A Relational Model of Semantic Similarity

Danushka Bollegala Yutaka Matsuo Mitsuru Ishizuka

The University of Tokyo

Semantic similarity is a central concept that extends across numerous fields such as artificial intelligence, natural language processing, cognitive science and psychology. Accurate measurement of semantic similarity between words is essential for various tasks such as, document clustering, information retrieval and synonym extraction. We propose a novel model of semantic similarity using the semantic relations that exist among words. Given two words, first, we represent the semantic relations that hold between those words using automatically extracted lexical pattern clusters. Next, the semantic similarity between the two words is computed using a Mahalanobis distance measure. The proposed similarity measure reports a high correlation with human ratings in Miller-Charles benchmark dataset.

1. Introduction

Similarity is a fundamental concept in theories of knowledge and behavior. Psychological experiments have shown that similarity acts as an organizing principle by which individuals classify objects, and make generalizations [9]. For example, a biologist would classify a newly found animal specimen based upon the properties that it shares with existing categories of animals. We can then make additional inferences on the new specimen using the properties known for the existing category. As the similarity between two objects X and Y increases, so does the probability of correctly inferring that Y has the property T upon knowing that X has T [20]. Accurate measurement of semantic similarity between lexical units such as words or phrases is important for numerous tasks in natural language processing such as word sense disambiguation [17], synonym extraction [14], and automatic thesauri generation [6]. In information retrieval, similar or related words are used to expand user queries to improve recall [18].

Semantic similarity is a context dependent and dynamic phenomenon. New words are constantly being created and existing words are assigned with new senses on the Web. To decide whether two words are semantically similar, it is important to know the semantic relations that hold between the words. For example, the words *horse* and *cow* can be considered semantically similar because both horses and cows are useful animals in agriculture. Similarly, a *horse* and a *car* can be considered semantically similar because cars and historically horses are used for transportation. Semantic relations such as ***X*** *and* ***Y*** *are used in agriculture*, or ***X*** *and* ***Y*** *are used for transportation* exist between two words *X* and *Y* in these examples. We use bold-italics, ***X***, to denote the slot of a word *X* in a lexical pattern.

We propose a *relational model* to compute the semantic similarity between two words. First, using snippets retrieved from a web search engine, we present an automatic lexical pattern extraction algorithm to represent the semantic relations that exist between two words. For example, given two words *ostrich* and *bird*, we extract ***X*** *is a* ***Y***, ***X*** *is a large* ***Y***, and ***X*** *is a flightless* ***Y*** from the Web. Using a set of semantically related words as training data, we evaluate the confidence of a lexical pattern as an indicator of semantic similarity. For example, the pattern ***X*** *is a* ***Y*** is a better indicator of semantic similarity between *X* and *Y* than the pattern ***X*** *and* ***Y***. Consequently, we would like to emphasize the former pattern by assigning it a higher confidence score. It is noteworthy that all lexical patterns are not independent – multiple lexical patterns can express the same semantic relation. For example, the pattern ***X*** *is a large* ***Y*** subsumes the more general pattern ***X*** *is a* ***Y*** and they both indicate a hypernymic relationship between *X* and *Y*. By clustering the semantically related patterns into groups, we can both overcome the data sparseness problem and reduce the number of parameters in training. To identify semantically related patterns, we propose a sequential pattern clustering algorithm using the distributional hypothesis [10]. We represent two words by a feature vector defined over the clusters of patterns. Finally, the semantic similarity is computed as Mahalanobis distance between points corresponding to the feature vectors. By using Mahalanobis distance instead of Euclidean distance, we can account for the inter-dependence between semantic relations. The proposed method outperforms all web-based semantic similarity measures on Miller-Charles [15] benchmark dataset.

2. Related Work

Geometric models, such as multi-dimensional scaling has been used in psychological experiments analyzing the properties of similarity [13]. These models represent objects as points in some coordinate space such that the observed dissimilarities between objects correspond to the metric distances between the respective points. Geometric models assume that objects can be adequately represented as points in some coordinate space and that dissimilarity behaves like a metric distance function satisfying minimality, symmetry and triangle inequality assumptions. However, both dimensional and metric assumptions are open to question.

Tversky [21] proposed the *contrast model* of similarity to overcome the problems in geometric models. The contrast model relies on featural representation of objects, and it is used to compute the similarity between the representations of two objects. Similarity is defined as an increasing function of common features (i.e. features in common to the two objects), and as a decreasing function of distinctive features (i.e. features that apply to one object but not the other). The attributes of objects are primal to contrast model and it does not explicitly incorporate the relations between objects when measuring similarity.

連絡先: 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.
danushka@mi.ci.i.u-tokyo.ac.jp

Given a taxonomy of concepts, a straightforward method to calculate similarity between two words (concepts) is to find the length of the shortest path connecting the two words in the taxonomy [16]. If a word is polysemous (i.e. has more than one sense) then multiple paths might exist between the two words. In such cases, only the shortest path between any two senses of the words is considered for calculating similarity. A problem that is frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent a uniform distance. As a solution to this problem, Schickel-Zuber and Faltings [19] propose ontology structure based similarity (OSS) between two concepts in an ontology as an asymmetric distance function.

Resnik [17] proposed a similarity measure using information content. He defined the similarity between two concepts C_1 and C_2 in the taxonomy as the maximum of the information content of all concepts C that subsume both C_1 and C_2. Then the similarity between two words is defined as the maximum of the similarity between any concepts that the words belong to. He used WordNet as the taxonomy; information content is calculated using the Brown corpus.

3. Relational Model of Similarity

We propose a model to compute the semantic similarity between two words a and b using the set of semantic relations $R(a,b)$ that hold between a and b. We call the proposed model the *relational model* of semantic similarity and it is defined by the following equation,

$$\mathrm{sim}(a,b) = \Xi(R(a,b)). \tag{1}$$

Here, $\mathrm{sim}(a,b)$ is the semantic similarity between the two words a and b, and Ξ is a weighting function defined over the set of semantic relations $R(a,b)$. Given that a particular set of semantic relations are known to hold between two words, the function Ξ expresses our confidence on those words being semantically similar.

A semantic relation can be expressed in a number of ways. For example, given a taxonomy of words such as WordNet, semantic relations (i.e. hypernymy, meronymy, synonymy etc.) between words can be directly looked up in the taxonomy. Alternatively, the labels of the edges in the path connecting two words can be used as semantic relations. However, in this paper we do not assume the availability of manually created resources such as dictionaries or taxonomies. We represent semantic relations using automatically extracted lexical patterns. Lexical patterns have been successfully used to represent various semantic relations between words such as hypernymy [11] and meronymy [1]. Following these previous approaches, we represent $R(a,b)$ as a set of lexical patterns. Moreover, we denote the frequency of a lexical pattern r for a word pair (a,b) by $f(r,a,b)$.

So far we have not defined the functional form of Ξ. A straightforward approach is to use a linearly weighted combination of relations as shown below,

$$\Xi(R(a,b)) = \sum_{r_i \in R(a,b)} w_i \times f(r_i,a,b). \tag{2}$$

Here, w_i is the weight associated with the lexical pattern r_i and can be determined using training data as described later in section 5.. However, this formulation has two fundamental drawbacks. First, the number of weight parameters w_i is equal to the number of lexical patterns. Typically two words can co-occur in numerous patterns. Consequently, we end up with a large number of parameters in the model. Complex models with a large number of parameters are difficult to train because they tend to overfit to the traning data. Second, the linear combination given in Equation 2 assumes the lexical patterns to be mutually independent. However, in practice this is not true. For example, both patterns *X is a Y* and *Y such as X* indicate a hypernymic relation between *X* and *Y*.

To overcome the above mentioned limitations, we first cluster the lexical patterns to identify the semantically related patterns. Our clustering algorithm is detailed in section 3.2. Next, we define Ξ using the formed clusters as follows,

$$\Xi(R(a,b)) = \mathbf{x}_{ab}^t \Lambda \mathbf{x}_{ab}. \tag{3}$$

Here, $\mathbf{x}_{ab}$ is a vector representing the words a and b. The j-th element of $\mathbf{x}_{ab}$ equals to the sum of frequencies of all patterns that belong to cluster c_j (i.e. $\sum_{r \in C_j} f(r,a,b)$). Λ is the inter-cluster correlation matrix. The (i,j) element of matrix Λ denotes the correlation between the two clusters c_i and c_j. Matrix Λ is expected to capture the dependence between semantic relations.

The proposed relational model of semantic similarity differs from feature models of similarity, such as the contrast model [21], in that it is defined over the set of semantic relations that exist between two words instead of the set of features for each word. In fact, modeling similarity as a phenomenon of relations between objects rather than features of individual objects is central to computational models of analogy-making such as the structure mapping theory (SMT) [7]. SMT claims that an analogy is a mapping of knowledge from one domain (base) into another (target) which conveys that a system of relations known to hold in the base also holds in the target. The target objects do not have to resemble their corresponding base objects. During the mapping process, features of individual objects are dropped and only relations are mapped. The proposed relational model of similarity use this relational view of similarity to compute semantic similarity between words.

3.1 Extracting Lexical Patterns

To compute semantic similarity between two words using the relational model (Equation 3), we must first extract the numerous lexical patterns from contexts in which those two words appear. For this purpose, we use the pattern extraction algorithm proposed by Bollegala et al. [3].

3.2 Clustering Lexical Patterns

A semantic relation can be expressed using more than one pattern. By grouping the semantically related patterns, we can both reduce the model complexity in Equation 2, and consider the dependence among semantic relations in Equation 3. We use the distributional hypothesis [10] to find semantically related lexical patterns. The distributional hypothesis states that words that occur in the same context have similar meanings. If two lexical patterns are similarly distributed over a set of word pairs then from the distributional hypothesis it follows that the two patterns must be similar.

We represent a pattern p by a vector $\mathbf{p}$ in which the i-th element is the frequency, $f(a_i,b_i,p)$, of word pair (a_i,b_i) in pattern p. Given a set P of patterns and a similarity threshold θ, Algorithm 1 returns clusters of similar patterns. First, the function $SORT$ sorts

Algorithm 1 Sequential pattern clustering algorithm.

```
Input: patterns $P = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$, threshold $\theta$
Output: clusters $C$
 1: SORT($P$)
 2: $C \leftarrow \{\}$
 3: for pattern $\mathbf{p_i} \in P$ do
 4:     $max \leftarrow -\infty$
 5:     $\mathbf{c}^* \leftarrow null$
 6:     for cluster $\mathbf{c_j} \in C$ do
 7:         $sim \leftarrow \mathrm{cosine}(\mathbf{p_i}, \mathbf{c_j})$
 8:         if $sim > max$ then
 9:             $max \leftarrow sim$
10:             $\mathbf{c}^* \leftarrow \mathbf{c_j}$
11:         end if
12:     end for
13:     if $max \geq \theta$ then
14:         $\mathbf{c}^* \leftarrow \mathbf{c}^* \oplus \mathbf{p_i}$
15:     else
16:         $C \leftarrow C \cup \{\mathbf{p_i}\}$
17:     end if
18: end for
19: return $C$
```

the patterns in the descending order of their total occurrences in all word pairs (i.e., $\sum_i f(a_i, b_i, p)$). Then the outer for-loop (starting at line 3), repeatedly takes a pattern $\mathbf{p_i}$ from the ordered set P, and in the inner for-loop (starting at line 6), finds the cluster, c^* ($\in C$) that is most similar to $\mathbf{p_i}$. Similarity between $\mathbf{p_i}$ and cluster centroid $\mathbf{c_j}$ is computed using cosine similarity. If the maximum similarity exceeds the threshold θ, we append $\mathbf{p_i}$ to $\mathbf{c}^*$ (line 14). Here, the operator $\oplus$ to denotes vector addition. Otherwise we form a new cluster $\{\mathbf{p_i}\}$ and append it to C, the set of clusters. After all patterns are clustered, we compute the (i, j) element of the inter-cluster correlation matrix Λ (Equation 3) as the inner-product between the centroid vectors $\mathbf{c_i}$ and $\mathbf{c_j}$ of the corresponding clusters i and j. The parameter θ ($\in [0, 1]$) determines the *purity* of the formed clusters and is set experimentally in section 5.. Algorithm 1 scales linearly with the number of patterns. Moreover, sorting the patterns by their total word-pair frequency prior to clustering ensures that the final set of clusters contains the most common relations in the dataset.

4. Evaluation Procedure

Evaluating a semantic similarity measure is difficult because the notion of semantic similarity is subjective. Miller-Charles dataset [15] has been frequently used to benchmark semantic similarity measures. Miller-Charles dataset contains 30 word-pairs rated by a group of 38 human subjects. The word-pairs are rated on a scale from 0 (no similarity) to 4 (perfect synonymy). Because of the omission of two word-pairs in earlier versions of WordNet, most researchers had used only 28 pairs for evaluations. The degree of correlation between the human ratings in the benchmark dataset and the similarity scores produced by an automatic semantic similarity measure, can be considered as a measurement of how well the semantic similarity measure captures the notion of semantic similarity held by humans. Following the previous work, we use Miller-Charles dataset to evaluate the proposed semantic similarity measure.

Table 1: Semantic similarity scores on Miller-Charles dataset

Method	Correlation
WebJaccard	0.260
WebDice	0.267
WebOverlap	0.382
WebPMI	0.549
NGD	0.205
SH	0.580
CODC	0.694
SVM	0.834
Proposed	0.867

5. Experiments

To extract lexical patterns that express numerous semantic relations, we first select synonymous words from WordNet synsets. A synset is a set of synonymous words assigned for a particular sense of a word in WordNet. Following Bollegala et al. [2], we randomly select 2000 synsets of nouns from WordNet. From each synset, a pair of synonymous words is selected. For polysemous nouns, we selected synonyms from the dominant sense. To perform a fair evaluation, we do not select any words that appear in the Miller-Charles benchmark dataset. This process yields 2000 synonymous word-pairs. We use YahooBOSS API[*1] and download 1000 snippets for each of those word-pairs. Experimentally, we set the values for the parameters in the pattern extraction algorithm (section 3.1): $L = 5$, $g = 2$, $G = 4$, and extract $5, 238, 637$ unique patterns. However, only $1, 680, 914$ of those patterns occur more than twice. Low frequency patterns often contain misspellings and are not suitable for training. Therefore, we selected patterns that occur at least 10 times in the snippet collection. Moreover, we remove very long patterns (ca. over 20 chars). The final set contains $140, 691$ unique lexical patterns. The remainder of the experiments described in the paper use those patterns.

We use the clustering algorithm 1 to cluster the extracted patterns. The only parameter is algorithm 1, clustering threshold θ, is set to the optimal value using the WordSimilarity-353 collection [8] as training data. This collection contains 353 word-pairs. Each pair has 13-16 human judgments, which were averaged for each pair to produce a single relatedness score. We removed 29 word-pairs from this collection prior to training because those pairs contained at least one word from the Miller-Charles dataset. The optimal value of θ is determined as follows. First, we set θ to a value in the range $[0, 1]$ and use algorithm 1 to produce a set of pattern clusters. Next, we compute the semantic similarity between two words (a, b) using Equation 3. We then compute the similarity scores for all training word-pairs in the WordSimilarity collection, and calculate their Pearson correlation coefficient against human ratings. This procedure is systematically repeated with different values of θ. The maximum correlation (i.e. 0.4722) is obtained for $\theta = 0.85$. We set θ to this optimal value and use Equation 3 to compute similarity scores for the Miller-Charles dataset.

Table 1 compares the proposed method against Miller-Charles ratings (MC), and previously proposed web-based semantic sim-

[*1] http://developer.yahoo.com/search/boss/

ilarity measures: WebJaccard, WebDice, WebOverlap, WebPMI [2], Normalized Google Distance (NGD) [5], Sahami and Heilman (SH) [18], co-occurrence double checking model (CODC) [4], and support vector machine-based (SVM) approach [2]. Pearson correlation coefficient with human ratings are shown in Table 1. From Table 1 we see that measures that use contextual information from snippets (e.g. SH, CODC, SVM, and proposed) outperform the ones that use only co-occurrence statistics (e.g. Jaccard, overlap, Dice, PMI, and NGD) such as page-counts. This is because similarity measures that use contextual information are better equipped to compute the similarity between polysemous words. Although both SVM and proposed methods use lexical patterns, unlike the proposed method, the SVM method does not consider the relatedness between patterns. The superior performance of the proposed method is attributable to its consideration of relatedness of patterns. Despite the fact that the proposed method does not use manually compiled resources such as WordNet for computing similarity, its performance is comparable to similarity measures that use WordNet: Edge-counting (0.664), Jiang & Conrath [12] (0.848), Lin [14] (0.822), Resnik [17] (0.745), and Li et al. [22] (0.891). We believe that the proposed method will be useful to compute the semantic similarity between named-entities for which manually created resources are either incomplete or do not exist.

6. Conclusion

We proposed a relational model to measure the semantic similarity between two words. First, to represent the numerous semantic relations that exist between two words, we extract lexical patterns from snippets retrieved from a web search engine. Second, we cluster the extracted patterns to identify the semantically related patterns. Third, using the pattern clusters we define a feature vector to represent two words and compute the semantic similarity by taking into account the inter-cluster correlation. The proposed method outperformed all existing web-based semantic similarity measures on a benchmark dataset, achieving a statistically significant correlation of 0.867 with human ratings.

References

[1] M. Berland and E. Charniak. Finding parts in very large corpora. In *Proc. of ACL'99*, pages 57–64, 1999.

[2] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proc. of WWW'07*, pages 757–766, 2007.

[3] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *Proc. of WWW'09 (to appear)*, 2009.

[4] H. Chen, M. Lin, and Y. Wei. Novel association measures using web search with double checking. In *Proc. of the COLING/ACL '06*, pages 1009–1016, 2006.

[5] R.L. Cilibrasi and P.M.B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[6] J. Curran. Ensemble menthods for automatic thesaurus extraction. In *Proc. of EMNLP*, 2002.

[7] B. Falkenhainer, K.D. Forbus, and D. Gentner. Structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63, 1989.

[8] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM TOIS*, 20:116–131, 2002.

[9] R. L. Goldstone. The role of similarity in categorization: providing a groundwork. *Cognition*, 52:125–157, 1994.

[10] Z. Harris. Distributional structure. *Word*, 10:146–162, 1954.

[11] M.A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proc. of 14th COLING*, pages 539–545, 1992.

[12] J.J. Jiang and D.W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of ROCLING'98*, 1998.

[13] C. L. Krumhansl. Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, 85:445–463, 1978.

[14] D. Lin. Automatic retreival and clustering of similar words. In *Proc. of the 17th COLING*, pages 768–774, 1998.

[15] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1998.

[16] R. Rada, H. Mili, E. Bichnell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):17–30, 1989.

[17] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of IJCAI'95*, 1995.

[18] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of WWW'06*, 2006.

[19] V. Schickel-Zuber and B. Faltings. Oss: A semantic similarity function based on hierarchical ontologies. In *Proc. of IJCAI'07*, pages 551–556, 2007.

[20] J. B. Tenenbaum. Bayesian modeling of human concept learning. In *NIPS'99*, 1999.

[21] A. Tversky. Features of similarity. *Psychological Review*, 84:327–652, 1977.

[22] D. McLean Y. Li, Zuhair A. Bandar. An approch for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):871–882, 2003.