Modern Management based on Big Data II and Machine Learning and Intelligent Systems III
A.J. Tallón-Ballesteros (Ed.)
© 2021 The authors and IOS Press.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA210282

Intuitive Contrasting Map for Antonym Embeddings

Igor SAMENKO [a], Alexey TIKHONOV [b] and Ivan P. YAMSHCHIKOV [c,1]

[a] *Institute of Computational Technologies, Russian Academy of Sciences, Novosibirsk, Russia*

[b] *Yandex, Berlin, Germany*

[c] *LEYA Lab, Yandex and Higher School of Economics in St. Petersburg, Russia*

Abstract. This paper shows that modern word embeddings contain information that distinguishes synonyms and antonyms despite small cosine similarities between corresponding vectors. This information is implicitly encoded in the geometry of the embeddings and could be extracted with a straightforward manifold learning procedure or a *contrasting map*. Such a map is trained on a small labeled subset of the data and can produce new embeddings that explicitly highlight specific semantic attributes of the word. The new embeddings produced by the map are shown to improve the performance on downstream tasks.

Keywords. word embeddings, antonyms, manifold learning, contrasting map, word representations

1. Introduction

Modern word embeddings, such as [1], [2] or [3] are based on the so-called distributional hypothesis [4]. If two words are often used in a similar context, they should have a small cosine similarity between the embeddings. Naturally, such methods often fail to recognize antonyms since antonymous words, e.g., ”fast” and ”slow”, occur in similar contexts. Many researchers address this issue from different angles.

Some authors address representations of antonyms, injecting additional information, and improving training procedures. In [5] the authors use deep learning combined with various types of semantic knowledge to produce new word embeddings that show better performance on a word similarity task. In [6] information from thesauri is combined with distributional information from large-scale unlabelled text data and obtained embeddings are used to distinguish antonyms. The authors of [7] represent semantic knowledge extracted from thesauri as many ordinal ranking inequalities and formulate the learning of semantic word embeddings as a constrained optimization problem. In [8] the authors develop these ideas further and adjust word vectors using the semantic intensity information alongside with thesauri. In [9] thesauri along with the sentiment are used to build new embeddings that contrast antonyms. In [10] authors improve the weights of feature

[1] Corresponding Author: LEYA Lab, Yandex and Higher School of Economics in St. Petersburg; Kantemirovskaya st. 3, Saint-Petersburg, Russia; E-mail: ivan@yamshchikov.info

vectors with a special method based on local mutual information and propose an extension of the skip-gram model that integrates the new vector representations into the objective function. In [11] and [12] it is shown that translation-based embeddings perform better in applications that require concepts to be organized according to similarity and better capture their true ontologic status. The authors of [13] use these ideas and demonstrate that adding a multilingual context when learning embeddings allows improving their quality via deep canonical correlation analysis.

Other researchers try to develop novel approaches that are not heavily relying on the distributional hypothesis. For example, in [14] authors introduce word-level vector representation based on symmetric patterns and report that such representations allow controlling the model judgment of antonym pairs. A special *contrasting embedding framework* is developed in [15]. While in [16] the authors train a neural network model that exploits lexico-syntactic patterns from syntactic parse trees to distinguish antonyms.

All works mentioned above were based on the assumption that antonym distinguishing information is not captured by modern word embeddings. However, this assumption is frequently questioned in the last several years. In particular, one can inject information on hyponyms, hyperonyms, synonyms, and antonyms to distinguish the obtained embeddings using additional linguistic constraints, see [17], [18] and [19]. Moreover, in [20] the authors come up with a two-phase training of a siamese network that transforms initial embeddings into the ones that clearly distinguish antonyms. While the authors of [21] develop an architecture of a distiller that extracts information on antonyms out of the pre-trained vectors.

In this work, we demonstrate that Word2Vec [1], GloVe [2], and especially FastText [3] embeddings contain information that allows distinguishing antonyms to certain extent. This information is encoded in the geometry of the obtained vector space. We propose a very simple and straightforward approach for the extraction of this information. Similarly to [20] it is based on a siamese network, yet does not require a two-phase training and is more intuitive than the one proposed in [21]. We also show that this approach could be used further to extract other semantic aspects of words out of the obtained embedding space with ease.

The contribution of this paper is as follows:

- we demonstrate that modern word embeddings contain information that allows distinguishing synonyms and antonyms;
- we show that this information could be retrieved by learning a nonlinear manifold via supervision provided by a small labeled sub-sample of synonyms and antonyms;
- we demonstrate that concatenation of these new embeddings with original embeddings improves the performance on the downstream tasks that are sensitive to synonym-antonym distinction.

2. Data

For the experiments, we used the small supervised set of synonyms and antonyms of English language provided by WordNet[2] that we enriched with additional data from [16]

[2] https://wordnet.princeton.edu/

and several other publicly available sources[3]. We tested the methodology described below across multiple modern word embeddings, namely, FastText[4], GloVe pre-trained on Wikipedia[5] and GloVe pre-trained on Google News[6] alongside with Word2Vec pre-trained on Google News[7]. In Figure 1 one could see initial distributions of cosines between synonyms and antonyms in four different training datasets respectively.

The WordNet dataset of synonyms and antonyms consists of 99833 word pairs. Synonymic relations are neither commutative nor transitive. For example, ”economical” could be labeled as a synonym to ”cheap,” yet the opposite is not true[8]. At the same time, if ”neat” is denoted as a synonym to ”cool” and ”cool” is denoted as a synonym to ”serene,” this does not imply that ”neat” and ”serene” are synonyms as well. All data sources used in this paper are in the public domain. To facilitate reproducibility, we share the code of the experiments[9]

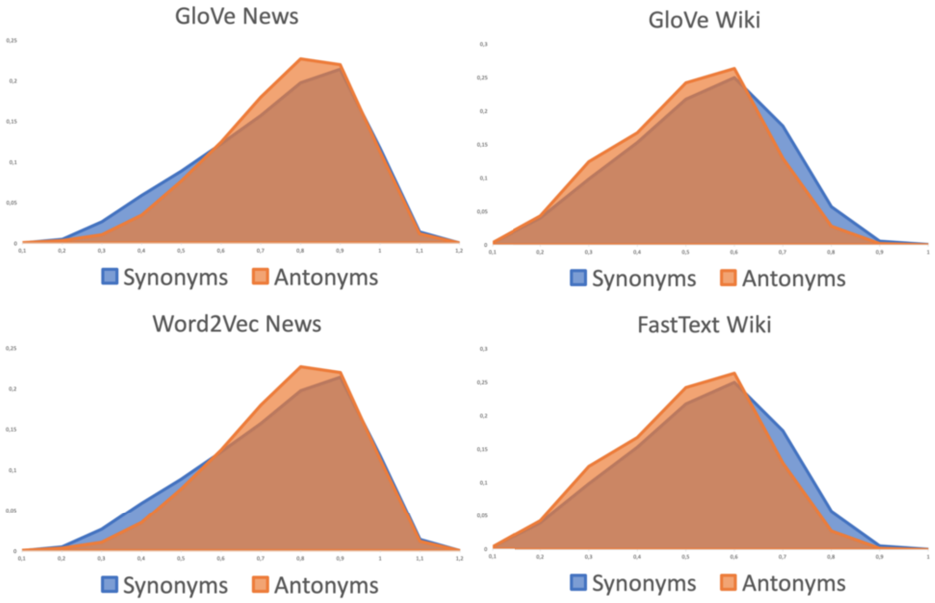

Figure 1. Distribution of cosine distances between synonyms and antonyms across four different sets of embeddings.

We propose the following train-test split procedure that guarantees that the words from the training dataset do not infiltrate the test set. We add pairs to train and test with

[3] https://github.com/ec2604/Antonym-Detection
[4] https://fasttext.cc/
[5] https://nlp.stanford.edu/projects/glove/
[6] https://www.kaggle.com/pkugoodspeed/nlpword2vecembeddingspretrained
[7] https://github.com/mmihaltz/word2vec-GoogleNews-vectors
[8] https://www.thesaurus.com/browse/cheap
[9] https://github.com/i-samenko/Triplet-net/

relative frequencies of 3 to 1. If one of the words in the pair was already in the train or test, we were adding the new pair to the corresponding subset. If one word in the pair occurs both in train and in the test, we deleted such a pair. After such a test-train split, we obtained 80 080 pairs. 65 292 pairs of 26 264 unique words formed the training dataset, and 14 788 pairs of 8737 unique words comprised the test dataset.

Figure 1 seems to back up the widespread intuition that modern embedding can not distinguish synonyms and antonyms. However, in the next sections, this paper demonstrates that this statement does not hold.

3. Learning Contrasting Map

If one assumes that information allowing to distinguish synonyms and antonyms is already present in the raw embeddings, one could try to extract it by building a manifold learning procedure that would take original embedding as input and try to map it in a new space of representations, where the synonym-antonym contrast becomes explicit.

The initial embedding space is $\mathbb{R}^m$ with a distance D_m defined on it, and for every word 'w', for any of its synonyms 's', and for any of its antonyms 'a' the following holds $D_m(w,s) \simeq D_m(w,a)$. A new embedding space of lower dimension $\mathbb{R}^k$ has a corresponding distance D_k. One would like to find a map $f : \mathbb{R}^m \to \mathbb{R}^k$ such that the following holds $D_k(w,s) < D_k(w,a)$ in a new $\mathbb{R}^k$ embedding space.

$$f = \begin{cases} f : \mathbb{R}^m \to \mathbb{R}^k, \quad m >> k; \\ D_k(w,s) < D_k(w,a), \quad \forall w,s,a. \end{cases} \tag{1}$$

Since the amount of synonyms and antonyms in any given language is growing excessively with the growth of the training sample of texts, one can not check these conditions for every word pair explicitly. One can only use a labeled subset of the vocabulary, where synonyms and antonyms are contrasted already, so it is hard to establish a procedure that would guarantee Inequalities 1, hence we use $\lesssim$ for the conditions. At the same time, despite the limited size of the training dataset, one would hope that the obtained representations are general enough to distinguish the synonyms and antonyms that are not included in the training data.

To train such a map let us regard an architecture, shown in Figure 2. It is a 'Siamese' network [22] where weights are shared across three identical EmbeddingNets. Each EmbeddingNet maps the word 'w', its synonym 's' and its antonym 'a' respectively. The resulting cosine similarities between synonyms and antonyms are simply included in the loss function in such a way that $D_k(w,s)$ is minimized and $D_k(w,a)$ is maximized explicitly. The whole system is trained end-to-end on 65 292 pairs of synonyms and antonyms described in Section 2.

4. Experiments

First of all, let us check if the condition listed in Equation 1 is satisfied in the transformed embedding space $\mathbb{R}^k$. Figure 3 illustrates the distributions of the cosine distances

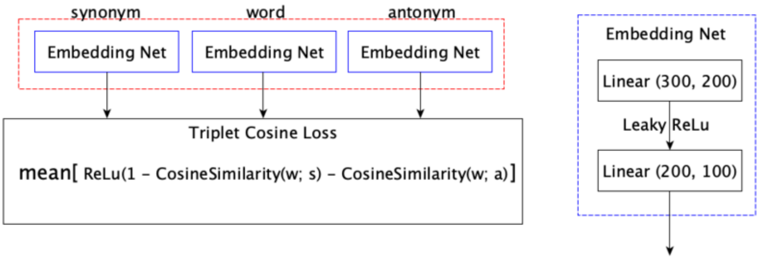

Figure 2. Siamese Triplet Network trained to distinguish synonyms and antonyms. EmbeddingNet is the contrasting map $f : \mathbb{R}^m \to \mathbb{R}^k$. The weights of three EmbeddingNets are shared in the end-to-end training. The resulting architecture is trained to minimize cosine similarities between synonyms and maximize the cosine similarities between antonyms in the transformed low-dimensional embeddings space $\mathbb{R}^k$.

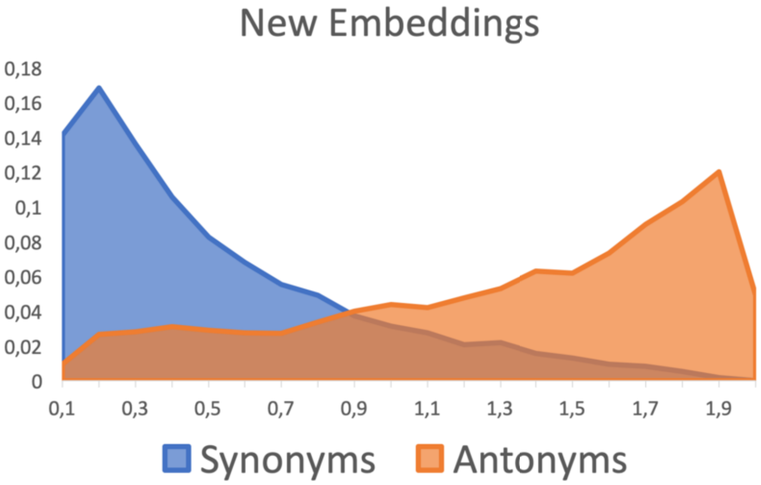

Figure 3. Distribution of cosine distances between synonyms and antonyms in the transformed space $\mathbb{R}^k$ for FastText. Test set. Different datasets produce similar results. Distances between synonyms tend to become smaller, distances between antonyms tend to increase.

between synonyms and antonyms in $\mathbb{R}^k$ for English FastText embeddings. The situation is drastically improved in contrast with raw embeddings shown in Figure 1.

One can have a close look at the tails of the distributions shown in Figure 3. To simplify further experiments and improve reproducibility we also publish the resulting distances for the test set[10].

[10] https://github.com/i-samenko/Triplet-net/

Here are some examples of word pairs that were marked as antonyms in the test dataset, yet are mapped close to each other by the contrast map: sonic — supersonic; fore — aft; actinomorphic — zygomorphic; cable — hawser; receive — give; ceiling — floor. Here are some examples of word pairs that were marked as synonyms in the test dataset, yet are mapped far of each other by the contrast map: financial — fiscal; mother — father; easy — promiscuous; empowered — sceptred; formative — plastic; frank — wiener; viii — eighter; wakefulness — sleeplessness. One can see that some of the contrasting map errors are due to the debatable labeling of the test dataset, others occur with the words that are rare.

To be sure that other properties of the original embeddings are preserved we concatenate new embedding with the old, raw ones. Figure 4 depicts the difference of the pairwise distance between synonyms and antonyms in the space of concatenated embeddings $D_{\mathbb{R}^m \bigoplus \mathbb{R}^k}$ and in the space of raw embeddings $D_{\mathbb{R}^m}$. The distributions are obtained for the test subset of data. The map did not see these word pairs in training.

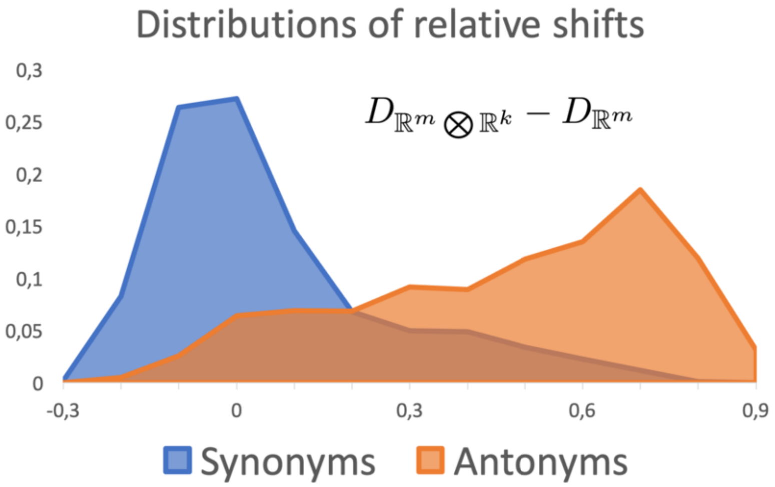

Figure 4. Cosine distances between synonyms and antonyms in the raw embeddings and in the space where they are concatenated with the new ones. FastText embeddings. Test set.

We train an XGBoost classifier on four different raw embeddings and check the resulting accuracy of the classifiers on the test subset of synonym and antonym pairs. Table 1 clearly shows that the accuracy of a classifier trained on raw embeddings is consistently lower than the accuracy of the same classifier trained on the newly transformed embeddings, produced by the EmbeddingNet. One can also see that a classifier trained on the concatenation of the raw embeddings with the new ones also outperforms the classifier trained solely on the original embeddings.

FastText embeddings are capturing more than 80% of synonym-antonym relations with the proposed contrasting map, and more than 70% out of these relations are captured out of the box. GloVe embeddings seem to contain the least information on the synonym-

Table 1. Comparison of four different embeddings. For every type of embedding, XGBoost classifier is trained to distinguish two input vectors as synonyms or antonyms.

Embeddings type	Raw	New	Concatenated
Word2Vec	0.67	0.85	0.81
GloVe Wiki	0.65	0.75	0.72
GloVe Google News	0.67	0.84	0.78
FastText	0.73	**0.88**	**0.85**

Table 2. Concatenation of the original FastText embeddings with transformed embeddings improves the accuracy of logistic regression-based classifiers trained on various datasets.

Dataset	FastText only	Concatenated
IMDB reviews	0.86	0.88 **(+2.2%)**
Cornell reviews	0.75	0.76 **(+1.0%)**
Toxic Comments	0.94	0.95 **(+0.6%)**
MDSD books	0.69	0.77 **(+11.3%)**
MDSD DVDs	0.70	0.76 **(+8.0%)**
MDSD electronic	0.72	0.78 **(+9.4%)**
MDSD kitchen	0.78	0.80 **(+3.4%)**
MDSD all categories	0.76	0.79 **(+3.6%)**

antonym relations. Further experiments are conducted on FastText embeddings since they capture the most out of synonym-antonym relations.

To illustrate the potential usage of such embeddings obtained with a contrasting map we run a series of experiments with various NLP datasets that intuitively might need to contrast synonyms and antonyms for the successful performance: binary sentiment classifier for IMDB reviews[11], binary sentiment classifier for Cornell movie reviews[12], binary classifier to identify toxic comments[13], sentiment classifiers across several thematic domains of Multi-Domain Sentiment Dataset[14].

For every dataset, we trained a logistic regression using pre-trained FastText embeddings and measured its accuracy on the test. Then we retrained the same logistic regression with new concatenated embeddings. Table 2 demonstrates how the usage of the transformed embeddings improves the accuracy on various datasets.

5. Discussion

The proposed methodology demonstrates that contrary to common intuition modern word embeddings contain information that allows distinguishing synonyms and antonyms. The approach could possibly be scaled to other semantic aspects of the words. In its most general form, the approach allows mapping original embeddings into spaces of lower dimensions that could explicitly highlight certain semantic aspects using a la-

[11] https://ai.stanford.edu/ãmaas/data/sentiment/

[12] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[13] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[14] http://www.cs.jhu.edu/m̃dredze/datasets/sentiment/

beled dataset of a limited size. This semantic information can be effectively incorporated into the downstream tasks.

Conceptually, the proposed methodology allows for revisiting the questions of language acquisition in the context of the distributional hypothesis. If one assumes that semantic information attached to a given word is not a rigid structure but depends on the training corpus, it seems that modern embeddings capture these diverse semantic fields successfully, provided the corpus is large enough. This result does not mean that such semantic aspects are explicit and could be immediately extracted out of the embeddings. The spaces of modern word embeddings could be profoundly nonlinear concerning a given semantic attribute of the word. A deeper understanding of the geometric properties of these spaces could significantly improve the quality of the resulting models. Indeed, the very assumption that semantic similarity could be captured with cosine distance in Euclidian space is debatable.

Though the choice of the embedding space and the notion of distance on it both need further, more in-depth investigations, this paper demonstrates the simple methods of representational learning applied to the raw embeddings can distill this implicitly encoded information reasonably well.

6. Conclusion

This paper demonstrates that, contrary to a widely spread opinion, modern word embeddings contain information that allows distinguishing synonyms from antonyms. This information is encoded in the geometry of the embeddings and could be extracted with manifold learning. The paper proposes a simple and intuitive approach that allows obtaining a *contrasting map*. Such a map could be trained on a small subset of the vocabulary and is shown to highlight relevant semantic information in the resulting vector embedding. The new embeddings, in which the information on synonyms and antonyms is disentangled, improve the performance on the downstream tasks. The proposed methodology of contrasting maps could potentially be further extended to other semantic aspects of the words.

References

[1] Mikolov T, Chen K, Corrado G, Dean J. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:13013781. 2013.

[2] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

[3] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 2017;5:135–146.

[4] Harris Z. Distributional Hypothesis. Word. 1954;10(23):146–162.

[5] Bian J, Gao B, Liu TY. Knowledge-powered deep learning for word embedding. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2014. p. 132–148.

[6] Ono M, Miwa M, Sasaki Y. Word embedding-based antonym detection using thesauri and distributional information. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2015. p. 984–989.

[7] Liu Q, Jiang H, Wei S, Ling ZH, Hu Y. Learning semantic word embeddings based on ordinal knowledge constraints. In: Proceedings of the 53rd Annual Meeting of the Association for Computational

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015. p. 1501–1511.

[8] Kim JK, de Marneffe MC, Fosler-Lussier E. Adjusting word embeddings with semantic intensity orders. In: Proceedings of the 1st Workshop on Representation Learning for NLP; 2016. p. 62–69.

[9] Dou Z, Wei W, Wan X. Improving word embeddings for antonym detection using thesauri and SentiWordNet. In: CCF International Conference on Natural Language Processing and Chinese Computing. Springer; 2018. p. 67–79.

[10] Nguyen KA, im Walde SS, Vu NT. 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); 2016. p. 454–459.

[11] Hill F, Cho K, Jean S, Devin C, Bengio Y. Not all neural embeddings are born equal. arXiv preprint arXiv:14100718. 2014.

[12] Hill F, Cho K, Jean S, Devin C, Bengio Y. Embedding word similarity with neural machine translation. arXiv preprint arXiv:14126448. 2014.

[13] Lu A, Wang W, Bansal M, Gimpel K, Livescu K. Deep multilingual correlation for improved word embeddings. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2015. p. 250–256.

[14] Schwartz R, Reichart R, Rappoport A. Symmetric pattern based word embeddings for improved word similarity prediction. In: Proceedings of the nineteenth conference on computational natural language learning; 2015. p. 258–267.

[15] Chen Z, Lin W, Chen Q, Chen X, Wei S, Jiang H, et al. Revisiting word embedding for contrasting meaning. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015. p. 106–115.

[16] Nguyen KA, Schulte im Walde S, Vu NT. 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 (EACL). Valencia, Spain; 2017. .

[17] Mrkšić N, Vulić I, Séaghdha DÓ, Leviant I, Reichart R, Gašić M, et al. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. Transactions of the association for Computational Linguistics. 2017;5:309–324.

[18] Vulić I, Mrkšić N. Specialising Word Vectors for Lexical Entailment. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers); 2018. p. 1134–1145.

[19] Vulić I. Injecting lexical contrast into word vectors by guiding vector space specialisation. In: Proceedings of The Third Workshop on Representation Learning for NLP; 2018. p. 137–143.

[20] Etcheverry M, Wonsever D. Unraveling Antonym's Word Vectors through a Siamese-like Network. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019. p. 3297–3307.

[21] Ali MA, Sun Y, Zhou X, Wang W, Zhao X. Antonym-synonym classification based on new sub-space embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33; 2019. p. 6204–6211.

[22] Bromley J, Guyon I, LeCun Y, Säckinger E, Shah R. Signature verification using a" siamese" time delay neural network. In: Advances in neural information processing systems; 1994. p. 737–744.