A Relational Model of Semantic Similarity between Words using Automatically Extracted Lexical Pattern Clusters from the Web

Danushka Bollegala * danushka@mi.ci.i.u-tokyo.ac.jp
Yutaka Matsuo matsuo@biz-model.t.u-tokyo.ac.jp
Mitsuru Ishizuka ishizuka@i.u-tokyo.ac.jp
The University of Tokyo
7-3-1, Hongo, Tokyo, 113-8656, Japan

Abstract

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. We compare the proposed similarity measure against previously proposed semantic similarity measures on Miller-Charles benchmark dataset and WordSimilarity-353 collection. The proposed method outperforms all existing web-based semantic similarity measures, achieving a Pearson correlation coefficient of 0.867 on the Millet-Charles 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 (Goldstone, 1994). 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 (Tenenbaum, 1999). 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 (Resnik, 1995), synonym extraction (Lin, 1998a), and automatic thesauri generation (Curran, 2002). In information retrieval, similar or related words are used to expand user queries to improve recall (Sahami and Heilman, 2006).

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

*Research Fellow of the Japan Society for the Promotion of Science (JSPS)

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 803–812,
Singapore, 6-7 August 2009.

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 during training. To identify semantically related patterns, we use a sequential pattern clustering algorithm that is based on the distributional hypothesis (Harris, 1954). We represent two words by a feature vector defined over the clusters of patterns. Finally, the semantic similarity is computed as the 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.

2 Related Work

Geometric models, such as multi-dimensional scaling has been used in psychological experiments analyzing the properties of similarity (Krumhansl, 1978). 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 (1977) 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.

Hahn et al. (2003) define similarity between two representations as the complexity required to transform one representation into the other. Their model of similarity is based on the *Representational Distortion* theory, which aims to provide a theoretical framework of similarity judgments. Their experiments using pattern sequences and geometric shapes show an inverse correlation between the number of transformations required to convert one pattern (or shape) to another, and the perceived similarity ratings by human subjects. How to represent an object, which transformations are allowed on a representation, and how to measure the complexity of a transformation, are all important decisions in the transformational model of similarity. Although distance measures such as edit distance have been used to find approximate matches in a dictionary, it is not obvious how to compute semantic similarity between words using representational distortion theory.

Given a taxonomy of concepts, a straightforward method to calculate similarity between two words (or concepts) is to find the length of the shortest path connecting the two words in the taxonomy (Rada et al., 1989). 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 (2007) propose ontology structure based similarity (OSS) between two concepts in an ontology, which is an asymmetric distance function.

Resnik (1995) 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.

Li et al., (2003) combined structural seman-

tic information from a lexical taxonomy, and information content from a corpus, in a nonlinear model. They proposed a similarity measure that uses shortest path length, depth and local density in a taxonomy. Their experiments reported a Pearson correlation coefficient of 0.8914 on the Miller-Charles benchmark dataset (Miller and Charles, 1998). Lin (1998b) defined the similarity between two concepts as the information that is in common to both concepts and the information contained in each individual concept.

Cilibrasi and Vitanyi (2007) proposed a distance metric between words using page-counts retrieved from a web search engine. The proposed metric is named *Normalized Google Distance* (NGD) and is defined as the normalized information distance (Li et al., 2004) between two strings. They evaluate NGD in a word classification task. Unfortunately NGD only uses page-counts of words and ignores the context in which the words appear. Therefore, it produces inaccurate similarity scores when one or both words between which similarity is computed are polysemous.

Sahami and Heilman (2006) measured semantic similarity between two queries using snippets returned for those queries by a search engine. For each query, they collect snippets from a search engine and represent each snippet as a TF-IDF-weighted term vector. Each vector is L_2 normalized and the centroid of the set of vectors is computed. Semantic similarity between two queries is then defined as the inner product between the corresponding centroid vectors. They did not compare their similarity measure with taxonomy-based similarity measures.

Chen et al., (2006) propose a web-based double-checking model to compute the semantic similarity between words. For two words X and Y, they collect snippets for each word from a web search engine. Then they count the number of occurrences of X in the snippets for Y, and Y in the snippets for X. The two values are combined nonlinearly to compute the similarity between X and Y. This method heavily depends on the search engine's ranking algorithm. Although two words X and Y may be very similar, there is no reason to believe that one can find Y in the snippets for X, or vice versa. This observation is confirmed by the experimental results in their paper which reports 0 similarity scores for many pairs of words in the Miller-Charles dataset.

In our previous work (Bollegala et al., 2007), we proposed a semantic similarity measure using page counts and snippets retrieved from a Web search engine. To compute the similarity between two words X and Y, we queried a web search engine using the query *X AND Y* and extract lexical patterns that combine X and Y from snippets. A feature vector is formed using frequencies of 200 lexical patterns in snippets and four co-occurrence measures: Dice coefficient, overlap coefficient, Jaccard coefficient and pointwise mutual information. We trained a two-class support vector machine using automatically selected synonymous and non-synonymous word pairs from WordNet. This method reports a Pearson correlation coefficient of 0.837 with Miller-Charles ratings. However, it does not consider the relatedness between patterns.

Gabrilovich and Markovitch (2007) represent words using weighted vectors of Wikipedia-based concepts, and define the similarity between words as the cosine of the angle between the corresponding vectors. Their method can be used to compute similarity between words as well as between texts. Although Wikipedia is growing in popularity, not all concepts found on the Web have articles in Wikipedia. Specially, novel or not very popular concepts are not adequately covered by Wikipedia. Moreover, their method requires the concepts to be independent. For non-independent, hierarchical taxonomies such as open directory project (ODP)[1], their method produces suboptimal results.

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,

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

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.

[1] http://www.dmoz.org

A semantic relation can be expressed in a number of ways. For example, given a taxonomy of words such as the 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 (Hearst, 1992), and meronymy (Berland and Charniak, 1999). 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,

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

Here, w_i is the weight associated with the lexical pattern r_i and can be determined using training data. 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 training 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,

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

Here, $\mathbf{x}_{ab}$ is a feature vector representing the words a and b. Each formed cluster contributes a feature in vector $\mathbf{x}_{ab}$ as described later in Section 5. The vector σ is a prototypical vector representing synonymous word pairs. We compute σ as the centroid of feature vectors representing synonymous word pairs. Λ is the inter-cluster correlation matrix. The (i, j)-th 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. Intuitively, if two clusters i and j are highly correlated, then the (i, j)-th element of Λ will be closer to 1. Equation 3 computes the similarity between a word pair (a, b) and a set of synonymous word pairs. Intuitively, if the relations that exist between a and b are typical relations that hold between synonymous word pairs, then Equation 3 returns a high similarity score for a and b.

The proposed relational model of semantic similarity differs from feature models of similarity, such as the contrast model (Tversky, 1977), 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. Specifically, in contrast model, the similarity $S(a, b)$ between two objects a and b is defined in terms of the features common to a and b, $A \cap B$, the features that are distinctive to a, $A - B$, and the features that are distinctive to b, $B - A$. The contrast model is formalized in the following equation,

$$S(a,b) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A). \quad (4)$$

Here, the function f measures the salience of a particular set of features, and non-negative parameters α, β, and θ determine the relative weights assigned to the different components. However, in the relational model of similarity we do not focus on features of individual words but on relations between two words.

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) (Falkenhainer et al., 1989). 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 uses this relational view

Ostrich, a large, flightless bird that lives in the dry grasslands of Africa.

Figure 1: A snippet returned for the query *"ostrich * * * * * bird"*.

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 propose a pattern extraction algorithm using snippets retrieved from a web search engine. The proposed method requires no language-dependent preprocessing such as part-of-speech tagging or dependency parsing, which can be both time consuming at Web scale, and likely to produce incorrect results because of the fragmented and ill-formed snippets.

Given two words a and b, we query a web search engine using the wildcard query *"a * * * * * b"* and download snippets. The "*" operator matches one word or none in a web page. Therefore, our wildcard query retrieves snippets in which a and b appear within a window of seven words. We attempt to approximate the local context of two words using wildcard queries. For example, Figure 1 shows a snippet retrieved for the query *"ostrich * * * * * bird"*.

For a snippet S, retrieved for a word pair (a, b), first, we replace the two words a and b, respectively, with two variables *X* and *Y*. We replace all numeric values by D, a marker for digits. Next, we generate all subsequences of words from S that satisfy all of the following conditions.

(i). A subsequence must contain exactly one occurrence of each *X* and *Y*

(ii). The maximum length of a subsequence is L words.

(iii). A subsequence is allowed to have gaps. However, we do not allow gaps of more than g number of words. Moreover, the total length of all gaps in a subsequence should not exceed G words.

(iv). We expand all negation contractions in a context. For example, *didn't* is expanded to *did not*. We do not skip the word *not* when generating subsequences. For example, this condition ensures that from the snippet *X is not a Y*, we do not produce the subsequence *X is a Y*.

Finally, we count the frequency of all generated subsequences and only use subsequences that occur more than N times as lexical patterns.

The parameters *L, g, G* and *N* are set experimentally, as explained later in Section 6. It is noteworthy that the proposed pattern extraction algorithm considers all the words in a snippet, and is *not* limited to extracting patterns only from the mid-fix (i.e., the portion of text in a snippet that appears between the queried words). Moreover, the consideration of gaps enables us to capture relations between distant words in a snippet. We use a modified version of the *prefixspan* algorithm (Pei et al., 2004) to generate subsequences from a text snippet. Specifically, we use the constraints (ii)-(iv) to prune the search space of candidate subsequences. For example, if a subsequence has reached the maximum length L, or contains the maximum number of gaps G, then we will not extend it further. By pruning the search space, we can speed up the pattern generation process. However, none of these modifications affect the accuracy of the proposed semantic similarity measure because the modified version of the prefixspan algorithm still generates the exact set of patterns that we would obtain if we used the original prefixspan algorithm (i.e. without pruning) and subsequently remove patterns that violate the above mentioned constraints. For example, some patterns extracted form the snippet shown in Figure 1 are: ***X**, a large **Y***, ***X** a flightless **Y***, and ***X**, large **Y** lives*.

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 (Harris, 1954) 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 p in a word pair (a_i, b_i). Given a set P of patterns and a similarity threshold θ, Algorithm 1 returns clusters of similar patterns. First, the function $SORT$ sorts the patterns in the descending order of their total occurrences in all word pairs. The total occurrences of a pattern p is defined as $\mu(p)$, and is given by,

$$\mu(p) = \sum_{(a,b) \in W} f(a, b, p). \quad (5)$$

Here, W is the set of word pairs. 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 the cluster centroid $\mathbf{c_j}$ is computed using cosine similarity. The centroid vector $\mathbf{c}_j$ of cluster c_j is defined as the vector sum of all pattern vectors for patterns in that cluster (i.e. $\mathbf{c}_j = \sum_{p \in c_j} \mathbf{p}$). If the maximum similarity exceeds the threshold θ, we append $\mathbf{p_i}$ to $\mathbf{c}^*$ (line 14). Here, the operator $\oplus$ 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.

Algorithm 1 Sequential pattern clustering algorithm.

Input: patterns $P = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$, threshold θ
Output: clusters C

```
1:  SORT(P)
2:  C ← {}
3:  for pattern p_i ∈ P do
4:      max ← −∞
5:      c* ← null
6:      for cluster c_j ∈ C do
7:          sim ← cosine(p_i, c_j)
8:          if sim > max then
9:              max ← sim
10:             c* ← c_j
11:         end if
12:     end for
13:     if max ≥ θ then
14:         c* ← c* ⊕ p_i
15:     else
16:         C ← C ∪ {p_i}
17:     end if
18: end for
19: return C
```

4 Evaluation Procedure

Evaluating a semantic similarity measure is difficult because the notion of semantic similarity is subjective. Miller-Charles (1998) dataset 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. In addition to Miller-Charles dataset we also evaluate on the WordSimilarity-353 (Finkelstein et al., 2002) dataset. In contrast to Miller-Charles dataset which has only 30 word pairs, WordSimilarity-353 dataset contains 353 word pairs. Each pair has 13-16 human judgments, which were averaged for each pair to produce a single relatedness score. Following the previous work, we use both Miller-Charles dataset and WordSimilarity-353 dataset to evaluate the proposed semantic similarity measure.

5 Computing Semantic Similarity

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. 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 dataset or the WordSimilarity-353

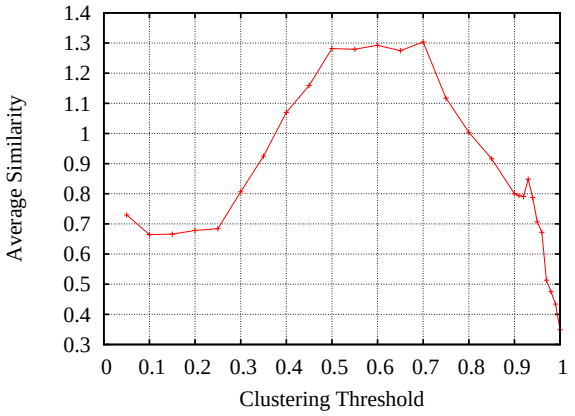

Figure 2: Average similarity vs. clustering threshold θ

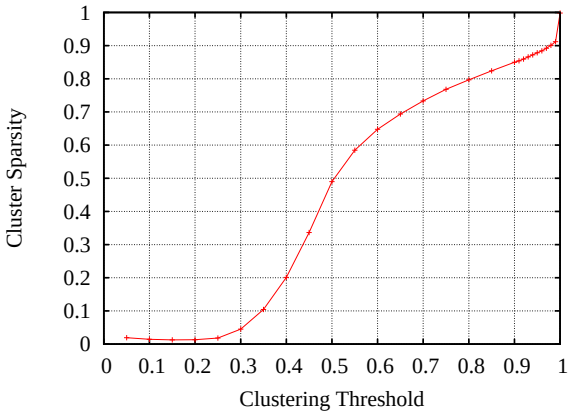

Figure 3: Sparsity vs. clustering threshold θ

dataset, which are used later for evaluation purposes. As we describe later, the clustering threshold θ is tuned using this set of 2000 word pairs selected from the WordNet.

We use the YahooBOSS API[2] 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 characters). The final set contains $140,691$ unique lexical patterns. The remainder of the experiments described in the paper use those patterns.

[2]http://developer.yahoo.com/search/boss/

We use the clustering Algorithm 1 to cluster the extracted patterns. The only parameter in Algorithm 1, the clustering threshold θ, is set as follows. We vary the value of theta θ from 0 to 1, and use Algorithm 1 to cluster the extracted set of patterns. We use the resultant set of clusters to represent a word pair by a feature vector. We compute a feature from each cluster as follows. First, we assign a weight w_{ij} to a pattern p_i that is in a cluster c_j as follows,

$$w_{ij} = \frac{\mu(p_i)}{\sum_{q \in c_j} \mu(q)}. \quad (6)$$

Here, $\mu(q)$ is the total frequency of a pattern, and it is given by Equation 5. Because we perform a hard clustering on patterns, a pattern can belong to only one cluster (i.e. $w_{ij} = 0$ for $p_i \notin c_j$). Finally, we compute the value of the j-th feature in the feature vector for word pair (a, b) as follows,

$$\sum_{p_i \in c_j} w_{ij} f(a, b, p_i). \quad (7)$$

For each set of clusters, we compute the element Λ_{ij} of the corresponding inter-cluster correlation matrix Λ by the cosine similarity between the centroid vectors for clusters c_i and c_j. The prototype vector σ in Equation 3 is computed as the vector sum of individual feature vectors for the synonymous word pairs selected from the WordNet as described above. We then use Equation 3 to compute the average of similarity scores for synonymous word pairs we selected from WordNet.

We select the θ that maximizes the average similarity score between those synonymous word pairs. Formally, the optimal value of θ, $\hat{\theta}$ is given by the following Equation,

$$\hat{\theta} = \text{argmax}_{\theta \in [0,1]} \left(\frac{1}{|W|} \sum_{(a,b) \in W} \text{sim}(a, b) \right). \quad (8)$$

Here, W is the set of synonymous word pairs (a, b), $|W|$ is the total number of synonymous word pairs (i.e. 2000 in our experiments), and $\text{sim}(a, b)$ is given by Equation 3. Because the averages are taken over 2000 word pairs this procedure gives a reliable estimate for θ. Moreover, this method does not require negative training instances such as, non-synonymous word pairs, which are difficult to create manually. Average similarity scores for various θ values are shown in Figure 2. From Figure 2, we see that initially average similarity increases when θ is increased.

This is because clustering of semantically related patterns reduces the sparseness in feature vectors. Average similarity is stable within a range of θ values between 0.5 and 0.7. However, increasing θ beyond 0.7 results in a rapid drop of average similarity. To explain this behavior consider Figure 3 where we plot the sparsity of the set of clusters (i.e. the ratio between singletons to total clusters) against threshold θ. As seen from Figure 3, high θ values result in a high percentage of singletons because only highly similar patterns will form clusters. Consequently, feature vectors for different word pairs do not have many features in common. The maximum average similarity score of 1.303 is obtained with $\theta = 0.7$, corresponding to $17,015$ total clusters out of which $12,476$ are singletons with exactly one pattern (sparsity = 0.733). For the remainder of the experiments in this paper we set θ to this optimal value and use the corresponding set of clusters to compute semantic similarity by Equation 3. Similarity scores computed using Equation 3 can be greater than 1 (see Figure 2) because of the terms corresponding to the non-diagonal elements in Λ. We do not normalize the similarity scores to $[0, 1]$ range in our experiments because the evaluation metrics we use are insensitive to linear transformations of similarity scores.

6 Experiments

Table 1 compares the proposed method against Miller-Charles ratings (MC), and previously proposed web-based semantic similarity measures: Jaccard, Dice, Overlap, PMI (Bollegala et al., 2007), Normalized Google Distance (NGD) (Cilibrasi and Vitanyi, 2007), Sahami and Heilman (SH) (2006), co-occurrence double checking model (CODC) (Chen et al., 2006), and support vector machine-based (SVM) approach (Bollegala et al., 2007). The bottom row of Table 1 shows the Pearson correlation coefficient of similarity scores produced by each algorithm with MC. All similarity scores, except for the human-ratings in Miller-Charles dataset, are normalized to $[0, 1]$ range for the ease of comparison. It is noteworthy that the Pearson correlation coefficient is invariant under a linear transformation. All similarity scores shown in Table 1 except for the proposed method are taken from the original published papers.

The highest correlation is reported by the proposed semantic similarity measure. The improvement of the proposed method is statistically significant (confidence interval $[0.73, 0.93]$) against all the similarity measures compared in Table 1 except against the SVM approach. 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.

Table 2 summarizes the previously proposed WordNet-based semantic similarity measures. 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. 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.

We evaluate the proposed method using the WordSimilarity-353 dataset. Experimental results are presented in Table 3. Following previous work, we use Spearman rank correlation coefficient, which does not require ratings to be linearly dependent, for the evaluations on this dataset. Likewise with the Miller-Charles ratings, we measure the correlation between the similarity scores produced by the proposed method for word pairs in the WordSimilarity-353 dataset and the human ratings. A higher Spearman correlation coefficient (value=0.504, confidence interval $[0.422, 0.578]$) indicates a better agreement with the human notion of semantic similarity. From Table 3 we can see that the proposed method outperforms a wide variety of semantic similarity measures developed using numerous resources including lexical resources such as WordNet and knowledge sources such as Wikipedia (i.e. WikiRelate!). In contrast to the Miller-Charles dataset which only contains common English words selected from the WordNet, the WordSimilarity-353 dataset contains word pairs where one or both words are named entities (e.g. (*Maradona*, *foot-*

Table 1: Semantic similarity scores on Miller-Charles dataset

Word Pair	MC	Jaccrad	Dice	Overlap	PMI	NGD	SH	CODC	SVM	Proposed
automobile-car	3.920	0.650	0.664	0.831	0.427	0.466	0.225	0.008	0.980	0.918
journey-voyage	3.840	0.408	0.424	0.164	0.468	0.556	0.121	0.005	0.996	1.000
gem-jewel	3.840	0.287	0.300	0.075	0.688	0.566	0.052	0.012	0.686	0.817
boy-lad	3.760	0.177	0.186	0.593	0.632	0.456	0.109	0.000	0.974	0.958
coast-shore	3.700	0.783	0.794	0.510	0.561	0.603	0.089	0.006	0.945	0.975
asylum-madhouse	3.610	0.013	0.014	0.082	0.813	0.782	0.052	0.000	0.773	0.794
magician-wizard	3.500	0.287	0.301	0.370	0.863	0.572	0.057	0.008	1.000	0.997
midday-noon	3.420	0.096	0.101	0.116	0.586	0.687	0.069	0.010	0.819	0.987
furnace-stove	3.110	0.395	0.410	0.099	1.000	0.638	0.074	0.011	0.889	0.878
food-fruit	3.080	0.751	0.763	1.000	0.449	0.616	0.045	0.004	0.998	0.940
bird-cock	3.050	0.143	0.151	0.144	0.428	0.562	0.018	0.006	0.593	0.867
bird-crane	2.970	0.227	0.238	0.209	0.516	0.563	0.055	0.000	0.879	0.846
implement-tool	2.950	1.000	1.000	0.507	0.297	0.750	0.098	0.005	0.684	0.496
brother-monk	2.820	0.253	0.265	0.326	0.623	0.495	0.064	0.007	0.377	0.265
crane-implement	1.680	0.061	0.065	0.100	0.194	0.559	0.039	0.000	0.133	0.056
brother-lad	1.660	0.179	0.189	0.356	0.645	0.505	0.058	0.005	0.344	0.132
car-journey	1.160	0.438	0.454	0.365	0.205	0.410	0.047	0.004	0.286	0.165
monk-oracle	1.100	0.004	0.005	0.002	0.000	0.579	0.015	0.000	0.328	0.798
food-rooster	0.890	0.001	0.001	0.412	0.207	0.568	0.022	0.000	0.060	0.018
coast-hill	0.870	0.963	0.965	0.263	0.350	0.669	0.070	0.000	0.874	0.356
forest-graveyard	0.840	0.057	0.061	0.230	0.495	0.612	0.006	0.000	0.547	0.442
monk-slave	0.550	0.172	0.181	0.047	0.611	0.698	0.026	0.000	0.375	0.243
coast-forest	0.420	0.861	0.869	0.295	0.417	0.545	0.060	0.000	0.405	0.150
lad-wizard	0.420	0.062	0.065	0.050	0.426	0.657	0.038	0.000	0.220	0.231
cord-smile	0.130	0.092	0.097	0.015	0.208	0.460	0.025	0.000	0	0.006
glass-magician	0.110	0.107	0.113	0.396	0.598	0.488	0.037	0.000	0.180	0.050
rooster-voyage	0.080	0.000	0.000	0.000	0.228	0.487	0.049	0.000	0.017	0.052
noon-string	0.080	0.116	0.123	0.040	0.102	0.488	0.024	0.000	0.018	0.000
Correlation	**-**	**0.260**	**0.267**	**0.382**	**0.549**	**0.205**	**0.580**	**0.694**	**0.834**	**0.867**

Table 2: Comparison with WordNet-based similarity measures.

Method	Correlation
Edge-counting	0.664
Jiang & Conrath (1998)	0.848
Lin (1998a)	0.822
Resnik (1995)	0.745
Li et al. (2003)	0.891

Table 3: Results on WordSimilarity-353 dataset.

Method	Correlation
WordNet Edges (Jarmasz, 1993)	0.27
Hirst & St-Onge (1997)	0.34
Jiang & Conrath (1998)	0.34
WikiRelate! (Strube and Ponzetto, 2006)	0.19-0.48
Leacock & Chodrow (1998)	0.36
Lin (1998b)	0.36
Resnik (1995)	0.37
Proposed	0.504

ball) and (*Jerusalem*, *Israel*)). Because the proposed method use snippets retrieved from a web search engine, it is capable of extracting expressive lexical patterns that can explicitly state the relationship between two entities.

If we must compare n objects using a feature model of similarity, then we only need to define features for each of those n objects. However, in the proposed relational model we must define relations between all pairs of objects. In the case where all n objects are different, this requires us to define relations for $n(n-1)/2$ object pairs. Defining relations for all pairs can be computationally costly for large n values. Efficiently comparing n objects using a relational model is an interesting future research direction of the current work.

7 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 two benchmark datasets.

References

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

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

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

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

J. Curran. 2002. Ensemble menthods for automatic thesaurus extraction. In *Proc. of EMNLP*.

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

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

E. Gabrilovich and S. Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of IJCAI'07*, pages 1606–1611.

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

U. Hahn, N. Chater, and L. B. Richardson. 2003. Similarity as transformation. *Cognition*, 87:1–32.

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

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

G. Hirst and D. St-Onge. 1997. Lexical chains as representations of context for the detection and correction of malapropisms.

M. Jarmasz. 1993. Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa.

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

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

C. Leacock and M. Chodorow. 1998. *Combining Local Context and WordNet Similarity for Word Sense Identification*. MIT.

M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitanyi. 2004. The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.

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

D. Lin. 1998b. An information-theoretic definition of similarity. In *Proc. of the 15th ICML*, pages 296–304.

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

J. Pei, J. Han, B. Mortazavi-Asi, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. 2004. Mining sequential patterns by pattern-growth: the prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440.

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

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

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

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

M. Strube and S. P. Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proc. of AAAI' 06*.

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

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

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