# Lexical Entailment with Hierarchy Representations by Deep Metric Learning

**Naomi Sato**[1] **Masaru Isonuma**[1] **Kimitaka Asatani**[1]
**Shoya Ishizuka**[2] **Aori Shimizu**[2] **Ichiro Sakata**[1]
[1] The University of Tokyo [2] Daikin Industries Ltd.
{nsato, isonuma, asatani, isakata}@ipr-ctr.t.u-tokyo.ac.jp
{shouya.ishizuka, aori.mito}@daikin.co.jp

## Abstract

In this paper, we introduce a novel method for lexical entailment tasks, which detects a hyponym-hypernym relation among words. Existing lexical entailment studies are lacking in generalization performance, as they cannot be applied to words that are not included in the training dataset. Moreover, existing work evaluates the performance by using the dataset that contains words used for training. This study proposes a method that learns a mapping from word embeddings to the hierarchical embeddings in order to predict the hypernymy relations of any input words. To validate the generalization performance, we conduct experiments using a train dataset that does not overlap with the evaluation dataset. As a result, our method achieved state-of-the-art performance and showed robustness for unknown words.

## 1 Introduction

Lexical entailment (LE) is a task to predict hypernym-hyponym relationships between two words, such as "A *swan* is a *bird*." and organize terms in a hierarchical order. LE can be used not only for the construction of thesaurus (Camacho-Collados, 2017; Yu et al., 2020), but also for semantic disambiguation (Martins et al., 2019) and visualization (Tanaka et al., 2018) of the substantial amount of information extracted from texts.

Methods utilizing hierarchical word embeddings are the recent mainstream for LE, and they can be classified into two categories. One is based on the distributive inclusion hypothesis (DIH), which assumes that if the specific word like "swan" is semantically entailed by the more general word like "bird", then the context in which "swan" occurs is relatively less frequent than and is included in the context in which "bird" occurs (Geffet and Dagan, 2005). Based on DIH, Vulíc and Mrkšíc (2018) developed LEAR, a method to post-process the pre-trained distributed representations to obtain embeddings emphasizing hypernymy relations. This type of embedding in Euclidian space is easy to apply in multiple ways(Iwamoto et al., 2021), such as visualization of lexicons and cross-lingual word translation(Vulić et al., 2019). Embedding hierarchical structures in hyperbolic space is the other category. Hyperbolic space's characteristic of exponentially increasing volume at points far from the origin is well suited to tree structure with multiple child nodes. Thus research on embedding in hyperbolic spaces has been attracting attention in the field of machine learning (Ganea et al., 2018) and also in LE (Nickel and Kiela, 2017).

Another stream is pattern-based approach that examines appearance patterns of hypernymies in large corpus. They began with Hearst patterns(Hearst, 1992; Roller et al., 2018), and in recent years, a variety of methods have been proposed, including using the Hierarchical Attention Network (Yu et al., 2020) or combining patterns with word embedding operations(Akhmouch et al., 2021).

One of the major drawbacks of these methods is that they are not able to predict hypernymies between unknown words, which are not included in the train dataset and/or corpus. Existing studies define the vocabulary set in advance and directly learn the distributed representation of each word as a parameter. Furthermore, this situation forces the performance evaluation to be laden with leakage. The evaluation datasets generally used in LE tasks consist of words contained in a train dataset, such as WordNet (Miller, 1994). This makes it impossible to evaluate generalization performance for unknown words properly, which is an undesirable situation in machine learning. Considering the real-world application of LE, the ability to locate unknown words that correspond to brand-new technologies and concepts at appropriate coordinates is critical. Only a few researches have coped with this problem, such as using GANs imitating the LEAR embeddings(Kamath et al., 2019), learning



*Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3517 - 3522

word-relation vectors by simulating the hypernymy generation process(Wang and He, 2020), and simply conducting unsupervised learning of GloVE on a hyperbolic space(Ifrea et al., 2019). However, the accuracy of the first study is limited by the performance of LEAR, and the rest have problems with embedding applications.

In this paper, we propose a method to obtain a hierarchical embedded representation that can be applied to unknown words, while maintaining the distinctiveness and usability. In order to acquire generalization performance for unknown words, our method is designed to learn the mapping from the pre-trained word vectors to the hierarchical embeddings, instead of hierarchical embeddings themselves. In addition, we aim to obtain representations more discriminative for hypernymy relations by using ranked list loss (Wang et al., 2019), a deep metric learning method, for learning this mapping function.

To evaluate the quality of our embeddings, we conducted an evaluation with standard word-level LE task datasets, such as BLESS (Baroni and Lenci, 2011). In order to evaluate the generalization performance for unknown words, training data were created with no word overlap with the evaluation dataset. Experimental results demonstrate that our method achieved state-of-the-art prediction performance and is robust to unknown words. We also confirm that the embedding is sufficiently expressive even with a low dimensionality of 5.

## 2 Methodology

Figure 1 illustrates the method for acquiring embeddings of hierarchical structures proposed in this paper. The category of a word is represented by the angle of the word embedding, and the concreteness is represented by its norm. Synonym pairs and hyponym-hypernym pairs with similar meanings are close in terms of the angle, and hyponyms have larger norms than hypernyms. The details of our method is explained as follows.

**Train Dataset** Let $\mathcal{B}_A = \{(w_l^1, w_r^1), ..., \}$ be a set of synonym pairs and $\mathcal{B}_L$ be a set of hyponym-hypernym pairs, which are used as train datasets. As described in a previous study (Vulíc and Mrkšíc, 2018), every pair is extracted from the thesaurus and divided into subsets consisting of $K$ pairs for mini-batch training. Particularly in $\mathcal{B}_L$, we always assign hyponym as $w_l$, while specifying a hypernym as $w_r$.

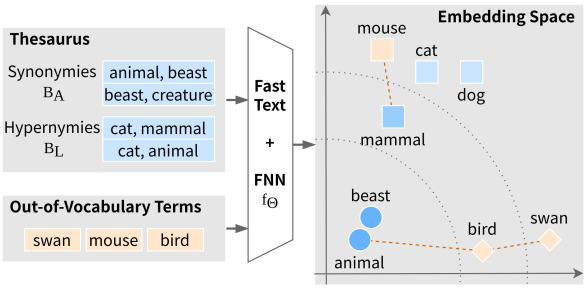


Figure 1: Overview of our proposed method.

**Encoding Hierarchy** In the next step, initial word embeddings are acquired with fastText (Bojanowski et al., 2017). As word embeddings are calculated by summarizing the vectors of its subwords, fastText provides two functions: estimating the representation for unknown words and retraining the model for additional words.

Then fastText representaions are transformed into a hierarchy embedding. Let $f_\Theta : \mathbb{R}^{d_{ft}} \rightarrow \mathbb{R}^{d_{he}}$ be the mapping from the fastText representation $\boldsymbol{v}$ in $d_{ft}$ dimension to the hierarchical embedded $\boldsymbol{u}$ in $d_{he}$ dimension. $f_\Theta$ is represented as a Fully-connected Neural Network (FNN) with an identity function as the activation function in the last layer and ReLU functions in the others.

**Loss Function** Here we enter into the details of the loss function used to train FNN. In order to learn a hierarchical embedded representation, the angle and norm are learned separately. Our loss function consists of two terms, $L_{angle}$ and $L_{norm}$ for the angle and norm, respectively.

For $L_{angle}$, we use ranked list loss (**RLL**; Wang et al., 2019), a deep metric learning method used in image classification. The concept of RLL is presented in Figure 2. Positive pairs (query-positive) are restricted close within a positive boundary, while negative pairs (query-negative) are kept farther away than a negative boundary. In this study, we use synonymies and hypernym/hyponym as positive examples, and all other pairs in the mini-batch are used as negative examples. The loss functions for positive and negative pairs are respectively defined as:

$$L_p(\boldsymbol{u}_l, \boldsymbol{u}_r) = [dcos(\boldsymbol{u}_l, \boldsymbol{u}_r) - (\alpha - m)]_+ \quad (1)$$

$$L_n(\boldsymbol{u}, \boldsymbol{t}) = [\alpha - dcos(\boldsymbol{u}, \boldsymbol{t})]_+ \quad (2)$$

Here, $dcos$ represents the cosine distance (i.e. $1 - cosine\_similarity$). It is employed to learn the angles of hierarchical embeddings efficiently. Then,

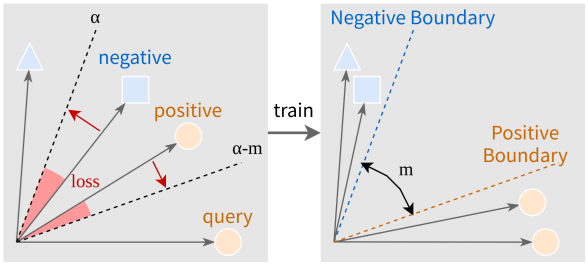


Figure 2: Concept of ranked list loss.

the loss for the entire mini-batch is defined as

$$L_{angle}(\mathcal{B}) = \sum_{(\boldsymbol{u}_l^i, \boldsymbol{u}_r^i) \in \mathcal{B}} \Bigg( 2 \cdot L_p\left(\boldsymbol{u}_l^i, \boldsymbol{u}_r^i\right) + \underset{\boldsymbol{t}^k \in \mathcal{B}, k \neq i}{mean} \left( L_n(\boldsymbol{u}_l^i, \boldsymbol{t}^k) + L_n(\boldsymbol{u}_r^i, \boldsymbol{t}^k) \right) \Bigg) \quad (3)$$

With regard to $L_{norm}$, we first quantify the difference between two words in terms of semantic hierarchy. Following Vulíc and Mrkšíc (2018), the hierarchical difference is defined as

$$D(\boldsymbol{u}_l, \boldsymbol{u}_r) = \frac{\|\boldsymbol{u}_l\| - \|\boldsymbol{u}_r\|}{\|\boldsymbol{u}_l\| + \|\boldsymbol{u}_r\|} \quad (4)$$

where $\boldsymbol{u}_l$ and $\boldsymbol{u}_r$ are embeddings of hyponym and hypernym, respectively.

$L_{norm}$ is defined to optimise the distance between this semantic hierarchy for all hypernymy pairs as follows:

$$L_{norm}(\mathcal{B}) = \sum_{(\boldsymbol{u}_l^i, \boldsymbol{u}_r^i) \in \mathcal{B}} -D(\boldsymbol{u}_l^i, \boldsymbol{u}_r^i) \quad (5)$$

In total, the loss function is summarized as:

$$L(\mathcal{B}_A, \mathcal{B}_L) = L_{angle}(\mathcal{B}_A \cup \mathcal{B}_L) + L_{norm}(\mathcal{B}_L) \quad (6)$$

**Lexical Entailment** Finally, we present methods for predicting lexical entailment between words based on hierarchical embeddings. In the Detection task, which discriminates implicated word pairs from other word pairs, the modified $HyperScore$ shown in Equation 7 is used to make prediction.

$$HyperScore'(\boldsymbol{u}_l, \boldsymbol{u}_r) = cos(\boldsymbol{u}_l, \boldsymbol{u}_r) \cdot \ln\left(\frac{\|\boldsymbol{u}_l\|}{\|\boldsymbol{u}_r\|}\right) \quad (7)$$

This is a modification of the existing judgment index, $HyperScore$ (Nguyen et al., 2017), to fit the characteristics of embedded representations obtained by the proposed method.

For the Directionality task, which determines the directionality of hypernymy relations, we simply compare the magnitude of the norm of the embeddings.

For the Graded Entailment task, which quantifies the strength of entailment relationships, we define $GradeScore$ as a quantitative measure according to Vulíc and Mrkšíc (2018):.

$$GradeScore(\boldsymbol{u}_l, \boldsymbol{u}_r) = cos(\boldsymbol{u}_l, \boldsymbol{u}_r) + D(\boldsymbol{u}_l, \boldsymbol{u}_r) \quad (8)$$

where $cos(\boldsymbol{u}_l, \boldsymbol{u}_r)$ represents the cosine similarity for semantic proximity, and $D(\boldsymbol{u}_l, \boldsymbol{u}_r)$ denotes the hierarchical distance defined in Equation 4.

## 3 Experiment

### 3.1 Setup

**Dataset** This study utilized datasets commonly used for evaluation in Lexical Entailment researches. Specifically, we used two types of detection tasks (BLESS (Baroni and Lenci, 2011), WBLESS (Weeds et al., 2014)), two types of directionality tasks (DBLESS (Nguyen et al., 2017), BIBLESS (Kiela et al., 2015)), and one graded entailment task (HYPERLEX (Vulić et al., 2017)).

The training data was created based on the English WordNet 2020 (McCrae et al., 2020) for two setups: inclusive and exclusive. In the *inclusive* setup, all pairs from the thesaurus were used as is. In the *exclusive* one, we omitted pairs containing words that were duplicated with the evaluation dataset. Accordingly, the vocabulary in the evaluation dataset can be considered pseudo-new words that are not included in the training data. By comparing the two setups, we can evaluate the generalization performance of the proposed method.

**Implementation Details** As for the model we used, the number of layers $H$ in FNN and the embedding size $d_{ft}$ were respectively set to 2 and 50. For the loss function, $m$ was set at 0.6 following the previous study (Vulíc and Mrkšíc, 2018) and $\alpha = 0.7$ for negative examples. We applied L2-normalization to each FNN parameter with normalization factor $\lambda = 10^{-7}$. Lastly, we fixed the number of epochs to 20, the size of mini-batches to 1024, and the learning rate to 0.01 during training.

**Baseline Methods** As a baseline method, we chose LEAR (Vulíc and Mrkšíc, 2018). LEAR has higher prediction accuracy among existing methods, and like the proposed method in this paper,

| task | Detection | | Directionality | | Graded |
|---|---|---|---|---|---|
| | BLESS | WBLESS | DBLESS | BIBLESS | HYPERLEX |
| evaluation index | AP | AP | Acc. | Acc. | $\rho$ |
| Baseline - inclusive | 0.527 | 0.924 | 0.956 | 0.764 | 0.435 |
| Ours - inclusive | 0.690 | 0.973 | 0.990 | 0.867 | 0.493 |
| Ours - exclusive | 0.535 | 0.962 | 0.985 | 0.799 | 0.430 |

Table 1: Performance of the baseline method and the proposed method for each task. AP, Acc., and $\rho$ respectively denote average precision score, accuracy, and Spearman's rank correlation coefficient.

it can be trained using only a thesaurus and pre-trained word vectors. Nevertheless, as LEAR only acquires embeddings for words included in the training data, we only considered the results obtained under an inclusive condition.

### 3.2 Results

Table 1 shows the performance of the baseline method and the proposed method for each task. Our method outperformed the baseline method trained on the same data in almost all cases with overlap. Furthermore, for many metrics, the difference in performance between the inclusive and exclusive conditions is less than 5%. In addition, for some tasks, the exclusive setting of the proposed method outperforms the baseline method. Particularly in the Detection and Directionality tasks, our model showed high generalization performance, with an accuracy of 4% over the existing study (Kamath et al., 2019; Ifrea et al., 2019) in the exclusive setup.

## 4 Discussions

### 4.1 Effect of RLL

As demonstrated above, the proposed method is useful in extracting entailment relations. In this section, we demonstrate that RLL improved the performance. To accomplish this, we replace only the loss function in the proposed method with Triplet Loss (Wang et al., 2014) and N-pair Loss (Sohn, 2016). All verifications were conducted under the exclusive setup with BLESS and WBLESS tasks.

The results are presented in Table 2. RLL demonstrated the highest performance among all tasks, confirming its usefulness for hierarchical embedding. In particular, the BLESS task achieved an accuracy improvement of more than 25% in AP scores. This is due to the fact that BLESS task contains a lot of associated words and co-hyponyms as negative examples, and they are fairly difficult to distinguish from hypernymies. The Positive Boundary setting of RLL is thought to have brought hypernymy pairs closer together, and therefore resulted in a more discriminative embedding space.

| | BLESS | | WBLESS | |
|---|---|---|---|---|
| | Acc. | AP | Acc. | AP |
| RLL | **0.852** | **0.690** | **0.907** | **0.973** |
| N-pair | 0.757 | 0.344 | 0.861 | 0.936 |
| Triplet | 0.723 | 0.425 | 0.870 | 0.914 |

Table 2: Performance comparison over loss functions.

| $d_{ft}$ | 2 | 5 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| AP | 0.367 | 0.490 | 0.484 | 0.535 | 0.529 |
| Accuracy | 0.792 | 0.815 | 0.818 | 0.829 | 0.821 |

Table 3: BLESS performance over dimentionalities.

| $d_{ft}$ | 2 | 5 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| AP | 0.927 | 0.951 | 0.963 | 0.962 | 0.958 |
| Accuracy | 0.843 | 0.881 | 0.896 | 0.879 | 0.883 |

Table 4: WBLESS performance over dimentionalities.

### 4.2 Dimensionality Study

In addition, we examine the relationship between the dimensionality $d_{he}$ and the extraction performance for comparison with other hierarchical embedding techniques. The verification was conducted under the exclusive setup in both BLESS and WBLESS tasks. As presented in Table 3 and 4, it is confirmed that the difference in the accuracy between the 50D and 5D was only less than 5%. Additionally, we achieved $0.881$ accuracy with $d_{he} = 5$, exceeding $0.86$ accuracy of Poincaré Embedding with the same dimension. Based on this result, it's possible to obtain an embedded representation that captures the hierarchical structure with relatively low dimensions even in the Cartesian coordinate system, which is consistent with the findings of Iwamoto et al. (2021).

## 5 Conclusion

This study proposed a method for embedding arbitrary input by learning mappings, and evaluated its discriminative and generalization performance. The method combines subword representations, FNN-based mappings, and a deep metric learning technique RLL to obtain hierarchical representations. Experimental results indicated that RLL

improved accuracy and FNN-based mapping contributed to generalization performance. Moreover, from the subsequent discussion, we found that the obtained embedding representation was sufficiently expressive even at low dimensions. In conclusion, this study confirmed the effectiveness of deep metric learning in acquiring hierarchical embedding representations, succeeded in developing an effective method for extracting entailment relations for arbitrary words, and extended the possibility of application of lexical entailment[1].

## 6 Limitations

There are two limitations to this study. The first is that fastText has to return correct output for input words, and the second is that a certain amount of annotated hypernym-hyponym pairs must be available for training. However, since the fastText model can be additionally trained with a very small corpus, the first assumption is not considered to be a major limitation.

## References


Houssam Akhmouch, Gaël Dias, and Jose G. Moreno. 2021. Understanding feature focus in multitask settings for lexico-semantic relation identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2762–2772, Online. Association for Computational Linguistics.

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*, page 1–10.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, pages 135–146.

Jose Camacho-Collados. 2017. Why we have switched from building full-fledged taxonomies to simply detecting hypernymy relations. *arXiv preprint arXiv:1703.04178*.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. *43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 107–114.


[1]The code of our work will be available on `github.com` for the reproducibility.


Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Alexandru T. Ifrea, Gary Bécigneul, and Octavian Eugen Ganea. 2019. Poincaré glove: Hyperbolic word embeddings. In *7th International Conference on Learning Representations*.

Ran Iwamoto, Ryosuke Kohita, and Akifumi Wachi. 2021. Polar embedding. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 470–480.

Aishwarya Kamath, Jonas Pfeiffer, Edoardo Maria Ponti, Goran Glavaš, and Ivan Vulić. 2019. Specializing distributional vectors of all words for lexical entailment. In *Proceedings of the 4th Workshop on Representation Learning for NLP*, pages 72–83.

Douwe Kiela, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 119–124.

Pedro Henrique Martins, Zita Marinho, and André F T Martins. 2019. Joint learning of named entity recognition and entity linking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 190–196.

John Philip McCrae, Alexandre Rademaker, Ewa Rudnicka, and Francis Bond. 2020. English wordnet 2020: Improving and extending a wordnet for english using an open-source methodology. In *Proceedings of the LREC 2020 Workshop on Multimodal Wordnets*, pages 14–19.

George A Miller. 1994. Wordnet: A lexical database for english. In *Proceedings of the Workshop on Human Language Technology*, page 468.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical embeddings for hypernymy detection and directionality. *Conference on Empirical Methods in Natural Language Processing*, pages 233–243.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363. Association for Computational Linguistics.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*.

Hiroaki Tanaka, Yuko Nakashio, and Yuya Kajikawa. 2018. Evaluation method of patent scope based on semantic information of words and dependency structure of patent claims. In *Portland International Conference on Management of Engineering and Technology: Managing Technological Entrepreneurship: The Engine for Economic Growth*.

Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2019. Multilingual and cross-lingual graded lexical entailment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4963–4974, Florence, Italy. Association for Computational Linguistics.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. 2017. Hyperlex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, pages 781–835.

Ivan Vulíc and Nikola Mrkšíc. 2018. Specialising word vectors for lexical entailment. *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1134–1145.

Chengyu Wang and Xiaofeng He. 2020. BiRRE: Learning bidirectional residual relation embeddings for supervised hypernymy detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3630–3640, Online. Association for Computational Linguistics.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. Learning fine-grained image similarity with deep ranking. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M Robertson. 2019. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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.

Changlong Yu, Jialong Han, Peifeng Wang, Yangqiu Song, Hongming Zhang, Wilfred Ng, and Shuming Shi. 2020. When hearst is not enough: Improving hypernymy detection from corpus with distributional models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6208–6217.