



Discovering Representative Space for Relational Similarity Measurement

Huda Hakami^(✉) , Angrosh Mandya, and Danushka Bollegala

Computer Science Department, University of Liverpool, Liverpool, UK
{hshhakam, angrosh.mandya, danushka.bollegala}@liv.ac.uk

Abstract. Relational similarity measures the correspondence of the semantic relations that exist between the two words in word pairs. Accurately measuring relational similarity is important for various natural language processing tasks such as, relational search, noun-modifier classification, and analogy detection. Despite this need, the features that accurately express the relational similarity between two word pairs remain largely unknown. So far, methods have been proposed based on linguistic intuitions such as the functional space proposed by Turney [1], which consists purely of verbs. In contrast, we propose a data-driven approach for discovering feature spaces for relational similarity measurement. Specifically, we use a linear-SVM classifier to select features using training instances, where two pairs of words are labeled as analogous or non-analogous. We evaluate the discovered feature space by measuring the relational similarity for relational classification task in which we aim to classify a given word-pair to a specific relation from a predefined set of relations. Linear classifier for ranking the best feature for relational space has been compared with different methods namely, Kullback Leibler divergence (KL), Pointwise Mutual Information (PMI). Experimental results show that our proposed classification method accurately discovers a discriminative features for measuring relational similarity. Furthermore, experiments show that the proposed method requires small number of relational features while still maintaining reasonable relational similarity accuracy.

Keywords: Relational similarity · Feature selection
Proportional analogy detection

1 Introduction

Identifying the semantic relations that exist between two words (or entities) is one of the fundamental steps in many natural language processing (NLP) tasks. For example, to detect word analogies between pairs of words [2–4] such as (*water, pipe*) and (*electricity, wire*), we must first identify the relations that exist between the two words in each word pair (in this case *flows in*). In relational information retrieval [5], given a query *x is to y as z is to?* we would like to retrieve entities that have a semantic relationship with *z* similar to that between

x and y . For example, given the relational search query *Bill Gates is to Microsoft as Steve Jobs is to?*, a relational search engine is expected to return the result *Apple Inc.*

Despite the wide applications of relations in NLP systems, it remains a challenging task for humans to come up with representative features for identifying the semantic relation between two given words. In our previous example, the relationship between *Bill Gates* and *Microsoft* can be complex as Bill Gates is both a founder, a lead developer in many products, and a former CEO of the Microsoft. In order for a human to suggest representative features for identifying a relationship given only via an entity-pair instance, he/she must not only be familiar with the individual entities, but also know the different relations that would exist between those entities. Therefore, more automated methods for representing relations using descriptive features are necessary.

A popular strategy for representing the relation between two words is to extract lexical or syntactic patterns from the co-occurrence contexts of those words [6]. The extracted lexical patterns can then be used to measure the relational similarity between two word-pairs using a similarity measure defined over the distributions of patterns. Although surface patterns have been used successfully to represent the semantic relations between two words, it suffers from the data sparseness. The co-occurrences of two words with a specific pattern can be sparse even in a large corpus, requiring some form of a dimensionality reduction in practice [7]. It is also computationally expensive method because we must consider co-occurrences between surface patterns and all pairs of words. The number of all pairwise combinations between words grows quadratically with the number of words, and we require a continuously increasing set of surface patterns to cover the relations that exist between the two words in each of those word-pairs.

To overcome the above mentioned issues in the holistic approach, Turney [1, 8] proposed the *Dual Space* approach, where the relations between two words is *composed* using features related to individual words. Specifically, he used *nouns* and *verbs* as features for describing respectively the *domain* and *function* spaces. The proposal to use verbs as a proxy for the functional attributes of words that are likely to contribute towards semantic relations is based on linguistic intuition. Although this intuition is justified by the experimental results, the question *can we learn descriptors of semantic relations from labeled data?* remains unanswered.

We address this question by proposing a method for ranking lexical descriptors for representing semantic relations that exist between two words. Given a set of word-pairs for a particular relation type, we model the problem of extracting descriptive features as a linear classification problem. Specifically, we train a linear-SVM to discriminate between positive (analogous) and randomly generated pseudo-negative (non-analogous) word-pairs using features associated with individual words. The weights learnt by the classifier for the features can then be used as a ranking-score for selecting most representative features for a particular semantic relation. Experimental results on a benchmark dataset for relation

classification show that the proposed feature selection method outperforms several competitive baselines and previously proposed heuristics.

The paper is organized as follows: in Sect. 2 we discuss some related work of feature selection in NLP. The methodology adopted in this work is presented in Sects. 3 and 4. The dataset applied in this research with the experimental results are discussed in Sect. 5. Finally, we conclude the paper and discuss some possible future works.

2 Related Work

Identifying appropriate feature space for NLP tasks is a problem that have been studied widely in the literature. The most popular and effective method is based on matrix factorisation such as Non-Negative Matrix Factorization (NMF), Principle Component Analysis (PCA) and Singular Value Decomposition (SVD). Basically, those methods aim to transform the high-dimensional distributional representations to low-dimensional latent space. For word-level representation, Latent Semantic Analysis (LSA) is a method relying on SVD to represent a word in a vector space using only top, i.e. 300 or more, dimensions to capture the meaning of words in the low-dimensional latent space [9]. For word pairs representation, Latent Relational Analysis (LRA) is a method proposed by Turney [10] for measuring the similarity in the semantic relations between two pairs of words. In LRA, SVD has been applied to pair-pattern matrix to represent a latent feature space. Although LRA achieve satisfied result for answering the 374 SAT questions (56.1%), it is complex process to factorize a huge matrix and thus it is time-consuming method (requires 9 days to run).

On the other hand, many feature selection methods have been proposed in the literature. Selecting important features using classification approach has been used for different NLP tasks such as sentiment analysis [11] and text classification [12, 13]. Given a number of examples for specific task, linear classifier ables to recover the features that are relevant to separate the examples into classes. For example, in text classification a documents are represented by words in the vocabulary which suffer from the curse of dimensionality. A linear classifier generates coefficients of the features in the space which are used to rank the most informative words that helps in separating documents into categories.

For sentence-level similarity, Ji and Eisenstein [14] apply data-driven approach for weighting the features for paraphrase classification task. Based on supervised (labeled) dataset, they propose new weighting metric for features in order to distinguish the deterministic features for sentence semantics. The weighting metric uses KL Divergence to weight the distributional features in the co-occurrence matrix for sentences before decomposing process. They report significant improvement on sentence similarity in comparison with other works.

Another approach to select a subset of informative feature is using mutual information based methodology. PMI statistical weighting method has been applied for feature selection for document categorisation [15, 16]. It calculates the amount of information that a feature includes about a specific categories.

Xu et al. [15] show that MI is not efficient approach to select relevant feature for text classification compared with other known approach such as Document Frequency (DF) and Information Gain (IG).

While there are efforts spent for feature selection for many NLP tasks, only few attentions have been directed to relational similarity between two pairs of words. Turney [1] heuristically identify a space for semantic relations called function space which consist of verb patterns. For example, for an analogy (*word, language*), (*note, music*), *word* and *note* share the same function, e.g. the function of building units (*vocabularies*). Similarity, *language* and *music* share the same function, the function of *communications*. To the best of our knowledge, there is no work yet on feature selection data-driven methods for relational similarity task. Consequently, this paper contribute to handle that issue.

3 Relational Similarity in Feature Space

Let us consider a feature x in some feature space \mathcal{S} . We do not impose any constraints on the type of features here, and the proposed method can handle any type of features that can be used to represent a word such as other words that co-occur with a target word in the corpus (lexical features), or their syntactic categories such as part-of-speech (POS) (syntactic features). The feature space \mathcal{S} is defined as the set containing all features we extract for all target words. We represent the salience of x in \mathcal{S} by the discriminative weight $w(x, \mathcal{S}) \in \mathbb{R}$. For example, if x is a representative feature of \mathcal{S} , then it will have a high $w(x, \mathcal{S})$. The concept of a discriminative weight can be seen as a feature selection method. If a particular feature is not a good representative of the space, then it will receive a small (ideally zero) weight, thereby effectively pruning out the feature from the space.

Given the above setting, the task of discovering relational feature spaces can be modelled as a problem of computing the discriminative weights for features. We use $\phi(A)$ to denote the set of non-zero features that co-occur with the word A . The salience $f(A, x, \mathcal{S})$ of x as a feature of A in \mathcal{S} is defined as:

$$f(A, x, \mathcal{S}) = h(A, x) \times w(x, \mathcal{S}) \quad (1)$$

Here, $h(A, x) \geq 0$ is the strength of association between A and x , and can be computed using any non-negative feature co-occurrence measure. In our experiments we use positive pointwise mutual information (PPMI) computed using corpus counts as $h(A, x)$.

(1) is analogous to the tf-idf score used in information retrieval in the sense that $h(A, x)$ corresponds to the term-frequency (tf) (i.e. how significant is the presence of x as a feature in A), and $w(x, \mathcal{S})$ corresponds to the document-frequency (df) (i.e. what is the importance of x as a feature in the space \mathcal{S}). The similarity, $\text{sim}_{\mathcal{S}}(A, C)$ between two words A and C in \mathcal{S} can then be defined as in (2) which is the sum of pointwise products over the intersection of the feature sets $\phi(A)$ and $\phi(C)$.

$$\text{sim}_{\mathcal{S}}(A, C) = \sum_{x \in \phi(A) \cap \phi(C)} f(A, x, \mathcal{S}) f(C, x, \mathcal{S}) \quad (2)$$

Moreover, by substituting (1) in (2) we get:

$$\text{sim}_{\mathcal{S}}(A, C) = \sum_{x \in \phi(A) \cap \phi(C)} h(A, x)h(C, x)w(x, \mathcal{S})^2 \quad (3)$$

Following the proposal by [1], we can then compute the relational similarity, $\text{sim}_{\text{rel}}((A, B), (C, D))$, between two word-pairs (A, B) and (C, D) as the geometric mean of their functional similarities:

$$\text{sim}_{\text{rel}}((A, B), (C, D)) = \sqrt{\text{sim}_{\mathcal{S}}(A, C) \times \text{sim}_{\mathcal{S}}(B, D)} \quad (4)$$

4 Learning Features Weights

The relational similarity measure described in Sect. 3 depends on the feature space \mathcal{S} via the discriminative weights $w(x, \mathcal{S})$ assigned to each feature x . Therefore, our goal of discovering a representative feature space from data can be seen as a problem of learning $w(x, \mathcal{S})$. We propose a supervised classification-based approach for computing discriminative weights using labeled dataset.

Let us denote a labeled dataset consists of word-pairs (A, B) and (C, D) annotated for $l = 1$ (i.e. the two word pairs are analogous) or $l = 0$ (otherwise). Here, $l \in \{0, 1\}$ denotes the class label. From (12) and (3), we see that for two analogous word-pairs, (A, B) and (C, D) , their relational similarity increases if the two products $h(A, x)h(C, x)$ and $h(B, x)h(D, x)$ increase. Following this observation, we define a feature x to appear in an instance word-pairs (A, B) and (C, D) iff:

$$(x \in \phi(A) \cap \phi(C)) \vee (x \in \phi(B) \cap \phi(D)) \quad (5)$$

4.1 Linear Classifier Method for Relational Feature Ranking

For the proposed classification-based approach, each positive instance word pairs $((A, B), (C, D))$ or negative word-pairs $((A', B'), (C', D'))$ have a corresponding feature vector in \mathcal{S} , such that the entry for x in the $(A, B), (C, D)$ positive instance is defined as follows:

$$g(((A, B), (C, D)), x) = \mathcal{I}[x \in \phi(A) \cap \phi(C)] + \mathcal{I}[x \in \phi(B) \cap \phi(D)] \quad (6)$$

Here, $g(((A, B), (C, D)), x)$ denotes the value of feature x in the feature vector representing the instance $((A, B), (C, D))$, and \mathcal{I} is the indicator function which return 1 if the expression evaluated is true, or 0 otherwise. Likewise for a negative instance. We train a linear-SVM binary classifier to learn a weight for each feature in the feature space. $w(x, \mathcal{S})$ can be interpreted as the confidence of the feature as an indicator of the strength of analogy (relational similarity) between (A, B) and (C, D) . The absolute value of a weight of a feature can be considered as a measure of the importance of that feature when discriminating the two classes in a binary linear classifier. Therefore, we rank the features in the space according to the absolute value of the weights $|w(x, \mathcal{S})|$. Only linearised kernel classifier explicitly associates weights to individual features. Therefore,

this approach is restricted to linear kernel. In the case of non-linear kernels such as polynomial kernels that can be expanded prior to learning to all feature combinations considered in the kernel computation, we can still apply this technique to identify salient feature combinations. However, we limit the discussion in this paper to finding relational feature spaces consisting of individual features and defer the study of salient feature combinations for relational similarity measurement to future work.

The proposed method is compared against baseline methods namely: KL and PMI in addition to random selection and heuristic verb space. KL and PMI methods also require labelled data as in the proposed classification-based approach.

4.2 KL Divergence-Based Ranking Approach

We consider KL divergence-based weighting approach proposed by [14] to compute $w(x, \mathcal{S})$ for relational similarity measurement. For this purpose, we will consider the two distributions for each feature x in \mathcal{S} -space namely, $p(x)$ and $q(x)$ where $p(x)$ is computed for analogous $((A, B), (C, D))$, while $q(x)$ is taken over the unrelated pairs of words $((A', B'), (C', D'))$. $p(x) = P(x \in \phi(A) | x \in \phi(C), l = 1 \text{ or } x \in \phi(B) | x \in \phi(D), l = 1)$. Similarly, $q(x) = P(x \in \phi(A') | x \in \phi(C'), l = 0 \text{ or } x \in \phi(B') | x \in \phi(D'), l = 0)$.

Specifically, we compute the probability $p(x)$ of a feature x being an indicator of the analogous class as follows:

$$\frac{1}{Z_p(x)} \sum_{(A,B),(C,D) \in \mathcal{N}_+} g(((A, B), (C, D)), x) \quad (7)$$

Here, \mathcal{N}_+ is the set of positive word-pairs, and the normalisation coefficient $Z_p(x)$ satisfies, $\sum_{x \in \mathcal{S}} p(x) = 1$. Likewise, we can compute $q(x)$, the probability of a feature x being an indicator of the negative (relationally dissimilar) class using the features occurrences in negative instances $((A', B'), (C', D'))$ as follows:

$$\frac{1}{Z_q(x)} \sum_{(A',B'),(C',D') \in \mathcal{N}_-} g(((A', B'), (C', D')), x) \quad (8)$$

Here, \mathcal{N}_- is the set of negative word-pairs, and the normalization coefficient $Z_q(x)$ satisfies, $\sum_x q(x) = 1$. Having computed $p(x)$ and $q(x)$, we then compute $w(x, \mathcal{S})$ as the KL divergence between the two distributions as,

$$w(x, \mathcal{S}) = p(x) \log \left(\frac{p(x)}{q(x)} \right). \quad (9)$$

4.3 PMI Ranking Approach

PMI is used to weight a feature x such that:

$$w(x, \mathcal{S}) = \text{PMI}(x, \mathcal{N}_+) - \text{PMI}(x, \mathcal{N}_-) \quad (10)$$

Where $\text{PMI}(x, \mathcal{N}_+)$ measures the association between a feature x with analogues word-pairs, whereas $\text{PMI}(x, \mathcal{N}_-)$ indicates the co-occurrence of a feature with relationally dissimilar pairs. PMI has been computed as follows:

$$\text{PMI}(x, \mathcal{N}_+) = \log \left(\frac{h(x, \mathcal{N}_+)}{h(x, \mathcal{N})|\mathcal{N}_+|} |\mathcal{N}| \right) \quad (11)$$

$$\mathcal{N} = \mathcal{N}_+ \cup \mathcal{N}_-$$

Here \mathcal{N} is the union set of the positive and negative word-pairs and $h(x, \mathcal{N}_+)$ is summed for all analogous pairs:

$$\sum_{(A,B),(C,D) \in \mathcal{N}_+} g(((A, B), (C, D)), x)$$

Similarly, $h(x, \mathcal{N}_-)$ is calculated considering negative instances in the dataset.

We rank the features according to the absolute values of their weights by each of the methods described to define the representative space to measure the relational similarity. The relational similarity between two given word pairs is computed as follows after reducing the word representations to the top ranked feature space:

$$\text{sim}_{\text{rel}}((A, B), (C, D)) = \sqrt{\text{sim}(A, C) \times \text{sim}(B, D)} \quad (12)$$

Cosine similarity between two vectors is defined as follows:

$$\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} \quad (13)$$

We experimented using both unnormalised word embeddings as well as ℓ_2 normalised word representations. We found that ℓ_2 normalised word representations perform better than the unnormalised version in most configurations. Consequently, we report results obtained only with the ℓ_2 normalised word representations in the remainder of the paper.

5 Experimental Design

5.1 Dataset

The above mentioned feature selection methods require a labelled dataset of word-pairs for a particular relation type. To generate such a dataset we use the following procedure. We used the DIFFVECS dataset proposed by Vylomova et al. [17] that consists of triples $\langle w_1, w_2, r \rangle$, where word w_1 and w_2 are connected by a relation r ¹. This dataset consists of 15 relation types, we include the relation types for which we have efficient number of pairs to generate the dataset. Consequently, 7 semantic relation types and their sub categories have been considered in this study as presented in Table 1.

¹ <https://github.com/ivri/DiffVec>.

Table 1. Statistic of the dataset used in this study.

Relation type	Sub-relations	Example of positive instance	No. of Pos instances	No. of testing pairs
Hypernym	–	(<i>colour : green</i>) (<i>tool : knife</i>)	1,100	57
Meronym	–	(<i>dishwasher : door</i>) (<i>tiger : mouth</i>)	1,100	57
Event (objects action)	–	(<i>arrive : train</i>) (<i>fix : oven</i>)	1,100	57
Cause-Purpose	Enabling-Agent: Object, Cause: Effect, Agent: Goal, Prevention	(<i>eating : fullness</i>) (<i>illness : discomfort</i>)	1,149	56
Space-Time	Item: Location, Location: Process	(<i>library : reading</i>) (<i>park : playing</i>)	1,435	56
Reference	Plan, Sign: Significant Expression, Representation	(<i>red : stop</i>) (<i>warning : trouble</i>)	1,047	54
Attribute	Object: TypicalAc- tion(noun.verb) ObjectState (noun.noun)	(<i>musician : sing</i>) (<i>tree : grow</i>)	256	30
Total	–	–	7,187	367

For each relation, we exclude some pairs of words for testing the methods, in total we have 367 testing pairs distributed among the relations. We generate positive training instances by pairing word-pairs that have same relation types (considering sub-relations), resulting in 7,187 positive instances from this procedure. Next, we randomly pair a word-pair from a relation r with a word-pair from a relation r' such that $r \neq r'$ to create a pseudo-negative training dataset that has approximately an equal number of instances as that in the positive training dataset (i.e., 7,000).

5.2 Evaluation Measures

During evaluation, we consider the problem of classifying a given pair of words (w_1, w_2) to a specific relation r in a predefined set of relations \mathcal{R} according to the relation that exists between w_1 and w_2 . We measure the relational similarity between a given pair and all the remaining pairs in the testing data. Then, we perform 1-NN relation classification such that if the 1-NN has the same relation label as the target pair, then we consider it to be a correct match. Macro-averaged classification accuracy is used as the evaluation measure. We use the

PPMI matrix from Turney [18], which contains PPMI values between a word and unigrams from the left and right contexts of that word in a corpus². The total number of features extracted ($|\mathcal{S}|$) is 139,246.

5.3 Results

For a classification method, we train linear SVM using scikit-learn library³. We use 5 folds cross-validation to find the optimal value of penalty parameter C of the error term. Following Turney [1], we used verbs as \mathcal{S} to evaluate the performance of the functional space for measuring relational similarity. We used the NLTK POS tagger⁴ for identifying verbs in the feature space. The verb space identified by the POS tagger contains 12k verbs.

In Table 2, we compare the feature weighting methods discussed in Sect. 4 for different semantic relation types used in the evaluated dataset (illustrated in Table 1). The accuracies for SVM-based, KL, PMI and random ranking methods are reported for the top 1k features. For verb-space, the results indicate the performance of the 12k verbs in the feature space. Classification approach of weighting features and verb-space perform equally for hypernym relation. For meronym, event and attribute relation types the proposed linear-SVM outperforms other methods of feature ranking. KL divergence-based method shows its ability to perform well compared with other methods for cause-purpose and space-time relations. Among different relation types compared in Table 2, classification-based weighting method reports the highest macro-average accuracy compared with other baselines. The fact that the proposed method could improve the performance for many relations of relational classification task empirically justifies our proposal for a data-driven approach for feature selection for relational similarity measurement.

Table 2. Accuracy per relation type for the top 1000 ranked features.

Relation	Classifier	KL	PMI	Verb-space	Random
Hypernym	73.68	71.93	56.14	73.68	54.39
Meronym	70.18	68.42	45.61	61.4	56.14
Event	78.95	73.68	29.82	66.67	54.39
Attribute	33.33	13.33	30.00	23.33	10.00
Cause-Purpose	41.07	44.64	28.57	37.50	21.43
Space-Time	58.93	64.29	33.93	62.5	46.43
Reference	57.41	59.26	42.59	64.81	33.33
Macro-average	59.08	56.51	38.10	55.7	39.44

² The corpus was collected by Charles Clarke at the University of Waterloo.

³ <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.

⁴ <http://www.nltk.org>.

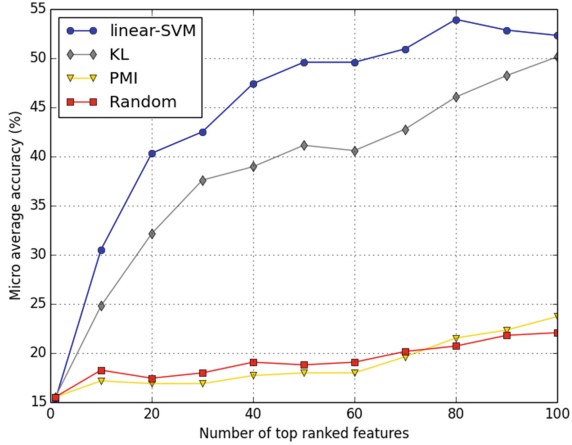


Fig. 1. Cumulative evaluation of feature weighting methods.

We evaluate which of the ranking methods ranks the relational features at the top of the weighted feature list. Figure 1 shows the micro-average accuracies of the top-ranked features selected by the different methods, verb-space is not included in this comparison as it is not a ranking method for feature selection. We start by evaluating the top ranked feature, subsequently adding 10 more features at a time. The random baseline randomly selects a subset of features from \mathcal{S} . As shown in the Fig. 1, the top-weighted features using the proposed linear SVM-based approach outperforms all other methods for relational similarity measurement. The proposed method statistically significantly outperforms (according to McNemar test with $p < 0.05$) all other methods for ranking the most informative features in the top ranked feature list. This indicates that the effective features for measuring relational similarity are indeed ranked at the top by the proposed method. In addition, our results show that it is possible to maintain a relational classification accuracy while using only small subset of the features (top 100 features). KL divergence-based ranking method follow classification approach for ranking the best features for relational similarity. However, PMI method performs badly as it gives accuracies comparable with the random feature selection method. PMI is known to give higher values to rare features thereby preferring rare features. We believe this might be an issue when selecting features for representing word-pairs.

6 Conclusion

We proposed the first-ever method for discovering a discriminative feature space for measuring relational similarity from data. The relational classification results show that using labeled data to train a linear classifier for feature selection can improve the feature space for relational similarity measurement. The proposed method outperforms KL and PMI methods for discovering relational

feature space. Using PMI to discover relational features has been demonstrated to have relatively poor performance, a finding which is consistent with previous work for text classification task [15]. In addition, classification-based weighting method reports better performance for many relation types compared with the functional verb space. Future researches can be carried out to improve the feature space for relational similarity task by incorporating verb space with the data-driven discovered features.

References

1. Turney, P.D.: Domain and function: a dual-space model of semantic relations and compositions. *J. Artif. Intell. Res.* **44**, 533–585 (2012)
2. Bollegala, D., Matsuo, Y., Ishizuka, M.: A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web. In: *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 803–812 (2009)
3. Turney, P.D.: A uniform approach to analogies, synonyms, antonyms, and associations. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 905–912 (2008)
4. Nakov, P., Kozareva, Z.: Combining relational and attributional similarity for semantic relation classification. In: *Proceedings of the Recent Advances in Natural Language Processing*, pp. 323–330 (2011)
5. Duc, N.T., Bollegala, D., Ishizuka, M.: Using relational similarity between word pairs for latent relational search on the web. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 196–199 (2010)
6. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 74–84 (2013)
7. Turney, P.D.: Similarity of semantic relations. *Comput. Linguist.* **32**(3), 379–416 (2006)
8. Turney, P.D.: Distributional semantics beyond words: supervised learning of analogy and paraphrase. *Trans. Assoc. Computat. Linguist.* **1**, 353–366 (2013)
9. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. *Discourse Process.* **25**(2–3), 259–284 (1998)
10. Turney, P.D.: Measuring semantic similarity by latent relational analysis. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1136–1141 (2005). arXiv preprint [arXiv:cs/0508053](https://arxiv.org/abs/cs/0508053)
11. Tripathi, G., Naganna, S.: Feature selection and classification approach for sentiment analysis. *Mach. Learn. Appl. Int. J.* **2**(2), 1–16 (2015)
12. Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Feature selection using support vector machines. *WIT Trans. Inf. Commun. Technol.* **28** (2002)
13. Mladenić, D., Brank, J., Grobelnik, M., Milic-Frayling, N.: Feature selection using linear classifier weights: interaction with classification models. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 234–241. ACM (2004)

14. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the Empirical Methods in Natural Language Processing, pp. 891–896 (2013)
15. Xu, Y., Jones, G.J., Li, J., Wang, B., Sun, C.: A study on mutual information-based feature selection for text categorization. *J. Comput. Inf. Syst.* **3**(3), 1007–1012 (2007)
16. Schneider, K.-M.: Weighted average pointwise mutual information for feature selection in text categorization. In: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 252–263. Springer, Heidelberg (2005). https://doi.org/10.1007/11564126_27
17. Vylomova, E., Rimmel, L., Cohn, T., Baldwin, T.: Take and took, gaggle and goose, book and read: evaluating the utility of vector differences for lexical relation learning. In: Proceedings of the Association for Computational Linguistics, pp. 1671–1682 (2016)
18. Turney, P.D., Neuman, Y., Assaf, D., Cohen, Y.: Literal and metaphorical sense identification through concrete and abstract context. In: Proceedings of the Empirical Methods in Natural Language Processing, pp. 27–31 (2011)