; Pantel and Pennacchiotti, 2006) along with some recent works exploring the possibility of using distributional semantic models (Morlane-Hondère, 2015).

Similarly, for co-hyponymy detection, researchers have investigated the usefulness of several distributional semantic models. One such attempt is made by Weeds et al. (2014), where they proposed a supervised framework and used several vector operations as features for the classification of hypernymy and co-hyponymy. Santus et al. (2016) proposed a supervised method based on a Random Forest algorithm to learn taxonomical semantic relations and they have shown that the model performs well for co-hyponymy detection. In another attempt, Jana and Goyal (2018b) proposed various complex network measures which can be used as features to build a supervised classifier model for co-hyponymy detection, and showed improvements over other baseline approaches. Recently, with the emergence of various network representation learning methods (Perozzi et al., 2014; Tang et al., 2015; Grover and Leskovec, 2016; Ribeiro et al., 2017), attempts have been made to convert

distributional thesauri network into low dimensional vector space. (Ferret, 2017) apply distributional thesaurus embedding for synonym extraction and expansion tasks whereas Jana and Goyal (2018a) use it to improve the state-of-the-art performance of word similarity/relatedness tasks, word analogy task etc.

Thus, a natural question arises as to whether network embeddings should be more effective than the handcrafted network features used by Jana and Goyal (2018b) for co-hyponymy detection. Being motivated by this connection, we investigate how the information captured by network representation learning methodologies on distributional thesaurus can be used in discriminating word pairs having co-hyponymy relation from the word pairs having hypernymy, meronymy relation or any random pair of words. We use the distributional thesaurus (DT) network (Riedl and Biemann, 2013) built using Google books syntactic n-grams. As a network representation learning method, we apply node2vec (Grover and Leskovec, 2016) which is an algorithmic framework for learning continuous feature representations for nodes in networks that maximizes the likelihood of preserving network neighborhoods of nodes. Thus obtained vectors are then used as feature vectors and plugged into the classifiers according to the state-of-the-art experimental setup.

Classification model: To distinguish the word pairs having co-hyponymy relation from the word pairs having hypernymy or meronymy relation, or from any random pair of words, we combine the network embeddings of the two words by concatenation (CC) and addition (ADD) operations to provide as features to train classifiers like Support Vector Machine (SVM) and Random Forest (RF).

Evaluation results: We evaluate the usefulness of DT embeddings against three benchmark datasets for co-hyponymy detection (Weeds et al., 2014; Santus et al., 2016; Jana and Goyal, 2018b), following their experimental setup. We show that the network embeddings outperform the baselines by a huge margin throughout all the experiments, except for co-hyponyms vs. random pairs, where the baselines already have very high accuracy and network embeddings are able to match the results.

2. Methodology

We take the distributional thesaurus (DT) (Riedl and Biemann, 2013) constructed from the Google books syntactic n-grams data (Goldberg and Orwant, 2013) spanning from 1520 to 2008 as the underlying network where each word’s neighborhood is represented by a list of top 200 words that are similar with respect to their bi-gram distribution (Riedl and Biemann, 2013).

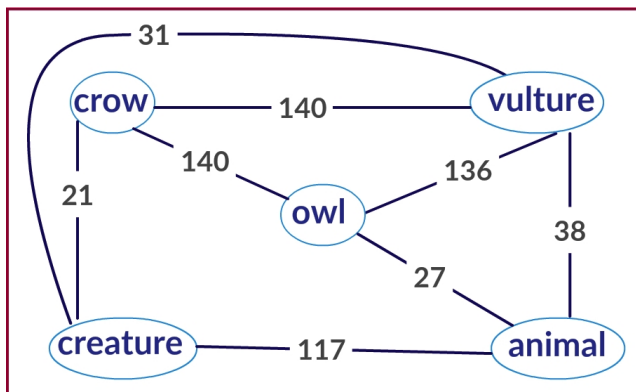


Figure 1: A sample snapshot of distributional thesaurus (DT) network, where each node represents a word and the weight of edge between two nodes is defined as the number of context features that these two words share in common. Here the word ‘owl’ shares more context features with its co-hyponyms – ‘crow’, ‘vulture’ compared to their hypernym ‘animal’.

The nodes in the network represent words and edges are present between a node and its top 200 similar nodes; the number of features that two nodes share in common is assigned as the weight of the edge connecting them. A snapshot of the DT is shown in Figure 1. We see that a target word ‘owl’ is connected with its co-hyponyms, ‘crow’ and ‘vulture’ via higher weighted edges, whereas the edge weights with its hypernyms like ‘animal’ are less. It may also happen that hypernyms of a target word are not even present in its neighborhood. For example, ‘creature’ is not present in the neighborhood of ‘owl’ but it is connected with ‘crow’ via less weighted edge. As per the DT network structure, distributionally similar words are present in a close proximity with similar neighborhood.

According to the literature dealing with lexical relation detection, words having co-hyponymy relation are distributionally more similar than the words having hypernymy or meronymy relation or any random pair of words. This is well captured by the DT. In a recent work, Jana and Goyal (2018b) used network features extracted from the DT to detect co-hyponyms. In our approach, we attempt to use embeddings obtained through a network representation learning method such as node2vec (Grover and Leskovec, 2016) when applied over the DT network. By choosing a flexible notion of a neighborhood and applying a biased random walk procedure, which efficiently explores diverse neighborhoods, node2vec learn representations for each node that organize nodes based on their network roles and/or

communities. We use the default setup of node2vec; having walk-length 80, walks per node 10, window size 10 and dimension of vector 128.

In order to do a qualitative analysis of the obtained vectors, we plot some sample words using t-SNE (Maaten and Hinton, 2008) in Figure 2. We observe that the relative distance between the co-hyponymy pairs is much smaller than those having hypernymy relations or meronymy relations for the DT embeddings. For instance, the co-hyponyms of ‘owl’ like ‘crow’, ‘vulture’, ‘sparrow’ are close to each other whereas hypernyms of ‘owl’ like ‘animal’, ‘vertebrate’, ‘creature’, as well as meronyms of ‘owl’ like ‘claw’, ‘feather’, are at distant positions.

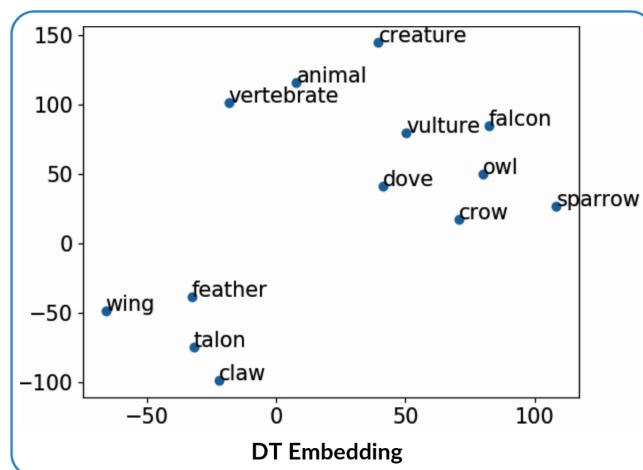


Figure 2: t-Distributed Stochastic Neighbor (t-SNE) (Maaten and Hinton, 2008) plot of DT embedding obtained using node2vec.

We aim to build a classifier that given a word pair, is able to detect whether or not they hold a co-hyponymy relation. Since we intend to explore the use of DT embeddings, we need to come up with specific ways to combine the embeddings of the word pair to be used as features for the classification. Following the literature (Weeds et al., 2014), we investigate four operations - vector difference (DIFF), vector concatenation (CC), vector pointwise addition (ADD) and vector pointwise multiplication (MUL). From our initial experiments, we find that CC and ADD prove to be the better combination methods overall. It is justified, as DIFF and MUL operations are somewhat intersective whereas both CC and ADD effectively come up with the union of the features in different ways and classifier fed with both shared and non-shared features has access to more information leading to better accuracy. We only report the performances for CC and ADD for Support Vector Machine (SVM) and Random Forest (RF) classifiers.

3. Experimental Results and Analysis

We perform experiments using three benchmark datasets for co-hyponymy detection (Weeds et al., 2014; Santus et al., 2016; Jana and Goyal, 2018b). For each of these, we follow the same experimental setup as discussed by the authors and compare our method with the method proposed by the author as well as the state-of-the-art models

by Jana and Goyal (2018b). We perform the analysis of three datasets to investigate the extent of overlap present in these publicly available benchmark datasets and find out that 45.7% word pairs of dataset prepared by Weeds et al. (2014) are present in dataset ROOT9 prepared by Santus et al. (2016). This intersection set comprises 27.8% of the ROOT9 dataset. Similarly 36.7% word pairs of dataset prepared by Weeds et al. (2014) are present in the whole dataset prepared by Jana and Goyal (2018b). This intersection set comprises 44.9% of the dataset prepared by Jana and Goyal (2018b).

Baseline Model	Description
svmDIFF	A linear SVM trained on the vector difference
svmMULT	A linear SVM trained on the pointwise product vector
svmADD	A linear SVM trained on the vector sum
svmCAT	A linear SVM trained on the vector concatenation
svmSING	A linear SVM trained on the vector of the second word in the given word pair
knnDIFF	k nearest neighbours (knn) trained on the vector difference
cosineP	The relation between word pair holds if the cosine similarity of the word vectors is greater than some threshold p
linP	The relation between word pair holds if the lin similarity (Lin, 1998) of the word vectors is greater than some threshold p

Table 1: Descriptions of the baseline models as described in (Weeds et al., 2014)

Model	Accuracy
svmDIFF	0.62
svmMULT	0.39
svmADD	0.41
svmCAT	0.40
svmSING	0.40
knnDIFF	0.58
cosineP	0.79
linP	0.78

Table 2: Accuracy scores on a ten-fold cross validation for *cohyponym*_{BLESS} dataset of all the baseline models described in (Weeds et al., 2014)

3.1. Experiment-1 (Weeds et al., 2014)

Weeds et al. (2014) prepared *cohyponym*_{BLESS} dataset from the BLESS dataset (Baroni and Lenci, 2011). *cohyponym*_{BLESS} contains 5,835 labeled pair of nouns; divided evenly into pairs having co-hyponymy relations and others (having hypernymy, meronymy relations along with random word pairs). In their work, Weeds et al. (2014) represent each word as positive pointwise mutual information (PPMI) based feature vector and propose

	Model	Accuracy
(Weeds et al., 2014)	svmDIFF	0.62
	cosineP	0.79
(Jana and Goyal, 2018b)	svmSS	0.84
Our models	SVM_CC	0.84
	SVM_ADD	0.9
	RF_CC	0.97
	RF_ADD	0.95

Table 3: Accuracy scores on a ten-fold cross validation for *cohyponym*_{BLESS} dataset of our models along with the top two baseline models (one supervised, one semi-supervised) described in (Weeds et al., 2014) and models described in (Jana and Goyal, 2018b)

Method	Co-Hyp vs Random	Co-Hyp vs Hyper
(Santus et al., 2016)	97.8	95.7
(Jana and Goyal, 2018b)	99.0	87.0
SVM_CC	96.5	91.4
SVM_ADD	93.5	97.6
RF_CC	99.0	98.6
RF_ADD	97.03	99.0

Table 4: Percentage F1 scores on a ten-fold cross validation of our models along with the best models described in (Santus et al., 2016) and (Jana and Goyal, 2018b) for ROOT9 dataset

a set of baseline methodologies, the descriptions of which are presented in Table 1.

Following the same experimental setup, we report the accuracy measure for ten-fold cross validation and compare our models with the baselines in proposed by Weeds et al. (2014). Table 2 represents the performance of all the baseline models proposed by Weeds et al. (2014). In Table 3 we show the performance of the best supervised model (svmDIFF) and the best semi-supervised model (cosineP) proposed by Weeds et al. (2014) along with our models. Here, the best model proposed by Jana and Goyal (2018b) uses SVM classifier which is fed with structural similarity of the words in the given word pair from the distributional thesaurus network. We see that all the 4 proposed methods perform at par or better than the baselines, and using RF_CC gives a 15.4% improvement over the best results reported.

3.2. Experiment-2 (Santus et al., 2016)

In the second experiment, we use **ROOT9** dataset prepared by Santus et al. (2016), containing 9,600 labeled pairs extracted from three datasets: EVALution (Santus et al., 2015), Lenci/Benotto (Benotto, 2015) and BLESS (Baroni and Lenci, 2011). There is an even distribution of the three classes (hypernyms, co-hyponyms and random) in the dataset. Following the same experimental setup as (Santus et al., 2016), we report percentage F1 scores on a ten-fold cross validation for binary classification of co-hyponyms vs random pairs, as well as co-hyponyms vs. hypernyms using both SVM and Random Forest classifiers. Table 4

represents the performance comparison of our models with the best state-of-the-art models reported in (Santus et al., 2016) and (Jana and Goyal, 2018b). Here, the best model proposed by Santus et al. (2016) uses Random Forest classifier which is fed with nine corpus based features like frequency of words, co-occurrence frequency etc., and the best model proposed by Jana and Goyal (2018b) use Random Forest classifier which is fed with five complex network features like structural similarity, shortest path etc. computed from the distributional thesaurus network. The results in Table 4 shows that, for the binary classification task of co-hyponymy vs random pairs, we achieve percentage F1 score of 99.0 with RF_CC which is at par with the state-of-the-art models. More importantly, both RF_CC and RF_ADD beat the baselines with significant margins for the classification task of co-hyponymy vs hypernymy pairs.

Model	Co-Hyp vs Random	Co-Hyp vs Mero	Co-Hyp vs Hyper
svmSS	0.96	0.86	0.73
rfALL	0.97	0.89	0.78
SVM_CC	0.9	0.89	0.854
SVM_ADD	0.943	0.89	0.869
RF_CC	0.97	0.978	0.98
RF_ADD	0.971	0.956	0.942

Table 5: Accuracy scores on a ten-fold cross validation of models (svmSS, rfALL) proposed by Jana and Goyal (2018b) and our models for the dataset prepared by Jana and Goyal (2018b).

3.3. Experiment-3 (Jana and Goyal, 2018b)

In the third experiment we use the dataset specifically build for co-hyponymy detection in one of the recent works by Jana and Goyal (2018b). This dataset is extracted from BLESS (Baroni and Lenci, 2011) and divided into three small datasets- **Co-Hypo vs Hyper**, **Co-Hypo vs Mero**, **Co-Hypo Vs Random**. Each of these datasets are balanced, containing 1,000 co-hyponymy pairs and 1,000 pairs for the other class. Following the same setup, we report accuracy scores for ten-fold cross validation for each of these three datasets of our models along with the best models (svmSS, rfALL) reported by Jana and Goyal (2018b) in Table 5. Jana and Goyal (2018b) use SVM classifier with structural similarity between words in a word pair as feature to obtain svmSS and use Random Forest classifier with five complex network measures computed from distributional thesaurus network as features to obtain rfALL. From the results presented in Table 5, RF_CC proves to be the best among our proposed models which performs at par with the baselines for **Co-Hypo vs Random** dataset. Interestingly, it beats the baselines comprehensively for **Co-Hypo vs Mero** and **Co-Hypo vs Hyper** datasets, providing improvements of 9.88% and 25.64%, respectively.

3.4. Error Analysis

We further analyze the cases for which our model produces wrong prediction. We point out some example word pairs such as ‘screw - screwdriver’, ‘gorilla - orangutan’ from *cohyponym*_{BLESS} dataset which our model wrongly flags as ‘false’. We observe a drastic difference in frequency between the words in these words pairs in the corpus from which the DT was constructed; for example ‘screw’ appears 592,857 times whereas ‘screwdriver’ has a frequency of 29,748; similarly ‘gorilla’ has a frequency of 40,212 whereas ‘orangutan’ has 3,567. In the DT network, edge weight depends on the overlap between top 1000 context features, and a drastic frequency difference might not capture this well. On the other hand, there are examples like ‘potato - peel’, ‘jacket - zipper’ which our model wrongly flags as ‘true’ co-hyponyms. We observe that the corpus does not contain many co-hyponyms of ‘peel’ or ‘zipper’, and thus their neighborhood in the DT network contains words like ‘ginger, lemon, onion, garlic’ and ‘pant, skirt, coat, jeans’ which are co-hyponyms of ‘potato’ and ‘jacket’, respectively. This leads to the false signal by the approach.

4. Conclusion

In this paper, we have investigated how the distributional thesaurus embeddings obtained using network representation learning can help improve the otherwise difficult task of discriminating co-hyponym pairs from hypernym, meronym and random pairs. By extensive experiments, we have shown that while the proposed models are at par with the baselines for detecting co-hyponyms vs. random pairs, they outperform the state-of-the-art models by a huge margin for the binary classification of co-hyponyms vs. hypernyms, as well as co-hyponyms vs. meronyms. It clearly shows that network representations can be very effectively utilized for a focused task like relation extraction. All the datasets, DT embeddings and codes (with instructions) used in our experiments are made publicly available¹.

The next immediate step is to try out DT embedding to build unsupervised model for co-hyponymy detection. In future, we plan to investigate some more sophisticated network representation learning techniques like path embedding, community embedding techniques to embed the path joining the given pair of words or the subgraph induced by the given pair of words etc. and apply it on distributional thesaurus network for robust detection of lexical relations. In this study, our focus has been distinguishing a horizontal relation, co-hyponymy, from parent-child relations like hypernymy and meronymy. However, the investigation on discriminating two analogous sibling relations, co-hyponymy and co-meronymy using the proposed method would be one of the interesting future direction. Finally, our broad objective is to build a general supervised and unsupervised framework based on complex network theory to detect different lexical relations from a given a corpus with high accuracy.

¹<https://tinyurl.com/u55np6o>

5. Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) under the project “Joining Ontologies and Semantics Induced from Text” (JOIN-T 2, BI 1544/4-2, PO 1900/1-2).

6. Bibliographical References

- Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.
- Berland, M. and Charniak, E. (1999). Finding parts in very large corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 57–64. Association for Computational Linguistics.
- Ferret, O. (2017). Turning distributional thesauri into word vectors for synonym extraction and expansion. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 273–283.
- Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (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.
- Geffet, M. and Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 107–114. Association for Computational Linguistics.
- Girju, R., Badulescu, A., and Moldovan, D. (2006). Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.
- Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.
- Jana, A. and Goyal, P. (2018a). Can network embedding of distributional thesaurus be combined with word vectors for better representation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 463–473.
- Jana, A. and Goyal, P. (2018b). Network features based co-hyponymy detection. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).
- Kiela, D., Rimell, L., Vulic, I., and Clark, S. (2015). Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*, pages 119–124. ACL.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 768–774. Association for Computational Linguistics.
- Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.
- Morlane-Hondère, F. (2015). What can distributional semantic models tell us about part-of relations? In *Net-WordS*, pages 46–50.
- Nguyen, K. A., Köper, M., Schulte im Walde, S., and Vu, N. T. (2017). Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243, Copenhagen, Denmark, September. Association for Computational Linguistics.
- Pantel, P. and Pennacchiotti, M. (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics.
- Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.
- Ribeiro, L. F., Saverese, P. H., and Figueiredo, D. R. (2017). struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 385–394. ACM.
- Riedl, M. and Biemann, C. (2013). Scaling to large³ data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 884–890.
- Roller, S. and Erk, K. (2016). Relations such as hypernymy: Identifying and exploiting hearst patterns in distributional vectors for lexical entailment. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2163–2172, Austin, Texas, November. Association for Computational Linguistics.
- Roller, S., Erk, K., and Boleda, G. (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.
- Santus, E., Lenci, A., Lu, Q., and Im Walde, S. S. (2014). Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42.
- Santus, E., Yung, F., Lenci, A., and Huang, C.-R. (2015). Evaluation 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 (LDL-2015)*, pages 64–69.
- Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., and Huang, C.-R. (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)*, Paris, France, may. European Language Resources Association (ELRA).
- Shwartz, V., Santus, E., and Schlechtweg, D. (2017). Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 65–75, Valencia, Spain, April. Association for Computational Linguistics.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.
- Weeds, J., Clarke, D., Reffin, J., Weir, D., and Keller, B. (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 City University and Association for Computational Linguistics.
- Yu, Z., Wang, H., Lin, X., and Wang, M. (2015). Learning term embeddings for hypernymy identification. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

7. Language Resource References

- Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.
- Benotto, G. (2015). *Distributional Models for Semantic Relations: A Study on Hyponymy and Antonymy*.
- Goldberg, Y. and Orwant, J. (2013). A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, volume 1, pages 241–247.
- Jana, A. and Goyal, P. (2018). Network features based co-hyponymy detection. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).
- Santus, E., Lenci, A., Chiu, T.-S., Lu, Q., and Huang, C.-R. (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)*, Paris, France, may. European Language Resources Association (ELRA).
- Weeds, J., Clarke, D., Reffin, J., Weir, D., and Keller, B. (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 City University and Association for Computational Linguistics.