Undersampling Improves Hypernymy Prototypicality Learning

Koki Washio, Tsuneaki Kato
Department of Language and Information Sciences
The University of Tokyo
3-8-1, Komaba, Meguroku, Tokyo 153-8902 Japan
{kokiwashio@g.ecc, kato@boz.c}.u-tokyo.ac.jp

Abstract

This paper focuses on supervised hypernymy detection using distributional representations for unknown word pairs. Levy et al. (2015) demonstrated that supervised hypernymy detection suffers from overfitting hypernyms in training data. We show that the problem of overfitting on this task is caused by a characteristic of datasets, which stems from the inherent structure of the language resources used, hierarchical thesauri. The simple data preprocessing method proposed in this paper alleviates this problem. To be more precise, we demonstrate through experiments that the problem that hypernymy classifiers overfit hypernyms in training data comes from a skewed word frequency distribution brought by the quasi-tree structure of a thesaurus, which is a major resource of lexical semantic relation data, and propose a simple undersampling method based on word frequencies that can effectively alleviate overfitting and improve distributional prototypicality learning for unknown word pairs.

Keywords: lexical semantic relations, hypernymy detection, taxonomy induction

1. Introduction

Detecting hypernymy relations between unknown words contributes to Taxonomy Induction, which induces a taxonomy in a new domain (Panchenko et al., 2016).

Supervised distributional hypernymy detection represents each word pair (x, y) as combined distributional representations, and trains a classifier that discriminates based on whether the word pair has a relation. Frequently used methods for combining word representations include vector concatenation $\vec{x} \oplus \vec{y}$ (CONCAT) and difference $\vec{y} - \vec{x}$ (DIFF) (Baroni et al., 2012; Fu et al., 2014; Roller et al., 2014; Weeds et al., 2014; Vylomova et al., 2016). The supervised methods have been reported to be better than the distributional unsupervised measures (Roller et al., 2014; Weeds et al., 2014).

However, Levy et al. (2015) demonstrated that supervised classifiers have some problems. Two major problems are as follows:

Problem 1 Classifiers do not learn relations in word pairs but only learn distributional prototypicality at best.

Problem 2 Classifiers overfit hypernyms, especially those in training data (*lexical memorization*).

Distributional prototypicality, if learned correctly, is still useful. Shwartz et al. (2016) integrated this information into their neural path-based model, which captures relations between two words, and improved the performance significantly. Roller and Erk (2016) demonstrated that the prototypicality learned by CONCAT captures Hearst patterns such as "fruits such as apples." Their method using the CONCAT model as a feature detector has high generalization performance. Thus, resolving Problem 2 is expected to improve the performance of the previous models.

In this paper, we investigate why classifiers overfit hypernyms in training data. We analyze this problem from the point of view of the skewed distribution of frequencies of hypernyms in training data, which stems from the inherent structure of language resources such as hierarchical thesauri. We show that an imbalance of word frequencies adversely affects classifiers, and we verify our analysis by experiments.

Moreover, we show that a simple undersampling method to balance frequencies of words in training data effectively alleviates the overfitting of hypernyms. Our experiment demonstrates that the undersampling method successfully improves the generalization performance for unknown word pairs.

2. Problems of Supervised Methods

Problem 1 is demonstrated by the tendency that supervised classifiers incorrectly assign hypernymy labels to switched pairs, which are mismatched instance-category pairs, *e.g.*, (apple, vehicle). Levy et al. (2015) provided a mathematical analysis on why linear classifiers cannot learn word relations. The forms of DIFF and CONCAT can be described as follows:

$$\begin{aligned} DIFF(x, y; \vec{\theta}) &= \vec{\theta} \cdot (\vec{y} - \vec{x}) \\ &= \vec{\theta} \cdot \vec{y} - \vec{\theta} \cdot \vec{x} \end{aligned} \quad (1)$$

$$\begin{aligned} CONCAT(x, y; \vec{\theta_1}, \vec{\theta_2}) &= (\vec{\theta_1} \oplus \vec{\theta_2}) \cdot (\vec{x} \oplus \vec{y}) \\ &= \vec{\theta_1} \cdot \vec{x} + \vec{\theta_2} \cdot \vec{y} \end{aligned} \quad (2)$$

where $\vec{\theta}$ and $(\vec{\theta_1}, \vec{\theta_2})$ are parameter vectors of DIFF and CONCAT, respectively. While the parameter vectors of DIFF and CONCAT can be interpreted as distributional prototypicality, these methods do not consider interactions between x and y. Thus, they cannot capture the relation between the word pair. Levy et al. (2015) also tried to use nonlinear kernel SVM, which can capture interactions between word vectors. However, the improvements were marginal.

Problem 2 is demonstrated by the fact that when the training data and test data have no lexical overlap (lexical split setting), classifiers perform extremely poorly. Levy et al. (2015) also showed that even if classifiers learn with only

Dataset		Mean	Median	Mean-Median	Max	Min
HyperLEX	hyper	2.37	1	1.37	62	1
	hypo	1.18	1	0.18	4	1
EVALution	hyper	2.73	1	1.73	71	1
	hypo	1.54	1	0.54	7	1
LEDS	hyper	3.41	1	2.41	60	1
	hypo	1.21	1	0.21	4	1

Table 1: Statistics of word frequencies in each position of hypernymy pairs of each dataset.

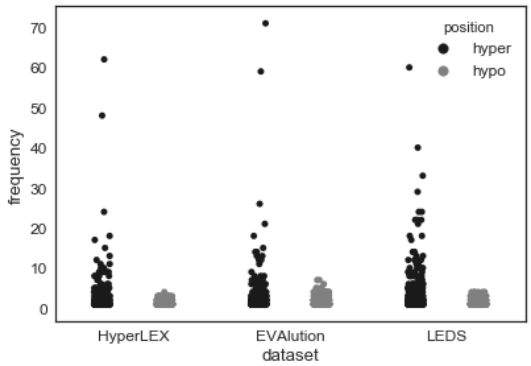

Figure 1: Strip plot of frequencies of hypernyms and hyponyms on each dataset.

$\vec{y}$ of word pair (x, y), their performance does not decrease so much in the lexical split setting. This indicates that classifiers ignore x's information. Problem 2 makes classifiers incapable of appropriately classifying words not included in the training data. This is a critical issue for a downstream task such as Taxonomy Induction.

While Problem 1 was provided with sufficient analysis by Levy et al. (2015), why do the classifiers overfit hypernyms in training data and ignore the information of hyponyms? This is the problem we address in this paper.

3. A Reason for Overfitting Hypernyms

We build a hypothesis that focuses on the distribution of hypernym frequencies[1] of training data to investigate what causes the overfitting of hypernyms.

Thesauri, major resources of word relation datasets, typically have a quasi-tree structure. One word/concept can have many hyponyms, but only one or a few hypernyms. If word pairs are extracted from thesauri, in training data, the same words that have general meanings appear naturally many times at the hypernym position of word tuples. As a result, the distribution of frequencies of a particular word being a hypernym of other words in training data becomes skewed in that the distribution is long-tailed or has many outliers. Figure 1 displays strip plots of the frequencies of hypernyms and hyponyms on three datasets: HyperLEX (Vulić, Ivan and Gerz, Daniela and Kiela, Douwe and Korhonen, Anna, 2016), EVAlution (Santus, Enrico and Yung, Frances and Lenci, Alessandro and Huang, Chu-Ren, 2015), and LEDS (Baroni, Marco and Bernardi, Raffaella and Do, Ngoc-Quynh and Shan, Chung-chieh, 2012). Table 1 displays the statistics of the word frequencies in each position of the hypernymy pairs of each dataset, where the difference of the mean and median of the hypernym position are larger than those of the hyponym position. These shows that the hypernym frequencies are largely skewed, while the hyponym frequencies are slightly skewed on all datasets. In these datasets, general and common hypernyms, such as *food*, *animal*, and *vehicle*, have significantly high frequencies.

How does this property affect DIFF and CONCAT? In training data, the number of types of hypernyms is small, and some types appear many times, while the number of types of hyponyms is large, and each type appears only a few times. Thus, words with a hypernym position have tendencies such as domain similarity in addition to the expected features of the prototypical hypernymy, while words with a hyponym position have few tendencies. This makes DIFF and CONCAT concentrate on repeated hypernym vectors $\vec{y}$ and ignore $-\vec{\theta} \cdot \vec{x}$ in equation (1) and $\vec{\theta}_1 \cdot \vec{x}$ in equation (2), as hyponyms share fewer features than do duplicating hypernyms. The biased supervised training shifts the parameter vectors to the features of hypernyms rather than to the true prototypicality, and results in overfitting hypernyms in the training data and ignoring hyponym information.

3.1. Experiments

To confirm our hypothesis, we conduct two experiments. First, we investigate how the skewed distribution of words affects the performance of the classifiers by adding extra pairs to the training data. Second, we examine the correlation between the hypernym frequencies and the mean inner products of the trained parameter vector (distributional prototypicality) and the feature vectors.

3.1.1. Setup

For distributional representation, we exploit the pretrained dependency word embeddings of Levy and Goldberg (2014). For datasets, we use HyperLEX, EVAlution, and LEDS. Only noun pairs of each dataset are used in our experiments. We remove samples containing words out of the vocabulary of the representations. We split each dataset into train/test subsets while keeping a roughly 75/25 ratio in random/lexical splitting.[2]

[1] In this paper, we use *hypernym/hyponym frequency* as the frequency of a particular word being a hypernym/hyponym of other words in word pair data.

[2] For HyperLEX, we use the standard train/test/dev splits that were provided in the dataset. In our experiment, the development

Split	HyperLEX	EVAlution	LEDS
random	607 (+1616)	880 (+2456)	328 (+2074)
lexical	586 (+1051)	547 (+799)	303 (+719)

Table 2: Numbers of added samples with frequent hypernyms. Values in parentheses show numbers of original training samples.

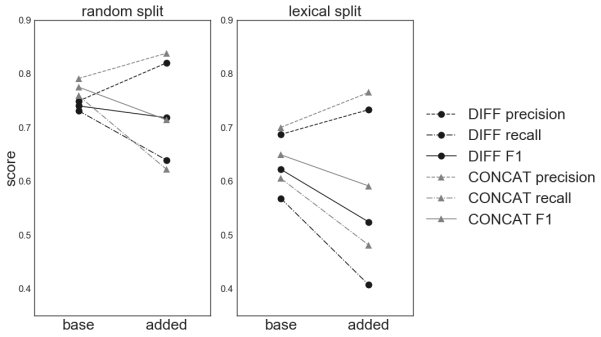

Figure 2: Plot of performance scores of DIFF and CONCAT classifiers in random and lexical splitting on HyperLEX.

We use logistic regression with L2 regularization for classifiers, exploiting scikit-learn[3] (Pedregosa et al., 2011) with the default hyperparameters, with the exception of the use of balanced class weights.

3.1.2. Skewed Distribution Influence

To investigate how skewed distributions of words affect the performance, we conduct the following experiment:
We extract the 10 most frequent hypernyms from the training data. Then, we extract direct hypernymy pairs with these frequent hypernyms from WordNet (Fellbaum, Christiane, 1998) and add the pairs that are not included in either the training or the test data to the training data.[4] This process makes the distribution of hypernym frequencies in training data more skewed. The numbers of the added samples are listed in Table 2. We evaluate the performance of classifiers in random and lexical splitting when adding new pairs for each splitting.
Figure 2 shows the results for HyperLEX. We obtained similar results for the other datasets. We can see that the more skewed the hypernym frequencies in the training data, the higher the precision and the lower the recall, dropping the F1 score as a result.[5] This tendency can be interpreted in that the classifiers focus only on frequent hypernyms in the training data and fail to correctly classify hypernymy pairs with infrequent hypernyms. This experiment demonstrates that skewed distributions of words give rise to overfitting.

3.1.3. Correlation Experiment

We train DIFF and CONCAT classifiers on each entire dataset without splitting, and examine the correlation between the hypernym frequencies and the mean of the inner products of the trained parameter vector and combined word representations on each hypernym frequency.
If the classifier goes through the ideal supervised learning, the obtained prototypical hypernymy, namely the parameter vector, should be irrelevant to frequencies of hypernyms in the training data. However, the correlations are significantly high at all settings ($\rho > 0.7$)[6]. This indicates that hypernym frequencies in the training data have a relationship to overfitting.
These two experiments demonstrate that a skewed distribution of hypernyms is a major factor in overfitting hypernyms. In addition, it leads to the ignoring of information about hyponyms whose distribution is not skewed.

4. Undersampling Method

Based on the analyses of Section 3, we propose an undersampling method to alleviate the overfitting of hypernyms and improve distributional prototypicality learning.
This method first calculates the third quartile of the hypernym frequencies in training data and removes from the training data hypernymy pairs including hypernyms that are more frequent than the third quartile. For each hypernym of those pairs, randomly chosen portions are brought back until the frequency of the hypernym matches the third quartile[7].
This is a simple method to correct the skew of the distribution of the frequencies of hypernyms in training data. We call this method *lexical undersampling* (LU), which is expected to alleviate overfitting hypernyms and improve the classifiers' performance on unknown word pairs.

4.1. Experiments and Results

We use the same datasets, word representations, and logistic regression model described in Section 3.1.1. The baselines are CONCAT and DIFF models with no data augmentation method.
Table 3 displays the results for each dataset. In almost all settings, LU lowers the precision but improves the recall, with the exception of CONCAT on the lexical split setting of EVAlution. This is the opposite trend as what was seen in Section 3.1.2. Thus, it seems that the overfitting of hypernyms is alleviated.
In almost all of the random split settings where classifiers benefit from lexical memorization, the baseline model outperforms +LU on F1 with the exception of DIFF on HyperLEX. This might be because LU makes it difficult for the models to take advantage of lexical memorization because of undersampling. These results indicate that LU is not beneficial to the random split because LU disturbs lexical memorization.
In all of the lexical split settings where the performance for unknown word pairs is evaluated, +LU outperforms the baselines on F1 for both CONCAT and DIFF. These results demonstrate that LU improves the learned distributional prototypicality and the generalization performance

set and the training set are merged, producing the new training set.

[3] http://scikit-learn.org/stable/

[4] In lexical split settings, we add only the pairs that do not contain the vocabulary of the test data.

[5] Only in the random splitting on LEDS does adding samples slightly lower the precision of DIFF ($0.780 \rightarrow 0.777$).

[6] The actual values in each dataset are listed in Table 4 along with the results of later proposals.

[7] If the third quartile has a decimal point, it is rounded off.

Dataset	Method	DIFF			CONCAT		
		precision	recall	F1	precision	recall	F1
Random split							
HyperLEX	baseline	**0.749**	0.731	0.740	**0.791**	0.759	**0.775**
	+LU	0.728	**0.795**	**0.760**	0.753	**0.795**	0.773
EVAlution	baseline	**0.507**	0.611	**0.554**	**0.528**	0.655	**0.585**
	+LU	0.441	**0.624**	0.517	0.477	**0.668**	0.556
LEDS	baseline	**0.780**	0.836	**0.807**	**0.766**	0.824	**0.794**
	+LU	0.765	**0.841**	0.802	0.756	**0.827**	0.790
Lexical split							
HyperLEX	baseline	**0.687**	0.568	0.622	**0.700**	0.605	0.649
	+LU	0.654	**0.630**	**0.642**	0.667	**0.741**	**0.702**
EVAlution	baseline	**0.424**	0.574	0.488	0.466	0.603	0.526
	+LU	0.410	**0.632**	**0.497**	**0.479**	**0.662**	**0.556**
LEDS	baseline	**0.782**	0.601	0.680	**0.821**	0.608	0.699
	+LU	0.763	**0.629**	**0.690**	0.769	**0.699**	**0.733**

Table 3: Performance for each model and splitting.

Method	baseline		+LU	
	DIFF	CONCAT	DIFF	CONCAT
HyperLEX	0.719	0.767	0.573	0.655
EVAlution	0.833	0.720	0.110	0.111
LEDS	0.744	0.710	0.347	-0.551

Table 4: Correlation between frequency of hypernyms in data and mean inner products.

	base	+LU
HyperLEX	0.488	**0.668**
EVAlution	0.471	**0.750**
LEDS	0.877	1.143

Table 5: Ratio of mean of squared parameters of hyponym to that of hypernym on CONCAT models.

for unknown words. These results indicate that LU is effective for unknown word pairs. Handling unknown words well is important to applications such as taxonomy induction. It is also possible to change the model depending on whether a pair in question includes a known word.

4.1.1. Diminished Correlation

In addition, we apply LU to the correlation experiments of Section 3.1.3. We use LU when learning the distributional prototypicality, and calculate the correlation with the original dataset. Table 4 shows that LU successfully diminishes the correlations and reduces the bias to frequent hypernyms. The negative correlation for CONCAT on LEDS might be because two-thirds of the negative samples of this dataset are switched pairs derived from the positive samples, which makes the frequent hypernyms negative signals when LU is applied.

4.1.2. Well-Balanced Weighted Features

In order to explore how the models weight words at the hypernym and hyponym positions, we investigate the ratio of the mean of the squared parameters of the hyponym position to that of the hypernym position of the CONCAT classifiers on each dataset. If the ratio is close to 1, the model equally weights the word at each position. On the other hand, if the ratio is close to 0, the model weights only the hypernym position word, which indicates that the model ignores the hyponym position word. Table 5 displays the ratio for each dataset. We can see that LU makes the classifiers focus more on the words of hyponym positions in all datasets. Applying LU successfully obtains a ratio close to 1 for HyperLEX and EVAlution, although it reverses the ratio for LEDS for the same reason as the negative correlation in Table 3. This means that the classifiers trained with LU on these datasets assign weights more equally to the hypernym vector and the hyponym vector. These results indicate that LU alleviates overfitting hypernyms and ignoring hyponyms.

	Roller and Erk (2016)	+LU
HyperLEX	0.667	**0.712**
EVAlution	0.538	**0.573**
LEDS	0.772	**0.801**

Table 6: F1 score of Roller and Erk (2016) in lexical split setting.

4.1.3. Contributing to Sophisticated Methods

Finally, LU contributes to the generalization performance of sophisticated methods using a distributional prototypicality such as that seen in Roller and Erk (2016) by providing valid components.

The model of Roller and Erk (2016) works through an iterative procedure similar to Principal Component Analysis, in which CONCAT is trained as a feature detector capturing a distributional prototypicality, and then this information is removed from the CONCAT vectors, resulting in a vector rejection. Training is repeated using the obtained vector rejection. With the CONCAT's parameters as a feature detector of distributional prototypicality, each process produces meta-features including the similarity of two words, hypernymy prototypicality, and distributional inclusion. After n

times of feature extracting, the final classifier is trained with these meta-features.

We apply LU to the feature detection step of their model and examine the F1 in the lexical split setting[8]. Table 6 shows that LU significantly improves the performance in the lexical split setting. This result demonstrates that the methods exploiting distributional prototypicality benefit from our undersampling method.

5. Conclusion

We investigated why classifiers overfit hypernyms in supervised distributional hypernymy detection. We showed that the skewed distribution of hypernym frequencies of training data makes classifiers overfit hypernyms and ignore hyponym information. This problem exemplifies the complex relationship between a task and its datasets.

Moreover, we proposed a simple undersampling method, lexical undersampling, to balance the hypernym frequencies in the training data. We demonstrated that this method successfully alleviates the overfit, and improves the distributional prototypicality learned by the classifiers and their generalization performance for unknown word pairs.

6. Acknowledgment

This work was supported by JSPS KAKENHI Grant numbers JP17H01831, JP15K12873.

7. Bibliographical References

Baroni, M., Bernardi, R., Do, N.-Q., and Shan, C.-c. (2012). Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32, Avignon, France.

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, Baltimore, Maryland.

Levy, O. and Goldberg, Y. (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (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.

Panchenko, A., Faralli, S., Ruppert, E., Remus, S., Naets, H., Fairon, C., Ponzetto, S. P., and Biemann, C. (2016). 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, California, June. Association for Computational Linguistics.

[8]We set the number of iterations of the feature detection at $n = 4$, following Roller and Erk (2016).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

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, Dublin, Ireland.

Shwartz, V., Goldberg, Y., and Dagan, I. (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.

Vylomova, E., Rimell, L., Cohn, T., and Baldwin, T. (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.

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, Ireland.

8. Language Resource References

Baroni, Marco and Bernardi, Raffaella and Do, Ngoc-Quynh and Shan, Chung-chieh. (2012). *Entailment above the word level in distributional semantics*. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics.

Fellbaum, Christiane. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, ISLRN 379-473-059-273-1.

Santus, Enrico and Yung, Frances and Lenci, Alessandro and Huang, Chu-Ren. (2015). *EVALution 1.0: an Evolving Semantic Dataset for Training and Evaluation of Distributional Semantic Models*. Proceedings of the 4th Workshop on Linked Data in Linguistics, Beijing.

Vulić, Ivan and Gerz, Daniela and Kiela, Douwe and Korhonen, Anna. (2016). *HyperLex: A Large-Scale Evaluation of Graded Lexical Entailment*. Language Technology Lab, University of Cambridge.