# An integrated approach to measuring Semantic Similarity between Words using Information available on the Web

**Blind Review**
No Name
No Affiliation
No Address
noname@email

**Blind Review**
No Name
No Affiliation
No Address
noname@email

## Abstract

Measuring semantic similarity between words is vital for various applications in natural language processing, such as language modeling, information retrieval, and document clustering. We propose a method that utilizes the information available on the Web to measure semantic similarity between a pair of words or entities. We integrate page counts for each word in the pair and lexico-syntactic patterns that occur among the top ranking snippets for the *AND* query using support vector machines. Experimental results on Miller-Charles' benchmark data set show that the proposed measure outperforms all the existing web based semantic similarity measures by a wide margin, achieving a correlation coefficient of $0.8129$. Moreover, the proposed semantic similarity measure significantly improves the accuracy (F-measure of $0.78$) in a named entity clustering task, proving the capability of the proposed measure to capture semantic similarity using web content.

## 1 Introduction

The study of semantic similarity between words has been an integral part of natural language processing and information retrieval for many years. Semantic similarity measures are vital for various applications in natural language processing such as word sense disambiguation (Resnik, 1999), language modeling (Rosenfield, 1996), synonym extraction (Lin, 1998a) and automatic thesaurus extraction (Curran, 2002).

Pre-compiled taxonomies such as WordNet [1] and text corpora have been used in previous work on semantic similarity (Lin, 1998a; Resnik, 1995; Jiang and Conrath, 1998; Lin, 1998b). However, semantic similarity between words change over time as new senses and associations of words are constantly created. One major issue behind taxonomies and corpora oriented approaches is that they might not necessarily capture similarity between proper names such as named entities (e.g., personal names, location names, product names) and the new uses of existing words. For example, *apple* is frequently associated with *computers* on the Web but this sense of apple is not listed in the WordNet. Maintaining an up-to-date taxonomy of all the new words and new usages of existing words is costly if not impossible.

The Web can be regarded as a large-scale, dynamic corpus of text. Regarding the Web as a live corpus has become an active research topic recently. Simple, unsupervised models have shown to perform better when n-gram counts are obtained from the web rather than from a large corpus (Keller and Lapata, 2003; Lapata and Keller, 2005). Resnik and Smith (2003) extracts bilingual sentences from the Web to create parallel corpora for machine translation. Turney (2001) defines a point wise mutual information (PMI-IR) measure using the number of hits returned by a web search engine to recognize synonyms. Matsuo et. al, (2006b) follows a similar

[1] http://wordnet.princeton.edu/

approach to measure the similarity between words and apply their method in a graph-based word clustering algorithm.

Due to the huge number of documents and the high growth rate of the Web, it is difficult to directly analyze each individual document separately. Search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most Web search engines. Page count of a query is the number of pages that contain the query words [2]. A snippet is a brief window of text extracted by a search engine around the query term in a document. Snippets provide useful information about the immediate context of the query term.

This paper proposes a web based semantic similarity metric which combines page counts and snippets using support vector machines. We extract lexico-syntactic patterns from snippets. For example, ***X is a Y*** indicates there is a high semantic similarity between ***X*** and ***Y***. Automatically extracted lexico-syntactic patterns have been successfully employed in various term extraction tasks (Hearst, 1992).

Our contributions are summarized as follows:

- We propose a lexico-syntactic patterns based approach to compute semantic similarity using snippets obtained from a web search engine.

- We integrate different web based similarity measures using WordNet synsets and support vector machines to create a robust semantic similarity measure. The integrated measure outperforms all existing web based semantic similarity measures in a benchmark data set and a named entity clustering task. To the best of our knowledge, this is the first attempt to combine both WordNet synsets and web content to leverage a robust semantic similarity measure.

## 2 Previous Work

Given a taxonomy of concepts, a straightforward method for calculating similarity between two words (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., having more than one sense) then multiple paths may exist between the two words. In such cases only the shortest path between any two senses of the words is considered for the calculation of similarity. A problem frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent uniform distances.

Resnik (1995) proposes a similarity measure based on information content. He defines 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 are defined as the maximum of the similarity between any concepts that the words belong to. He uses WordNet as the taxonomy and information content is calculated using Brown corpus.

Li et al., (2003) combines structural semantic information from a lexical taxonomy and information content from a corpus in a non-linear model. They propose a similarity measure that uses shortest path length, depth and local density in a taxonomy. Their experiments using WordNet and Brown corpus reports a pearson correlation coefficient of $0.8914$ on the Miller and Charles' (1998) benchmark dataset. They do not evaluate their method on similarities between named entities. Recently, some work has been carried out on measuring semantic similarity using web content. Matsuo et al., (2006a) propose the use of web hits for the extraction of communities on the Web. They measure the association between two personal names using the simpson coefficient, calculated based on the number of web hits for each individual name and their conjunction.

Sahami et al., (2006) measures semantic similarity between two queries using the 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 do not compare their similarity measure with taxonomy based similarity measures.

Chen et al., (2006) propose a snippet based

[2] page count may not necessarily be equal to the word frequency because the queried word may appear many times in a page

double-checking model to compute semantic similarity between words. For two words $P$ and $Q$, they collect snippets for each word from a web search engine. Then they count the number of occurrences of word $P$ in the snippets for word $Q$ and the number of occurrences of word $Q$ in the snippets for word $P$. These values are combined non-linearly to compute the similarity between $P$ and $Q$. This method heavily depends on the search engine's ranking algorithm. Although two words $P$ and $Q$ may be very similar, there is no reason to believe that one can find $Q$ in the snippets for $P$, 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 and Charles (1998) data set.

## 3 Method

In this section we will describe the various similarity features we use in our model. We utilize page counts and snippets returned by the Google [3] search engine for simple text queries to define various similarity scores.

### 3.1 Page Counts based Similarity Scores

For the rest of this paper we use the notation $H(P)$ to denote the page count for the query $P$ in a search engine. Terra and Clarke (2003) evaluate various statistical metrics for measuring similarity between words in a corpus. We modify the traditional Jaccard, Simpson, Dice and PMI measures for the purpose of measuring similarity using page counts. WebJaccard coefficient between words (or phrases) $P$ and $Q$, $\mathrm{WebJaccard}(P, Q)$, is defined by,

$$\mathrm{WebJaccard}(P, Q) = \begin{cases} 0 & \text{if } H(P \cap Q) \leq c \\ \frac{H(P\cap Q)}{H(P)+H(Q)-H(P\cap Q)} & \text{otherwise} \end{cases} \quad (1)$$

Here, $P \cap Q$ denotes the conjunction query *P AND Q*. Given the scale and noise in web data, it is a possible that two words may appear on some pages accidentally. In order to reduce the adverse effect due to random co-occurrences, we set the WebJaccard coefficient to zero if the page counts for the query $P \cap Q$ is less than a threshold $c$. [4]

Likewise, we define WebSimpson coefficient, $\mathrm{WebSimpson}(P, Q)$, as,

$$\mathrm{WebSimpson}(P, Q) = \begin{cases} 0 & \text{if } H(P \cap Q) \leq c \\ \frac{H(P\cap Q)}{\min(H(P),H(Q))} & \text{otherwise} \end{cases}. \quad (2)$$

We define *WebDice* as a variant of Dice coefficient. $\mathrm{WebDice}(P, Q)$ is defined as,

$$\mathrm{WebDice}(P, Q) = \begin{cases} 0 & \text{if } H(P \cap Q) \leq c \\ \frac{2H(P\cap Q)}{H(P)+H(Q)} & \text{otherwise} \end{cases}. \quad (3)$$

We define *WebPMI* as a variant form of PMI using page counts by,

$$\mathrm{WebPMI}(P, Q) = \begin{cases} 0 & \text{if } H(P \cap Q) \leq c \\ \log_2\left(\frac{\frac{H(P\cap Q)}{N}}{\frac{H(P)}{N}\frac{H(Q)}{N}}\right) & \text{otherwise} \end{cases} \quad (4)$$

Here, $N$ is the number of documents indexed by the search engine. Probabilities in Formula 4 are estimated according to the maximum likelihood principle. In order to accurately calculate PMI using Formula 4, we need to know $N$, the number of documents indexed by the search engine. Although estimating the number of documents indexed by a search engine (Bar-Yossef and Gurevich, 2006) is an interesting task itself, it is beyond the scope of this work. In this work, we set $N = 10^{10}$ according to the number of indexed pages reported by Google.

### 3.2 Snippets based Synonymous Word Patterns

Page counts based similarity measures do not consider the relative distance between $P$ and $Q$ in a page or the length of the page. Although $P$ and $Q$ may occur in a page they might not be related at all. Therefore, page counts based similarity measures are prone to noise and are not reliable when $H(P \cap Q)$ is low. On the other hand snippets capture the local context of query words. We propose lexico-syntactic patterns extracted from snippets as a solution to the problems with page counts based similarity measures.

To illustrate our pattern extraction algorithm consider the following snippet from Google for the query ***jaguar*** *AND* ***cat***.

[3] http://www.google.com

[4] we set $c = 5$ in our experiments

*”The **Jaguar** is the largest **cat** in Western Hemisphere and can subdue large prey than can the puma”*

Here, the phrase *is the largest* indicates a hypernymic (instance-class) relationship between Jaguar and the cat. Phrases such as *also known as, is a, part of, is an example of* all indicate different degrees of semantic similarities. Such indicative phrases have been successfully applied in various tasks such as synonym extraction, hyponym extraction (Hearst, 1992) and fact extraction (Pasca et al., 2006).

We will describe our pattern extraction algorithm in three steps.

#### Step 1

We replace the query terms in the snippet by a marker (let us assume ’X’ to be the query term marker) and extract all word $n$-grams that contain both query terms (i.e., all n-grams with exactly two ’X’s). In our experiments we extracted n-grams for $n =$ 2,3,4 and 5. We ignore the snippets where only one of the query terms appear. For example, from the previous snippet we extract the pattern, ***X** is the largest **X***. In order to leverage the pattern extraction process, we select $5000$ pairs of synonymous nouns from WordNet synsets. We ignore the nouns which do not have synonyms in the WordNet. For nouns with more than one sense, we select synonyms from its dominant sense. For each pair of synonyms $P$ and $Q$, we query google for ”$P$” AND ”$Q$” and download the snippets. Let us call this collection of snippets as the *positive corpus*. We apply the above mentioned n-gram based pattern extraction procedure and count the frequency of each valid pattern in the positive corpus.

#### Step 2

Pattern extraction process described in step 1 yields $4,562,471$ unique patterns. $80\%$of these patterns occur less than $10$ times in the positive corpus. It is impossible to learn with such a large number of sparse patterns. Moreover, some patterns may occur purely randomly in a snippet and are not good indicators of semantic similarity. To measure the reliability of a pattern as an indicator of semantic similarity we adopt the following feature selection procedure. We create a set of non-synonymous word pairs by randomly shuffling the words in our data set of synonymous word pairs. We check each pair of words in this newly created data set against WordNet and confirmed that they do not belong to any synsets of the WordNet. From this procedure we created $5000$ non-synonymous pairs of words. For each non-synonymous word pair, we query Google for the conjunction of its words and download snippets. Let us call this collection of snippets as the *negative corpus*. For each pattern generated in step 1, we count its frequency in the negative corpus.

Table 1: Contingency table

| | $v$ | other than $v$ | All |
|---|---|---|---|
| Freq. in positive corpus | $p_v$ | $P - p_v$ | $P$ |
| Freq. in negative corpus | $n_v$ | $N - n_v$ | $N$ |

#### Step 3

We create a contingency table as shown in Table 1 for each pattern $v$ extracted in step 1 using its frequency $p_v$ in positive corpus and $n_v$ in negative corpus. In Table 1, $P$ denotes the total frequency of all patterns in the positive corpus and $N$ denotes that in the negative corpus.

Using the information in Table 1, we calculate $\chi^2$ (Manning and Schütze, 2002) value for each pattern as,

$$\chi^2 = \frac{(P+N)(p_v(N-n_v) - n_v(P-p_v))^2}{PN(p_v+n_v)(P+N-p_v-n_v)}. \tag{5}$$

We selected the top ranking $200$ patterns experimentally as described in section 4.2 according to their $\chi^2$ values. Some of the selected patterns are shown in Table 2.

### 3.3 Training

For each pair of synonymous and non-synonymous words in our data sets, we count the frequency of occurrence of the patterns selected in Step 3. We normalize the frequency count of each pattern by dividing from the total frequency of all patterns in the snippets. We also compute the page counts based features as given by formulae (1-4). Using the $200$ pattern features and the $4$ page counts based features we create $204$ dimensional feature vectors for each training instance in our synonymous and non-synonymous datasets. We train a two class support vector machine (SVM) (Vapnik, 1998), where class

$+1$ represents synonymous word pairs and class $-1$ represents non-synonymous word pairs. We train the parameters of a sigmoid function to map the SVM outputs into probabilities (Platt, 2000). We consider the probability of a given pair of words belonging to class $+1$ as the semantic similarity between the two words.

## 4 Experiments

To evaluate the performance of the proposed semantic similarity measure, we conduct two sets of experiments. First we compare the similarity scores produced by the proposed measure against the Miller-Charles' benchmark data set. We analyze the performance of the proposed measure with the number of snippets and the size of the training data set. Secondly, we apply the proposed measure in a real-world named entity clustering task and measure its performance.

### 4.1 The Benchmark Data Set

We evaluated the proposed method against Miller-Charles (1998) data set, a dataset of $30$ [5] word pairs rated by a group of $38$ human subjects. Word pairs are rated on a scale from $0$ (no similarity) to $4$ (perfect synonymy). Miller-Charles data set is a subset of Rubenstein-Goodenough's (1965) original data set of $65$ word pairs. Although Miller-Charles experiment was carried out $25$ years later than Rubenstein-Goodenough's, two sets of ratings are highly correlated (pearson correlation coefficient=$0.97$). Therefore, Miller-Charles ratings can be considered as a reliable benchmark for evaluating semantic similarity measures.

### 4.2 Pattern Selection

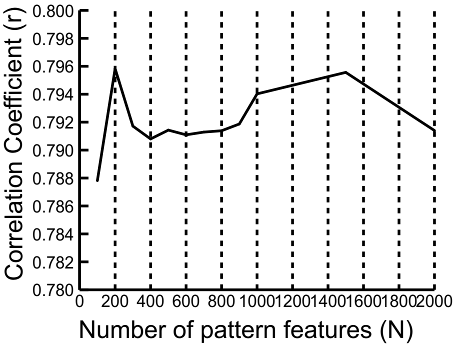

Figure 1: Correlation vs No of pattern features

Table 2: Features with the highest SVM linear kernel weights

| feature | $\chi^2$ | SVM weight |
|---|---|---|
| WebDice | N/A | 8.19 |
| X/X | 33459 | 7.53 |
| X, X : | 4089 | 6.00 |
| X or X | 3574 | 5.83 |
| X X for | 1089 | 4.49 |
| X . the X | 1784 | 2.99 |
| with X ( X | 1819 | 2.85 |
| X=X | 2215 | 2.74 |
| X and X are | 1343 | 2.67 |
| X of X | 2472 | 2.56 |

We trained a linear kernel SVM with top $N$ pattern features (ranked according to their $\chi^2$ value) and calculated the correlation coefficient against the Miller-Charles benchmark data set. Experimental results are shown in Figure 1. From Figure 1 we select $N = 200$, where correlation maximizes. Features with the highest linear kernel weights are shown in Table 2 alongside with their $\chi^2$ values. Weight of a feature in the linear kernel can be considered as a rough estimate of the influence it has on the final SVM output. WebDice has the highest kernel weight followed by a series of pattern based features. WebSimpson (rank=$18$, weight=$2.45$), WebJaccard (rank=$66$, weight=$0.618$) and WebPMI (rank=$138$, weight=$0.0001$) are not shown in Table 2 due to space limitations. It is interesting to note that the pattern features in Table 2 agree with the intuition. Lexical patterns (e.g., *X or X, X and X are, X of X*) as well as script patterns (e.g., bracketing, comma usage) are extracted by our method.

### 4.3 Semantic Similarity

We score the word pairs in Miller-Charles data set using the page count based similarity measures, previous work on web based semantic similarity measures (Sahami (2006), Chen (2006)) and the proposed method (SVM). Results are shown in Table 4.3. All figures except for the Miller-Charles ratings are normalized into $[0, 1]$ range for the ease

[5] Due to the omission of two word pairs in earlier versions of WordNet most researchers had used only $28$ pairs for evaluations

Table 3: Semantic Similarity of Human Ratings and baselines on Miller-Charles dataset

| Word Pair | Miller-Charles | Web Jaccard | Web Dice | Web Simpson | Web PMI | Sahami (2006) | Chen (CODC) (2006) | SVM (linear) |
|---|---|---|---|---|---|---|---|---|
| cord-smile | 0.13 | 0.102 | 0.108 | 0.036 | 0.207 | 0.090 | 0 | 0.005 |
| rooster-voyage | 0.08 | 0.011 | 0.012 | 0.021 | 0.228 | 0.197 | 0 | 0.030 |
| noon-string | 0.08 | 0.126 | 0.133 | 0.060 | 0.101 | 0.082 | 0 | 0.026 |
| glass-magician | 0.11 | 0.117 | 0.124 | 0.408 | 0.598 | 0.143 | 0 | 0.207 |
| monk-slave | 0.55 | 0.181 | 0.191 | 0.067 | 0.610 | 0.095 | 0 | 0.441 |
| coast-forest | 0.42 | 0.862 | 0.870 | 0.310 | 0.417 | 0.248 | 0 | 0.594 |
| monk-oracle | 1.1 | 0.016 | 0.017 | 0.023 | 0 | 0.045 | 0 | 0.417 |
| lad-wizard | 0.42 | 0.072 | 0.077 | 0.070 | 0.426 | 0.149 | 0 | 0.277 |
| forest-graveyard | 0.84 | 0.068 | 0.072 | 0.246 | 0.494 | 0 | 0 | 0.621 |
| food-rooster | 0.89 | 0.012 | 0.013 | 0.425 | 0.207 | 0.075 | 0 | 0.082 |
| coast-hill | 0.87 | 0.963 | 0.965 | 0.279 | 0.350 | 0.293 | 0 | 0.942 |
| car-journey | 1.16 | 0.444 | 0.460 | 0.378 | 0.204 | 0.189 | 0.290 | 0.418 |
| crane-implement | 1.68 | 0.071 | 0.076 | 0.119 | 0.193 | 0.152 | 0 | 0.156 |
| brother-lad | 1.66 | 0.189 | 0.199 | 0.369 | 0.644 | 0.236 | 0.379 | 0.352 |
| bird-crane | 2.97 | 0.235 | 0.247 | 0.226 | 0.515 | 0.223 | 0 | 0.922 |
| bird-cock | 3.05 | 0.153 | 0.162 | 0.162 | 0.428 | 0.058 | 0.502 | 0.674 |
| food-fruit | 3.08 | 0.753 | 0.765 | 1 | 0.448 | 0.181 | 0.338 | 0.999 |
| brother-monk | 2.82 | 0.261 | 0.274 | 0.340 | 0.622 | 0.267 | 0.547 | 0.447 |
| asylum-madhouse | 3.61 | 0.024 | 0.025 | 0.102 | 0.813 | 0.212 | 0 | 0.792 |
| furnace-stove | 3.11 | 0.401 | 0.417 | 0.118 | 1 | 0.310 | 0.928 | 0.918 |
| magician-wizard | 3.5 | 0.295 | 0.309 | 0.383 | 0.863 | 0.233 | 0.671 | 1 |
| journey-voyage | 3.84 | 0.415 | 0.431 | 0.182 | 0.467 | 0.524 | 0.417 | 0.998 |
| coast-shore | 3.7 | 0.786 | 0.796 | 0.521 | 0.561 | 0.381 | 0.518 | 0.977 |
| implement-tool | 2.95 | 1 | 1 | 0.517 | 0.296 | 0.419 | 0.419 | 0.810 |
| boy-lad | 3.76 | 0.186 | 0.196 | 0.601 | 0.631 | 0.471 | 0 | 0.985 |
| automobile-car | 3.92 | 0.654 | 0.668 | 0.834 | 0.427 | 1 | 0.686 | 0.993 |
| midday-noon | 3.42 | 0.106 | 0.112 | 0.135 | 0.586 | 0.289 | 0.856 | 0.871 |
| gem-jewel | 3.84 | 0.295 | 0.309 | 0.094 | 0.687 | 0.211 | 1 | 0.756 |
| **Correlation** | - | 0.2598 | 0.26711 | 0.3822 | 0.5489 | 0.5797 | 0.6936 | 0.8129 |

of comparison [6]. Proposed method (SVM) reports the highest correlation of $0.8129$ in our experiments. Our reimplementation of Co-occurrence Double Checking (CODC) measure (Chen et al., 2006) reports the second best correlation of $0.6936$. However, CODC measure reports zero similarity for many word pairs. This is because for a word-pair $(P, Q)$, we might not necessarily find $Q$ among the top snippets for $P$ (and vice versa). CODC measure returns zero under these conditions. Sahami et al. (2006) is ranked third with a correlation of $0.5797$. Among the four page counts based measures WebPMI reports the highest correlation ($r = 0.5489$). In general, the results in Table 4.3 suggest that snippet based measures are accurate than page counts based measures in capturing semantic similarity. This is evident for word-pairs where at least one of the words is a polysemous word (e.g., pairs that include *cock*, *brother*). Page counts based measures do not consider the context in which the words appear in a page, thus cannot disambiguate the multiple senses.

Table 4: Comparison with taxonomy based methods

| Method | correlation |
|---|---|
| Human replication | 0.9015 |
| Resnik (1995) | 0.7450 |
| Lin (1998) | 0.8224 |
| Li et al (2003) | 0.8914 |
| Edge-counting | 0.664 |
| Information content | 0.745 |
| Jiang & Conrath (1998) | 0.8484 |
| proposed (SVM) | 0.8129 |

As summarized in Table 4.3, proposed method is comparable with the WordNet based methods. In fact, the proposed method outperforms simple WordNet based approaches such as Edge-counting and Information Content measures. However, considering the high correlation between human sub-

[6] Pearson correlation coefficient is invariant against a linear transformation

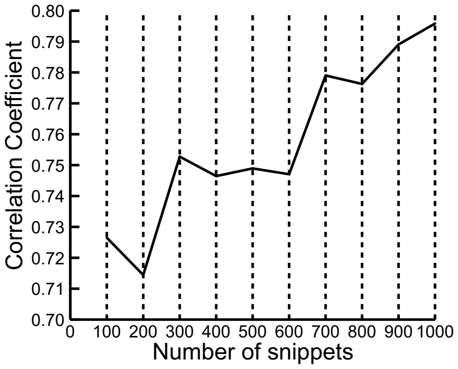

Figure 2: Correlation vs No of snippets

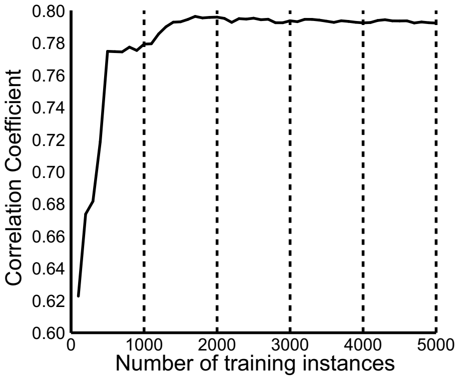

Figure 3: Correlation vs No of positive and negative training instances

jects ($0.9$), there is still room for improvement.

Figure 2 illustrates the effect of the number of snippets on the performance of the proposed method. Correlation coefficient steadily improves with the number of snippets used for extracting patterns. When few snippets are processed only a few patterns are found, thus the feature vector becomes sparse, resulting in poor performance.

We train an SVM with $N$, equal number of positive and negative training instances and compute the correlation with Miller-Charles ratings. Results are shown in Figure 3. The maximum correlation is reached with $2000$ training instances. Therefore, it is sufficient to consider just $2000$ pairs of synonyms from WordNet to leverage the proposed semantic similarity measure.

### 4.4 Named Entity Clustering

Measuring semantic similarity between named entities is vital in many applications such as query expansion (Sahami and Heilman, 2006) and community mining (Matsuo et al., 2006a). Since most named entities are not covered by WordNet, similarity measures based on WordNet cannot be directly used in these tasks. Unlike common English words, named entities are constantly being created. Manually maintaining an up-to-date taxonomy of named entities is costly, if not impossible. The proposed semantic similarity measure is appealing as it does not require pre-compiled taxonomies. In order to evaluate the performance of the proposed measure in capturing the semantic similarity between named entities, we set up a named entity clustering task. We selected $50$ person names from $5$ categories : tennis players, golfers, actors, politicians and scientists, ($10$ names from each category) from the *domz* directory [7]. For each pair of names in our data set, we measure their similarity using the proposed method and baselines. We use group-average agglomerative hierarchical clustering to cluster the names in our dataset into five clusters. We employed the B-CUBED metric (Bagga and Baldwin, 1998) to evaluate the clustering results. As summarized in Table 5 the proposed method outperforms all the baselines with a statistically significant ($p \leq 0.01$ Tukey HSD) F score of $0.7897$.

Table 5: Performance of named entity clustering

| Method | Precision | Recall | F Measure |
|---|---|---|---|
| WebJaccard | $0.5926$ | $0.712$ | $0.6147$ |
| WebSimpson | $0.5976$ | $0.68$ | $0.5965$ |
| WebDice | $0.5895$ | $0.716$ | $0.6179$ |
| WebPMI | $0.2649$ | $0.428$ | $0.2916$ |
| Sahami (2006) | $0.6384$ | $0.668$ | $0.6426$ |
| Chen (2006) | $0.4763$ | $0.624$ | $0.4984$ |
| Proposed | $0.7958$ | $0.804$ | $0.7897$ |

## 5 Conclusion

We introduce an SVM based approach to combine page counts and lexico-syntactic patterns extracted from snippets to leverage a robust web based semantic similarity measure. The proposed similarity measure outperforms existing web based similarity measures and competes with models trained on WordNet. It requires just $2000$ synonymous word pairs automatically extracted from WordNet synsets for training. Moreover, the proposed method proves

[7] http://domz.org

its usefulness in a named entity clustering task. In future we intend to apply the lexico-syntactic patterns extracted by our method to find synonyms from the web.

## References

A. Bagga and B. Baldwin. 1998. Entity-based cross document coreferencing using the vector space model. In *Proc. of 36th COLING-ACL*, pages 79–85.

Z. Bar-Yossef and M. Gurevich. 2006. Random sampling from a search engine's index. In *Proceedings of 15th International World Wide Web Conference*.

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

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

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

J.J. Jiang and D.W. Conrath. 1998. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the International Conference on Research in Computational Linguistics ROCLING X*.

F. Keller and M. Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.

M. Lapata and F. Keller. 2005. Web-based models ofr natural language processing. *ACM Transactions on Speech and Language Processing*, 2(1):1–31.

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.

C. D. Manning and H. Schütze. 2002. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka. 2006a. Polyphonet: An advanced social network extraction system. In *Proc. of 15th International World Wide Web Conference*.

Y. Matsuo, T. Sakaki, K. Uchiyama, and M. Ishizuka. 2006b. Graph-based word clustering using web search engine. In *Proc. of EMNLP 2006*.

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

M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. 2006. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *Proc. of AAAI-2006*.

J. Platt. 2000. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74.

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 and N. A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of 14th International Joint Conference on Aritificial Intelligence*.

P. Resnik. 1999. Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language. *Journal of Aritificial Intelligence Research*, 11:95–130.

R. Rosenfield. 1996. A maximum entropy approach to adaptive statistical modelling. *Computer Speech and Language*, 10:187–228.

H. Rubenstein and J.B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8:627–633.

M. Sahami and T. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of 15th International World Wide Web Conference*.

E. Terra and C.L.A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proc. of the NAACL/HLT*, pages 165–172.

P. D. Turney. 2001. Minning the web for synonyms: Pmi-ir versus lsa on toefl. In *Proc. of ECML-2001*, pages 491–502.

V. Vapnik. 1998. *Statistical Learning Theory*. Wiley, Chichester, GB.

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.