

Bridging the Defined and the Defining: Exploiting Implicit Lexical Semantic Relations in Definition Modeling

Koki Washio^{*†1}

Satoshi Sekine^{†2}

Tsuneaki Kato^{*3}

^{*}Department of Language and Information Sciences, The University of Tokyo

[†]RIKEN Center for Advanced Intelligence Project

{¹kokiwashio@g.ecc, ³kato@boz.c}.u-tokyo.ac.jp

²satoshi.sekine@riken.jp

Abstract

Definition modeling includes acquiring word embeddings from dictionary definitions and generating definitions of words. While the meanings of defining words are important in dictionary definitions, it is crucial to capture the lexical semantic relations between defined words and defining words. However, thus far, the utilization of such relations has not been explored for definition modeling. In this paper, we propose definition modeling methods that use lexical semantic relations. To utilize implicit semantic relations in definitions, we use unsupervisedly obtained pattern-based word-pair embeddings that represent semantic relations of word pairs. Experimental results indicate that our methods improve the performance in learning embeddings from definitions, as well as definition generation.

1 Introduction

Dictionary definitions are rich resources of semantic information for both humans and machines. Recent studies on definition modeling are primarily divided into two categories. One is definition generation, in which a definition is generated for a target word from its word representation. Definition generation involves analyzing word embeddings using generated definitions (Noraset et al., 2017) and machine explanations of word meanings for human readers (Ni and Wang, 2017; Ishiwatari et al., 2019). The other is learning word embeddings from definitions to obtain semantic-oriented word representations (Tissier et al., 2017; Bosc and Vincent, 2018).

Although previous methods for encoding or decoding definitions using recurrent neural networks (RNNs) yielded promising results in definition modeling (Noraset et al., 2017; Bosc and Vincent, 2018), they did not explicitly utilize lexical semantic relations between defined words and defin-

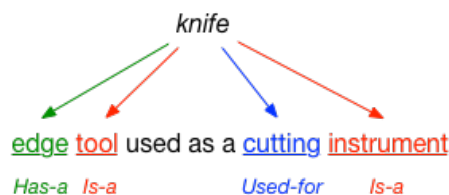


Figure 1: Definition of *knife* from WordNet (Fellbaum, 1998) and lexical semantic relations between the defined word and the defining words.

ing words. Various lexical relations exist in definitions (Amsler, 1981), as displayed in Figure 1, where the defined word *knife* exhibits an *Is-a* relation with the defining words, *tool* and *instrument*, *Has-a* relation with *edge*, and *Used-for* relation with *cutting*. Utilizing structures of definitions about lexical semantic relations facilitates the understanding and generation of definitions.

Based on this observation, we propose definition modeling methods that exploit lexical semantic relations between defined and defining words. However, lexical semantic relations in definitions are not explicit. To solve this problem, we use unsupervisedly learned word-pair embeddings that represent semantic relations of word pairs based on co-occurring relational patterns in a corpus (Turney, 2005; Washio and Kato, 2018b). Experimental results show that our definition modeling methods improve previous models, with respect to both definition generation and the acquisition of word embeddings from definitions.

2 Background

2.1 Definition Embedding

Relationships between words captured in embeddings are studied in terms of *similarity* or *relatedness* (Hill et al., 2015). For example, *coffee* and *cup* have high relatedness because *coffee* is often contained in a *cup*; meanwhile, these words ex-

hibit low similarity as *coffee* is a beverage and *cup* is a container.

Definition embeddings that are learned from word definitions are useful to capture similarity, while word embeddings based on distributional hypothesis (Mikolov et al., 2013) tend to capture relatedness (Bosc and Vincent, 2018).

Bosc and Vincent (2018) proposed a method that learns the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) that encodes a word definition into an embedding. Given a defining word sequence $D = \{w_1, \dots, w_T\}$, an LSTM definition encoder processes D , as follows:

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, \mathbf{w}_t) \quad (1)$$

$$\mathbf{h} = \mathbf{W}_e \mathbf{h}_T + \mathbf{b}_e \quad (2)$$

where *LSTM* computes the hidden state given the previous hidden state \mathbf{h}_{t-1} and the input word embedding \mathbf{w}_t along with the LSTM architecture. \mathbf{h}_T is the final hidden state. \mathbf{W}_* and \mathbf{b}_* are weight matrices and bias terms, respectively. The definition encoder is trained to reconstruct a bag-of-words representation of the definition from \mathbf{h} with a consistency penalty that renders \mathbf{h} closer to the corresponding word embedding used in the LSTM. This method is referred to as *consistency penalized autoencoder (CPAE)*.

2.2 Definition Generation

Definition generation was introduced by Noraset et al. (2017). The goal of definition generation is to predict the probability of the defining word sequence D given the defined word w_{trg} . In the aforementioned study, the LSTM conditional language model was used as a definition decoder to model this probability as follows:

$$p(D|w_{trg}) = \prod_{t=1}^T p(w_t|w_{i<t}, w_{trg}) \quad (3)$$

$$p(w_t|w_{i<t}, w_{trg}) = Softmax(\mathbf{W}_d \mathbf{h}'_t + \mathbf{b}_d) \quad (4)$$

where \mathbf{h}'_t is a hidden state from the LSTM definition decoder. *Softmax* is the softmax function. They conditioned the decoder by providing the embedding of the defined word at the first step of the LSTM. They referred to this model as the *Seed (S)*. Moreover, they extended this model to update the output of the recurrent unit with a gate function depending on the embedding of the defined word. This gate function, *Gate (G)*, controls the amount of information from the defined word

that contributes to the definition generation at each step. As additional features, they used morphological information from a character-level convolutional neural network (CNN) to process a character sequence of the defined word and the embeddings of the hypernyms from the WebIsA database (Seitner et al., 2016). These features are called *CH* and *HE*, respectively.

Gadetsky et al. (2018) introduced context-aware definition generation to disambiguate polysemous words with their context and generate the corresponding definitions. They extended Equation 3 to consider the context word sequence of the defined word $C = \{c_1, \dots, c_m\}$ as follows:

$$p(D|w_{trg}, C) = \prod_{t=1}^T p(w_t|w_{i<t}, w_{trg}, C) \quad (5)$$

To model this probability, they used an attention mechanism to extract meaningful information from C and chose relevant dimensions of the embedding of the defined word. They referred to this model as *Input Attention (I-Attention)*.

2.3 Word-Pair Embedding

Word-pair embeddings represent relations of word pairs. Although representation of word pairs as the vector offsets of their pretrained word embeddings is a simple and powerful method (Mikolov et al., 2013), recent studies have shown that neural pattern-based word-pair embeddings are more effective than vector offsets in various tasks such as calculating relational similarity (Washio and Kato, 2018b), natural language inference, and question answering (Joshi et al., 2019).

Neural pattern-based word-pair embedding models (Washio and Kato, 2018a,b; Joshi et al., 2019) unsupervisedly learn two neural networks: a word-pair encoder and pattern encoder, both of which encode the word-pair and lexico-syntactic pattern respectively into the same embedding space. These networks are trained by predicting co-occurrences between word-pairs and patterns in a corpus with the negative sampling objective. After the unsupervised learning, the word-pair encoder provides word-pair embeddings for any word pair given their word embeddings.

3 Method

We propose methods that consider lexical semantic relations between the defined word and the

defining words in the definition encoder for definition embeddings (Section 3.1) and the definition decoder for definition generation (Section 3.2). To utilize the implicit semantic relations in definitions, we use word-pair embeddings that represent semantic relations of word pairs. We describe how word-pair embeddings are obtained in Section 4.1

3.1 For Definition Encoder

To consider lexical semantic relations in acquiring embedding from definitions, we feed the word-pair embeddings to the definition encoder. Assuming that the pair embedding $\mathbf{v}_{(w_{trg}, w_t)}$ represents the relation between the defined word w_{trg} and the t -th defining word w_t , we calculate \mathbf{h}_t as follows:

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, [\mathbf{w}_t; \mathbf{v}_{(w_{trg}, w_t)}]) \quad (6)$$

where $;$ denotes vector concatenation. To exclude meaningless relations between the defined word and functional words, we replace $\mathbf{v}_{(w_{trg}, w_t)}$ with the zero vector if w_t is a stopword. With word-pair embeddings as inputs, the definition encoder can recognize the role of information that is provided by the current word w_t , for example, a type of w_{trg} (*Is-a*), a goal of w_{trg} (*Used-for*), or a component of w_{trg} (*Has-a*).

3.2 For Definition Decoder

To provide the definition decoder with information regarding lexical semantic relations, we use an additional loss function with word-pair embeddings as follows:

$$L_{rel} = \frac{1}{|K|} \sum_{w_t \in K} \left\| \mathbf{v}_{(w_{trg}, w_t)} - (\mathbf{W}_r \mathbf{h}'_t + \mathbf{b}_r) \right\|^2 \quad (7)$$

$$K = \{w_t | w_t \in D \wedge w_t \notin S\} \quad (8)$$

where S is a set of stopwords. As in Section 3.1, we ignore the loss when w_t is a stopword. This additional loss allows the definition decoder to learn the pattern of what semantic relations occur in definitions and how they occur. For example, if w_{trg} indicates a type of tools, a defining word that has the *Is-a* relation to w_{trg} tends to be followed by the *Used-for* word.

4 Experiments and Results

The experiments conducted to evaluate our methods for definition encoding and decoding are presented in this section. In the Appendix, we describe the details of the hyperparameter settings and optimization methods used in the experiments.

4.1 Obtaining Word-Pair Embedding

To obtain pattern-based word-pair embeddings, we extracted triples $(w_1, w_2, p) \in T$ from the Wikipedia corpus, where (w_1, w_2) is a word pair composed of nouns, verbs, or adjectives in the the 100K most frequent words of the GloVe¹ (Pennington et al., 2014), and p is the co-occurring shortest dependency path². We discarded the triples if p occurred less than five times and subsampled the triples based on word-pair probability with a threshold of $5 \cdot 10^{-7}$, following Joshi et al. (2019).

For the word-pair encoder, we used the neural networks as follows:

$$\mathbf{h}_{(w_1, w_2)} = MLP([\mathbf{v}_{w_1}; \mathbf{v}_{w_2}; \mathbf{v}_{w_1} \odot \mathbf{v}_{w_2}]) \quad (9)$$

where MLP is a four-layer perceptron with the ReLU activation, \mathbf{v}_w is a word embedding of w , and \odot is the element-wise product. Each size of the hidden states of MLP was 300.

The dependency path p is a sequence composed of one to three lemmas and dependency relations e_1, \dots, e_n . The sequence of the corresponding embeddings e_1, \dots, e_n was encoded using the bidirectional LSTM with the 300-dimensional hidden state as the pattern encoder. Then, the 300-dimensional pattern representation \mathbf{v}_p was calculated with the final output vectors \mathbf{h}_f and \mathbf{h}_b from the forward and backward LSTM as follows:

$$\mathbf{v}_p = \mathbf{W}_p[\mathbf{h}_f; \mathbf{h}_b] + \mathbf{b}_p \quad (10)$$

The word embeddings in the models were initialized by the 300-dimensional GloVe. We used the multivariate negative sampling objective (Joshi et al., 2019) to train the parameters with the data T for 10 epochs. Adagrad, which has a learning rate of 0.01, was used as the optimizer (Duchi et al., 2011). After the training was completed, word-pair embeddings are calculated as follows:

$$\mathbf{v}_{(w_1, w_2)} = \left[\frac{\mathbf{h}(w_1, w_2)}{\|\mathbf{h}(w_1, w_2)\|}; \frac{\mathbf{h}(w_2, w_1)}{\|\mathbf{h}(w_2, w_1)\|} \right] \quad (11)$$

4.2 Definition Embedding

For the evaluation of definition embeddings, we used the modified Word Embedding Benchmarks projects³, following Bosc and Vincent (2018). These benchmarks include SimLex999 (SL999),

¹<http://nlp.stanford.edu/data/glove.6B.zip>

²We used spaCy for the dependency parsing.

³<https://github.com/tombosc/cpae>

	Development		Similarity			Relatedness				
	SV-dev	MEN-dev	SL999	SL333	SV-test	RG	SCWS	MEN-test	MT	WS353
GloVe	22.0	73.5	37.1	20.7	22.0	77.0	55.9	74.2	65.0	47.8
Google	39.3	76.0	44.2	29.7	35.8	76.1	66.0	75.6	67.1	63.5
CPAE	47.6	69.4	48.1	33.3	42.4	82.7	63.4	70.1	60.2	66.8
Ours	49.9	72.8	48.5	33.8	44.4	81.2	66.7	74.3	67.6	65.4

Table 1: Spearman’s correlation coefficient $\rho \times 100$ on the benchmarks.

Context-agnostic (Noraset et al., 2017)		
Model	PPL	BLEU
S+G+CH+HE	46.8	35.4
w/ L_{rel}	39.5	37.9
Context-aware (Gadetsky et al., 2018)		
Model	PPL	BLEU
S+I-Attention	59.6	12.0
w/ L_{rel}	43.8	12.3

Table 2: Perplexity and equally-weighted BLEU scores for up to 4-grams on the definition generation datasets

SimLex333 (SL333) (Hill et al., 2015), SimVerb (SV) (Gerz et al., 2016), MEN (Bruni et al., 2014), RG (Rubenstein and Goodenough, 1965), WS353 (Finkelstein et al., 2002), SCWS (Huang et al., 2012), and MTurk (Radinsky et al., 2011; Halawi et al., 2012). To evaluate the definition embeddings, we scored word pairs in the benchmarks using the cosine similarity between the corresponding definition embeddings and calculated Spearman’s correlation to the ground truth. The definitions in WordNet were used to train the definition encoder. The development sets of the SimVerb and MEN were used for the hyperparameter tuning.

We implemented the CPAE (Section 2.1) as a baseline and compared it to CPAE with the word-pair embeddings (Section 3.1), which is our proposed method. The word embeddings in the definition encoder were initialized by the Google Word2Vec vectors⁴. Google Word2Vec vectors and GloVe were used as the other baselines.

Table 1 shows the results of the similarity and relatedness benchmarks. While the baseline CPAE outperformed our model on the two out of five datasets pertaining to word relatedness, our model consistently outperformed the baseline on the similarity benchmarks. These results indicate that the word-pair embeddings provide the definition encoder with useful semantic information about the target word.

Hidden States of Definition Encoder

To analyze the functionality of injecting semantic relations between the defined word and the defin-

⁴<https://code.google.com/archive/p/word2vec/>

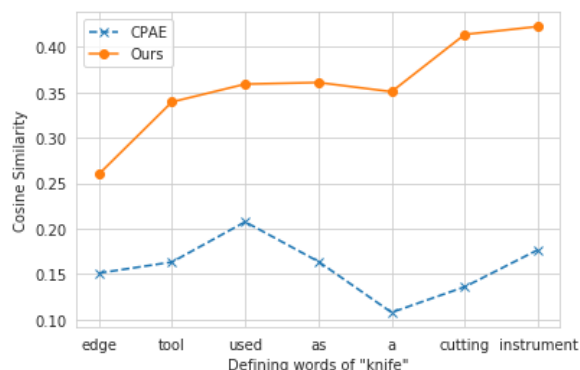


Figure 2: Cosine similarities between the last hidden states of the definition encoders encoding *kettle*’s definition and each hidden states representing the *knife*’s definition

ing words into the definition encoder, we investigate the similarities between hidden states of the definition encoders.

We encoded the *kettle*’s definition (*a metal pot for stewing or boiling*) and the *knife*’s definition (*edge tool used as a cutting instrument*) with the definition encoders. These two definitions are similar in their style, composed of the defining words of *Is-a* relations (*pot* and *tool*) and *Used-for* relations (*stewing*, *boiling*, and *cutting*.)

Figure 2 displays the cosine similarities between the last hidden state of the encoded *kettle*’s definition and each hidden state of the encoded *knife*’s definition. This figure shows that when *tool* with *Is-a* relation and *cutting* with *Used-for* relation to *knife* were input to the encoder, our method increased the similarities more than the baseline. This indicates that our method successfully helps the model capture the similarities of definitions in terms of lexical semantic relations.

4.3 Definition Generation

We evaluated our method for the definition decoder (Section 3.2) on a context-agnostic dataset (Noraset et al., 2017) and context-aware dataset (Gadetsky et al., 2018) for the definition generation. For evaluation purposes, we calculated the perplexity (PPL) and BLEU score (Papineni et al.,

	S+G+CH+HE	w/ L_{rel}
academician	a person who specializes in a particular profession	one who is versed in a scholarly or scientific field
artist	one who is a person who is a person or thing is made	one who creates a picture or representation of a creative work
adolescence	the state of being pregnant	the state of being mature

Table 3: Examples of definitions generated by the baseline $S + G + CH + HE$ and the one with L_{rel} on the development set of Noraset et al. (2017)

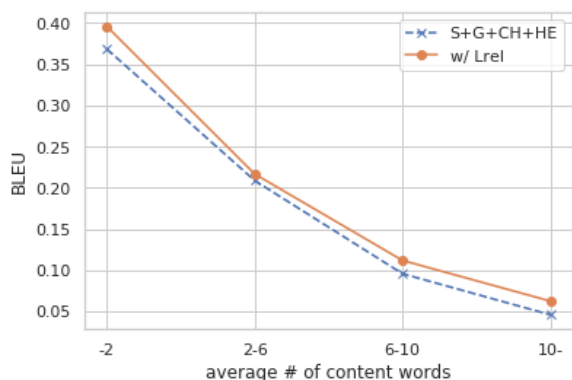


Figure 3: BLEU scores at the average numbers of content words in the reference definitions on the development set.

2002), following Noraset et al. (2017). We implemented $S+G+CH+HE$ for the context-agnostic dataset and $S+I-Attention$ for the context-aware dataset as baselines, and compared these models to the ones with L_{rel} in Section 3.2. The word embeddings in the models were initialized by Google Word2Vec vectors, as in Section 4.2.

The results in Table 2 show that training with our L_{rel} improves the performance in both the context-agnostic and context-aware settings.

Effect to Definition Generation

To analyze the effect of L_{rel} , we plotted the BLEU scores at each average number of content words in reference definitions from the development set of Noraset et al. (2017), as shown in Figure 3. This figure shows that when many content words exist in reference definitions, the model with L_{rel} is stronger than that without L_{rel} . This indicates that our method allows the model to select correct words when the defined word requires detailed descriptions.

Generated Definitions

Table 3 displays examples of generated definitions by the baseline $S + G + CH + HE$ and

the one with L_{rel} . The model with the proposed method successfully generated the definitions for *academician* and *artist*, while the baseline did not. Although the baseline generated the generalized class of the target words, for examples, *person* and *one*, it fails to generate details when the target word is distinct from others in the same class. In contrast, the model with the proposed method chose the correct words at both the generalized class and the details.

For *adolescence*, both models could not generate the correct definitions. Even with our method, the model produced a definition that has an opposite meaning to the target word. The generation of opposite definitions is a significant problem in the definition generation from word embeddings (Noraset et al., 2017). Although our method helps the model generate details about the target words, this problem requires other approaches that consider antonym relations.

5 Conclusion

In this paper, we proposed definition modeling methods that utilize lexical semantic relations between defined words and defining words. To utilize implicit semantic relations in dictionary definitions, we used pattern-based word-pair embeddings. The experimental results demonstrated that applying our methods to the definition encoder and the definition decoder improved their performance. In our future work, we will extend our methods for phrase-level definition generation (Ishiwatari et al., 2019).

Acknowledgments

This work was supported by JSPS KAKENHI Grant numbers JP17H01831. We would like to thank Shonosuke Ishiwatari for his help with the experiments.

References

- Robert A. Amsler. 1981. [A taxonomy for english nouns and verbs](#). In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Stanford, California, USA. Association for Computational Linguistics.
- Tom Bosc and Pascal Vincent. 2018. [Auto-encoding dictionary definitions into consistent word embeddings](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1532, Brussels, Belgium. Association for Computational Linguistics.
- Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.
- Lev Finkelstein, Evgeniy Gabilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2002. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.
- Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. [Conditional generators of words definitions](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 266–271, Melbourne, Australia. Association for Computational Linguistics.
- Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. [SimVerb-3500: A large-scale evaluation set of verb similarity](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.
- Guy Halawi, Gideon Dror, Evgeniy Gabilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. [SimLex-999: Evaluating semantic models with \(genuine\) similarity estimation](#). *American Journal of Computational Linguistics*, 41(4):665–695.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Eric Huang, Richard Socher, Christopher Manning, and Andrew Ng. 2012. [Improving word representations via global context and multiple word prototypes](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–882, Jeju Island, Korea. Association for Computational Linguistics.
- Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. [Learning to describe phrases with local and global contexts](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Omer Levy, Daniel S. Weld, and Luke Zettlemoyer. 2019. [pair2vec: Compositional word-pair embeddings for cross-sentence inference](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS’13*, pages 3111–3119, USA. Curran Associates Inc.
- Ke Ni and William Yang Wang. 2017. [Learning to explain non-standard english words and phrases](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 413–417, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. [Definition modeling: Learning to define word embeddings in natural language](#). In *The Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. [Bleu: a method for automatic evaluation of machine translation](#). In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Kira Radinsky, Eugene Agichtein, Evgeniy Gabilovich, and Shaul Markovitch. 2011. A

- word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.
- Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
- Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A large database of hyponymy relations extracted from the web. In *LREC*.
- Julien Tissier, Christophe Gravier, and Amaury Habrard. 2017. [Dict2vec : Learning word embeddings using lexical dictionaries](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Copenhagen, Denmark. Association for Computational Linguistics.
- Peter D. Turney. 2005. [Measuring semantic similarity by latent relational analysis](#). In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1136–1141.
- Koki Washio and Tsuneaki Kato. 2018a. [Filling missing paths: Modeling co-occurrences of word pairs and dependency paths for recognizing lexical semantic relations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1123–1133, New Orleans, Louisiana. Association for Computational Linguistics.
- Koki Washio and Tsuneaki Kato. 2018b. [Neural latent relational analysis to capture lexical semantic relations in a vector space](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 594–600, Brussels, Belgium. Association for Computational Linguistics.