A Mixture-of-Experts Model for Antonym-Synonym Discrimination

Zhipeng Xie
School of Computer Science
Fudan University, Shanghai, China
xiezp@fudan.edu.cn

Nan Zeng
School of Computer Science
Fudan University, Shanghai, China
19212010017@fudan.edu.cn

Abstract

Discrimination between antonyms and synonyms is an important and challenging NLP task. Antonyms and synonyms often share the same or similar contexts and thus are hard to make a distinction. This paper proposes two underlying hypotheses and employs the mixture-of-experts framework as a solution. It works on the basis of a divide-and-conquer strategy, where a number of localized experts focus on their own domains (or subspaces) to learn their specialties, and a gating mechanism determines the space partitioning and the expert mixture. Experimental results have shown that our method achieves the state-of-the-art performance on the task.

1 Introduction

Antonymy-synonymy discrimination (ASD) is a crucial problem in lexical semantics and plays a vital role in many NLP applications such as sentiment analysis, textual entailment and machine translation. Synonymy refers to semantically-similar words (having similar meanings), while antonymy indicates the oppositeness or contrastiveness of words (having opposite meanings). Although telling apart antonyms and synonyms looks simple on the surface, it actually poses a hard problem because of their interchangeable substitution.

A few research efforts have been devoted to computational solutions of ASD task, which comprises two mainstreams: *pattern-based* and *distributional* approaches. The underlying idea of pattern-based methods exists in that antonymous word pairs co-occur with each other in some antonymy-indicating lexico-syntactic patterns within a sentence (Roth and im Walde, 2014; Nguyen et al., 2017). In spite of their high precision, pattern-based methods suffer from limited recall owing to the sparsity of lexico-syntactic patterns and the lexical variations.

Distributional methods work on the basis of *distributional hypothesis* stating that "the words similar in meaning tend to occur in similar contexts" (Harris, 1954). Traditional distributional methods are based on discrete context vectors. Scheible et al. (2013) verified that using only the contexts of certain classes can help discriminate antonyms and synonyms. Santus et al. (2014) thought that synonyms are expected to have broader and more salient intersection of their top-K salient contexts than antonyms, and proposed an Average-Precision-based unsupervised measure.

With the advent of word embeddings as the continuous representations (Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Pennington et al., 2014), several neural methods have been proposed to elicit ASD-specific information from pretrained word embeddings in a supervised manner. Etcheverry and Wonsever (2019) used a siamese network to ensure the symmetric, reflexive and transitive properties of synonymy and a parasiamese network to model the antitransitivity of antonymy. Ali et al. (2019) projected word embeddings into the synonym and antonym subspaces respectively, and then trained a classifier on the features from these distilled subspaces, where the trans-transitivity of antonymy was taken into consideration.

This paper follows the distributional approach and studies the ASD problem on the basis of pretrained word embeddings. Two hypotheses underlie our method: (a) antonymous words tend to be similar on most semantic dimensions but be different on only a few salient dimensions; (b) the salient dimensions may vary significantly for different antonymies throughout the whole distributional semantic space. With respect to the hypothesis (b), we find that a tailored model of mixture-of-experts (MoE) (Jacobs et al., 1991) fits it well. The semantic space is divided into a number of subspaces, and each subspace has one specialized expert to elicit

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 558–564
August 1–6, 2021.

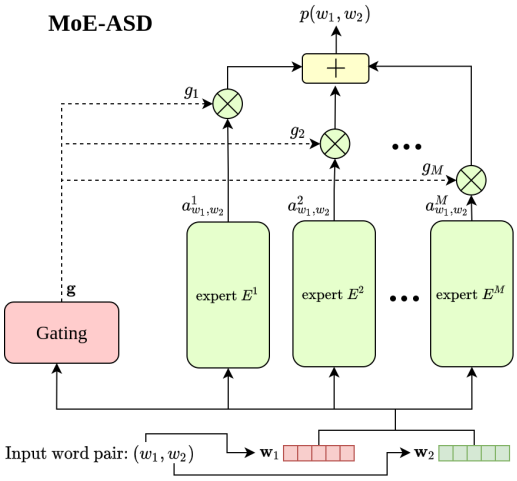

Figure 1: The architecture of MoE-ASD

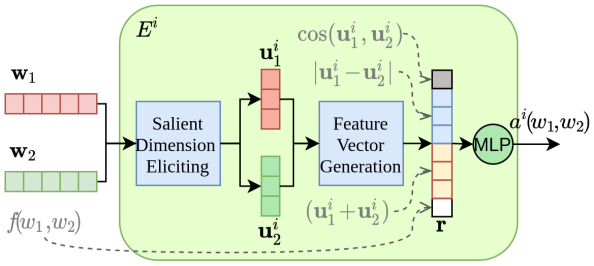

Figure 2: A localized expert

the salient dimensions and learn a discriminator for this subspace. As to the hypothesis (a), a similar opinion was also expressed by Cruse (1986) that antonymous words tend to have many common properties, but differ saliently along one dimension of meaning. In addition, our experimental results have shown that each expert requires only four salient dimensions to achieve the best performance.

Finally, we would like to point out the main difference of our method from the existing ones. Firstly, our MoE-ASD model adopts a divide-and-conquer strategy, where each subspace is in the charge of one relatively-simple localized expert that focuses on only a few salient dimensions; while existing methods rely on a global model which must grasp all the salient dimensions across all the subspaces. Secondly, our method simply enforces the symmetric property of synonymy and antonymy, but ignores the other algebraic properties such as the transitivity of synonymy and trans-transitivity of antonymy, because these algebraic properties do not always hold on the word level for the polysemy characteristic of words.

2 Method

This paper proposes a novel ASD method based on the mixture-of-experts framework (called **MoE-ASD**)[1]. Its architecture is illustrated in Figure 1. It solves the problem in a divide-and-conquer manner by dividing the problem space into a number of subspaces and each subspace is in the charge of a specialized expert. The expert focuses on the salient dimensions of the subspace and makes the decision for word pairs. A gating module is trained jointly with these experts. The details are as follows.

2.1 Localized Experts

All the experts are homogeneous, and they have the same network architecture but with different parameter values. Given a word pair (w_1, w_2) as input, each expert E^i computes its unnormalized probability $a^i(w_1, w_2)$ of being antonymy. As stated in Section 1, our method adopts the hypothesis that antonymous words tend to be similar on most semantic dimensions but be different on a few salient dimensions. Each expert has to first elicit the salient dimensions, and then makes a decision based on a feature vector constructed from them. Figure 2 illustrates how an expert works.

Let $\mathbf{w}_1$ and $\mathbf{w}_2$ denote the pre-trained word embeddings of words w_1 and w_2 respectively, whose dimensionality is d_e. Each expert E^i distills d_u salient dimensions from them by projecting them from $\mathbb{R}^{d_e}$ into $\mathbb{R}^{d_u}$:

$$\mathbf{u}^i_1 = \mathbf{w}_1 \cdot \mathbf{M}^i_u + \mathbf{b}^i_u \text{ and } \mathbf{u}^i_2 = \mathbf{w}_2 \cdot \mathbf{M}^i_u + \mathbf{b}^i_u \quad (1)$$

where $\mathbf{M}^i$ is a matrix of size $d_e \times d_u$ and $\mathbf{b}^i$ is a vector of length d_u. Next, a relational feature vector $\mathbf{r}$ is constructed by concatenating the sum $(\mathbf{u}^i_1 + \mathbf{u}^i_2)$, the absolute difference $|\mathbf{u}^i_1 - \mathbf{u}^i_2|$, the cosine similarity $\cos(\mathbf{u}^i_1, \mathbf{u}^i_2)$ and the prefix feature f_{w_1,w_2}:

$$\mathbf{r} = (\mathbf{u}^i_1 + \mathbf{u}^i_2) \oplus |\mathbf{u}^i_1 - \mathbf{u}^i_2| \oplus \cos(\mathbf{u}^i_1, \mathbf{u}^i_2) \oplus f_{w_1,w_2} \quad (2)$$

Here, f_{w_1,w_2} is the Negation-Prefix feature that denotes whether w_1 and w_2 differ only by one of the known negation prefixes: {*de*, *a*, *un*, *non*, *in*, *ir*, *anti*, *il*, *dis*, *counter*, *im*, *an*, *sub*, *ab*}, following Ali et al. (2019) and Rajana et al. (2017).

It is evident that the feature vector is symmetric with respect to the input word pair. This is,

[1] Our code and data are released at https://github.com/Zengnan1997/MoE-ASD

the word pairs (w_1, w_2) and (w_2, w_1) lead to the same feature vector. It is worth noting that the absolute difference is used instead of the difference, in order to preserve the symmetric properties of both synonymy and antonymy. We note that Roller et al. (2014) used the difference between two word vectors as useful features for detecting hypernymy which is asymmetric.

The relational feature vector $\mathbf{r}$ goes through an MLP to get the antonymy-score $a^i(w_1, w_2)$:

$$a^i(w_1, w_2) = (\mathbf{m}_o^i)^\top \cdot \text{ReLU}(\mathbf{r} \cdot \mathbf{M}_h^i + \mathbf{b}_h^i) + b_o^i \quad (3)$$

where the hidden layer has d_h units, $\mathbf{M}_h^i$ is a matrix of size $(2d_u + 2) \times d_h$, $\mathbf{b}_h^i$ and $\mathbf{m}_o^i$ are two vectors of length d_h, and b_o^i is the bias.

2.2 Gating Mechanism for Expert Mixture

Assume there are M localized experts in the MoE-ASD model. For an input word pair (w_1, w_2), we shall get M antonymy-scores $\mathbf{a} = [a^i(w_1, w_2)]_{1 \le i \le M}$, where each $a^i(w_1, w_2)$ is obtained from the expert E^i. Now, the problem is how to derive the final score for antonymy detection.

In our MoE-ASD model, the *final score* is a weighted average of the M scores from the localized experts:

$$s(w_1, w_2) = \mathbf{g}^\top \cdot \mathbf{a} \quad (4)$$

where $\mathbf{g}$ is located in the M-dimensional simplex, and denotes the proportional contributions of the experts to the final score. A gating mechanism is used to calculate $\mathbf{g}$ for each specific word pair (w_1, w_2), fulfilling a dynamic mixture of experts:

$$\mathbf{g} = \text{softmax}\left((\mathbf{w}_1 + \mathbf{w}_2)^\top \cdot \mathbf{M}_g\right) \quad (5)$$

where $\mathbf{M}_g \in \mathbb{R}^{d_e \times M}$ is the parameter matrix of the gating module. The i-th column of $\mathbf{M}_g$ can be thought of as the representative vector of the i-the expert, and the dot product between the sum of two word embeddings and the representative vector is the attention weight of the expert E^i. Softmax is then applied on the attention weights to get $\mathbf{g}$. It is evident that the gating module is also symmetric with respect to the input word pair. The symmetric properties of both the gating module and the local expert module endow our model with symmetry that make it distinct from the other state-of-the-arts such as **Parasiam** (Etcheverry and Wonsever, 2019) and **Distiller** (Ali et al., 2019).

Category	Train	Dev	Test	Total
Adjective	5562	398	1986	7946
Verb	2534	182	908	3624
Noun	2836	206	1020	4062

Table 1: Antonym/Synonym Dataset

2.3 Model Prediction and Loss Function

Given word pair (w_1, w_2), the probability of being antonymy is obtained by simply applying sigmoid function to the final score:

$$p(w_1, w_2) = \sigma(s(w_1, w_2)) \quad (6)$$

Let A denote the training set of N word pairs, $A = \{(w_1^{(n)}, w_2^{(n)})\}_{n=1}^N$, $t^{(n)}$ denote the gold label of the n-th word pair, and $p^{(n)}$ the predicted probability of being antonymy. Our model uses the cross-entropy loss function:

$$L = \frac{1}{N} \sum_{n=1}^{N} \left[t^{(n)} \log p^{(n)} + \left(1 - t^{(n)}\right) \log \left(1 - p^{(n)}\right) \right] \quad (7)$$

3 Evaluation

Dataset. We evaluate our method on the dataset (Nguyen et al., 2017) that was previously created from WordNet (Miller, 1995) and Wordnik[2]. The word pairs of antonyms and synonyms were grouped according to the word class (*Adjective*, *Noun* and *Verb*). The ratio of antonyms to synonyms in each group is 1:1. The statistics of the dataset are shown in Table 1. In order to make a fair comparison with previous algorithms, the dataset is splitted into training, validation and testing data the same as previous works.

Methods for Comparison: We make a comparison against the following ASD methods: (1) **Concat** - a baseline method that concatenates two word vectors and feeds it into an MLP with two hidden layers (with 400 and 200 hidden units respectively) and ReLU activation functions. (2) **AntSynNET** (Nguyen et al., 2017) is a pattern-based method that encodes the paths connecting the joint occurrences of candidate pairs using a LSTM; (3) **Parasiam** (Etcheverry and Wonsever, 2019) used a siamese network and a parasiamese network to ensure the algebraic properties of synonym and antonym, respectively. (4) **Distiller** (Ali et al., 2019) is a two-phase method that first distills

[2] http://www.wordnik.com

Method	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
Concat (Baseline)	0.596	0.751	0.651	0.596	0.750	0.656	0.688	0.745	0.708
AntSynNet (Nguyen et al., 2017)	0.750	0.798	0.773	0.717	0.826	0.768	0.807	0.827	0.817
Parasiam (Etcheverry and Wonsever, 2019)	0.855	0.857	0.856	0.864	**0.921**	0.891	0.837	0.859	0.848
Distiller (Ali et al., 2019)	0.854	**0.917**	0.884	0.871	0.912	0.891	0.823	0.866	0.844
MoE-ASD (Our method)	**0.878**	0.907	**0.892**	**0.895**	0.920	**0.908**	**0.841**	**0.900**	**0.869**

Table 2: Performance evaluation of our model and the baseline models (with vanilla word embeddings)

Method	Adjective			Verb			Noun		
	P	R	F1	P	R	F1	P	R	F1
AntSynNet (Nguyen et al., 2017)	0.763	0.807	0.784	0.743	0.815	0.777	0.816	0.898	0.855
Parasiam (Etcheverry and Wonsever, 2019)	0.874	**0.950**	0.910	0.837	**0.953**	0.891	0.847	0.939	0.891
Distiller (Ali et al., 2019)	0.912	0.944	0.928	0.899	0.944	0.921	0.905	0.918	0.911
MoE-ASD	**0.935**	0.941	**0.938**	**0.914**	0.944	**0.929**	**0.920**	**0.950**	**0.935**

Table 3: Performance evaluation with the dLCE embeddings

task-specific information and then trains a classifier based on distilled sub-spaces.

3.1 Experimental Settings

We use the 300-dimension FastText word embeddings (Bojanowski et al., 2017)[3]. The model is optimized with the Adam algorithm (Kingma and Ba, 2015). We run our algorithm 10 times and record the average Precision, Recall and F-scores. The number of salient dimensions (d_u) and the number of localized experts (M) are tuned on the validation data by grid search, with $M \in \{2^i\}_{1 \leq i \leq 8}$ and $d_u \in \{2^i\}_{1 \leq i \leq 8}$. The best configuration is $(d_u = 4, M = 256)$ for both *Noun* and *Verb*, while $(d_u = 4, M = 128)$ for *Adjective*.

3.2 Comparison with SOTA methods

Table 2 compares our method with the state-of-the-arts, which are restricted to pretrained vanilla word embeddings. Both the Parasiam method and our MoE-ASD method use FastText embeddings (Bojanowski et al., 2017), while *Distiller* uses Glove embeddings (Pennington et al., 2014).

It is observed that our model consistently outperforms the state-of-the-arts on all the three subtasks, which manifests the effectiveness of the mixture-of-experts model for ASD and validates the hypothesis (b) that the salient dimensions may vary significantly throughout the whole space.

We also find that the performance on *Noun* class is relatively low when compared with *Verb* and *Adjective* classes, which coincide with the observations obtained in (Scheible et al., 2013; Ali et al., 2019), possibly for the reason that polysemy phenomenon is more significant among nouns.

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-english/wiki-news-300d-1M.vec.zip

F1-score	Adjective	Verb	Noun
Our full method	0.892	0.908	0.869
–prefix feature	0.886	0.905	0.868
–cosine sim	0.883	0.905	0.856
–absolute diff	0.890	0.897	0.866
–sum	0.888	0.903	0.867

Table 4: Ablation analysis of the features

Besides vanilla word embeddings, existing ASD methods also used dLCE (Nguyen et al., 2016) embeddings, and often obtained better results. However, a large number of antonymies and synonymies have been used in the process of learning dLCE embeddings, which may lead to severe overfitting. In spite of this concern, we also test our method with dLCE embeddings on the dataset and find that it outperforms these competitors with dLCE and list the results in Table 3.

3.3 Ablation Analysis of Features

We also make an ablation analysis about the four kinds of features, by removing each of them from our model. It can be seen from Table 4 that all the features are making their own contributions to the ASD. Different parts of speech have different sensitivities to different features. Specifically, verb is most sensitive to "*absolute difference*", while both adjective and noun are most sensitive to "*cosine*". The reason behind the observations deserves further exploration.

newdataset	Model	Adjective			Verb			Noun		
		P	R	F1	P	R	F1	P	R	F1
FastText	Parasiam	0.694	**0.866**	0.769	0.642	**0.824**	0.719	0.740	**0.759**	0.748
	MoE-ASD	**0.808**	0.810	**0.809**	**0.830**	0.693	**0.753**	**0.846**	0.722	**0.776**
dLCE	Parasiam	0.768	**0.952**	0.850	0.769	**0.877**	0.819	0.843	**0.914**	0.876
	MoE-ASD	**0.877**	0.908	**0.892**	**0.860**	0.835	**0.847**	**0.912**	0.869	**0.890**

Table 5: Performance of our model and the baseline models on the lexical-split datasets

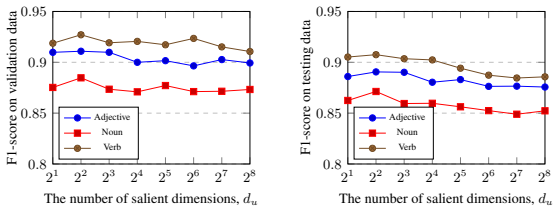

Figure 3: The effect on performance by varying the number of salient dimensions (fixing $M = 256$)

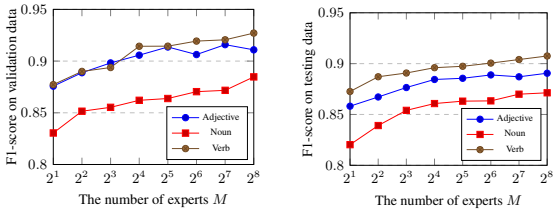

Figure 4: The effect on performance by varying the number of experts (fixing $d_u = 4$)

3.4 Hyperparameter Analysis

The number of salient dimensions (d_u) and the number of experts (M) are two prominent hyperparameters in our MOE-ASD model. By varying their values, we study their influence on the performance.

Firstly, by fixing $M = 256$, we vary d_u from 2^1 to 2^8 and plot the F1-scores on the validation data and the testing data in Figure 3. It is observed that all the three subtasks (*Adjective*, *Noun* and *Verb*) arrive at the best performance at $d_u = 4$ on both validation data and testing data. It validates our hypothesis (a) that antonymous words tend to be different on only a few salient dimensions.

Secondly, by fixing $d_u = 4$, we vary M from 2^1 to 2^8 and plot the F1-scores in Figure 4. Overall, the performance becomes better with the larger number of experts. We conjecture that marginal improvement will be obtained by increasing the number of experts further, but we do not make such experiments.

Category	Train	Dev	Test	Total
Adjective	4227	303	1498	6028
Verb	2034	146	712	2892
Noun	2667	191	954	3812

Table 6: The datasets after lexical split

3.5 Lexical Memorization

To eliminate the bias introduced by the lexical memorization problem (Levy et al., 2015), we perform lexical splits to obtain train and test datasets with zero lexical overlap. The statistics of the lexical-split datasets are listed in Table 6. Table 5 shows the results of our method and Parasiam on the lexical-split datasets by using FastText and dLCE pretrained word embeddings. It can be seen that our MoE-ASD model outperforms Parasiam on all three lexical-split datasets. However, significant decreases in the F1 scores are also observed.

4 Conclusions

This paper first presents two hypotheses for ASD task (i.e., antonymous words tend to be different on only a few salient dimensions that may vary significantly for different antonymies) and then motivates an ASD method based on mixture-of-experts. Finally, experimental results have manifested its effectiveness and validated the two underlying hypotheses. It is worth noting that our method is distinct from the other state-of-the-arts in two main aspects: (1) it works in a *divide-and-conquer* strategy by dividing the whole space into multiple subspaces and having one expert specialized for each subspace; (2) it is *inherently symmetric* with respect to the input word pair.

Acknowledgments

This work is supported by National Key Research and Development Program of China (No.2018YFB1005100) and National Natural Science Foundation of China (No.62076072). We are grateful to the anonymous reviewers for their valuable comments.

References

Muhammad Asif Ali, Yifang Sun, Xiaoling Zhou, Wei Wang, and Xiang Zhao. 2019. Antonym-synonym classification based on new sub-space embeddings. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, the 31st Innovative Applications of Artificial Intelligence Conference, the 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 6204–6211.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Benjamin Börschinger and Mark Johnson. 2011. A particle filter algorithm for Bayesian wordsegmentation. In *Proceedings of the Australasian Language Technology Association Workshop 2011*, pages 10–18, Canberra, Australia.

Dvaid A Cruse. 1986. *Lexical Semantics*. Cambridge University Press.

Mathías Etcheverry and Dina Wonsever. 2019. Unraveling antonym's word vectors through a siamese-like network. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 3297–3307.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for Abstract Meaning Representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–11, Berlin, Germany. Association for Computational Linguistics.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 1, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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. Association for Computational Linguistics.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2265–2273.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short Papers*.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Distinguishing antonyms and synonyms in a pattern-based neural network. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 1: Long Papers*, pages 76–85.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Sneha Rajana, Chris Callison-Burch, Marianna Apidianaki, and Vered Shwartz. 2017. Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, *SEM @ACM 2017, Vancouver, Canada, August 3-4, 2017*, pages 12–21. Association for Computational Linguistics.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1025–1036. ACL.

Michael Roth and Sabine Schulte im Walde. 2014. Combining word patterns and discourse markers for paradigmatic relation classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 524–530. The Association for Computer Linguistics.

Enrico Santus, Qin Lu, Alessandro Lenci, and Chu-Ren Huang. 2014. Taking antonymy mask off in vector space. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, pages 135–144.

Silke Scheible, Sabine Schulte im Walde, and Sylvia Springorum. 2013. Uncovering distributional differences between synonyms and antonyms in a word space model. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 489–497.