1G2-5

WebSim: A Web-based Semantic Similarity Measure

Danushka Bollegala[*1] Yutaka Matsuo[*2] Mitsuru Ishizuka[*1]

[*1]The University of Tokyo
[*2]Japanese National Institute of Advanced Industrial Science and Technology

Semantic similarity measures are important for numerous tasks in natural language processing such as word sense disambiguation, automatic synonym extraction, language modelling and document clustering. We propose a method to measure semantic similarity between two words using information available on the Web. We extract page counts and snippets for the *AND* query of the two words from a Web search engine. We define numerous similarity scores based on page counts and lexico-syntactic patterns. These similarity scores are integrated using support vector machines to form a robust semantic similarity measure. Proposed method outperforms all existing Web-based semantic similarity measures on Miller-Charles benchmark dataset achieving a high correlation coefficient of 0.834 with human ratings.

1. Introduction

The study of semantic similarity between words has long been an integral part of natural language processing. Semantic similarity measures are successfully employed in various natural language tasks such as word sense disambiguation , language modelling, synonym extraction and automatic thesauri extraction.

Manually compiled taxonomies such as WordNet [*1] and text corpora have been used in previous work on semantic similarity [5, 12, 3, 6]. However, semantic similarity between two words is a dynamic phenomenon that varies over time and across domains. For example, *apple* is frequently associated with *computers* on the Web. A user who searches for *apple* on the Internet might be interested in this sense of apple but not in apple as a fruit. General purpose taxonomies do not completely cover all types of named-entities (i.e., personal names, product names, organization names, etc). Maintaining an up to date taxonomy of all the new words and the new senses assigned to existing words is costly if not impossible. Therefore, semantic similarity measures based on taxonomies alone are insufficient.

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 [4]. Web search engines provide an efficient interface to the vast information available on the Web. *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. A snippet is a brief window of text extracted by a search engine around the query term in a document. Snippets provide useful information regarding the local context of the query term. For example, consider the snippet shown in Figure 1 retrieve by *Google* for the query ***Jaguar** AND **cat***.

Here, the phrase *is the largest* indicates a hypernymic relationship between Jaguar and cat. Phrases such as *also known as, is a, part of, is an example of* all indicate various semantic relations. Such indicative phrases have been applied to numerous tasks with good results, such as hyponym extraction [2]. From the previous example, we form the pattern ***X** is the largest **Y***, by replacing the two words *Jaguar* and *cat* by two wildcards **X** and **Y**. In this paper we propose an automatically extracted lexico-syntactic patterns based approach to compute semantic similarity using text snippets obtained from a web search engine.

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

Figure 1: A snippet for the query ***Jaguar** AND **cat***

2. Previous Work

Given a taxonomy of concepts, a straightforward method to compute similarity between two words (concepts) is to find the length of the shortest path connecting the two words in the taxonomy [11]. 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. A problem frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent uniform distances.

Resnik [12] 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.

Recently, some work has been carried out on measuring semantic similarity using Web content. Matsuo et al., [8] propose the use of web hits for the extraction of communities on the Web. They measure the association between two personal names using the overlap coefficient, calculated based on the number of web hits for each individual name and their conjunction.

Sahami et al., [13] measure 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 vec-

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

*1 http://wordnet.princeton.edu/

tor. 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., [1] propose a 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. They count the number of occurrences of word P among the top n snippets for word Q and the number of occurrences of word Q among the top n snippets for word P. These values are combined non-linearly to compute the similarity between P and Q. Although two words P and Q are semantically similar, there is no guarantee that one can find Q among the top n snippets for P, or vice versa, because search engines consider many factors such as freshness, link authority (Page Rank) when ranking the search results. This observation is confirmed by the experimental results in their paper which reports 0 similarity scores for many word-pairs in the benchmark dataset.

3. Method

3.1 Page-count-based Similarity Scores

Page counts for the query *P AND Q*, can be considered as an approximation of co-occurrence of two words P and Q on the Web. We modify four popular co-occurrence measures; Jaccard, Overlap (Simpson), Dice, and PMI (point-wise mutual information), to compute semantic similarity using page counts. 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.

WebJaccard coefficient between words P and 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 possible that two words may appear on some pages purely accidentally. In order to reduce the adverse effects attributable to random co-occurrences, we set the WebJaccard coefficient to zero if the page count for the query $P \cap Q$ is less than a threshold c. *2

Similarly, we define WebOverlap coefficient as,

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

We define WebDice as a variant of Dice coefficient by,

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

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

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

*2 we set $c = 5$ in our experiments

Algorithm 3..1: EXTRACTPATTERNS(S)

comment: Given set S of word-pairs, extract patterns

for each $\mathrm{pair}(A,B) \in S$
  do $D \leftarrow \mathrm{GetSnippets}(\text{“}A\ B\text{”})$
$N \leftarrow null$
for each $\mathrm{snippet}d \in D$
  do $N \leftarrow N + \mathrm{GetNgrams}(d, A, B)$
$Pats \leftarrow \mathrm{CountFreq}(N)$
return $(Pats)$

Figure 2: Extract patterns from snippets.

Probabilities in Eq. 4 are estimated according to the maximum likelihood principle. To calculate PMI accurately using Eq. 4, we must know N, the number of documents indexed by the search engine. In the present work, we set $N = 10^{10}$ according to the number of indexed pages reported by Google.

3.2 Extracting Lexico-Syntactic Patterns

Page counts based similarity measures do not consider the relative distance between words that co-occur in a page. Although two words co-occur in a page they might not be related. Therefore, similarity scores defined purely on page counts are prone to noise and are not reliable when the page counts are low. On the other hand, snippets capture the local context of query words. We propose lexico-syntactic patterns, automatically extracted from snippets, to overcome these drawbacks.

Our pattern extraction algorithm is illustrated in Figure 2. Given a set S of synonymous word-pairs, *GetSnippets* function returns a list of text snippets for the query *”A” AND ”B”* for each word-pair A, B in S. For each snippet found, we replace the two words in the query by two wildcards *X* and *Y*. For each snippet d in the set of snippets D returned by *GetSnippets*, function *GetNgrams* extracts word n-grams for $n = 2, 3, 4$ and 5. We select n-grams which contain exactly one *X* and one *Y*. For example, the snippet in Figure 1 yields the pattern *X is the largest Y*. Finally, function *CountFreq* counts the frequency of each pattern we extracted. This procedure considers the words that fall between X and Y as well as words that precede X and succeeds Y.

To leverage the pattern extraction process, we randomly select 5000 pairs of synonymous nouns from WordNet synsets. For polysemous nouns we selected the synonyms for the dominant sense. The pattern extraction algorithm described in Figure 2 yields $4,562,471$ unique patterns. Of those patterns, 80% occur less than 10 times. It is impossible to train a classifier with such numerous sparse patterns. We must measure the confidence of each pattern as an indicator of synonymy. For that purpose, we employ the following procedure.

First, we run the pattern extraction algorithm described in Figure 2 with a set of non-synonymous word-pairs and count the frequency of the extracted patterns. We then use a test of statistical significance to evaluate the probable applicability of a pattern as an indicator of synonymy. The fundamental idea of this analysis is that, if a pattern appears a statistically significant number of times in snippets for synonymous words than in snippets for non-synonymous words, then it is a reliable indicator of synonymy.

To create a set of non-synonymous word-pairs, we select two

Table 1: Contingency table

	v	other than v	All
Freq. in snippets for synonymous word-pairs	p_v	$P - p_v$	P
Freq. in snippets for non-synonymous word-pairs	n_v	$N - n_v$	N

nouns from WordNet arbitrarily. If the selected two nouns do not appear in any WordNet synset then we select them as a non-synonymous word-pair. We repeat this procedure until we obtain 5000 pairs of non-synonymous words.

For each extracted pattern v, we create a contingency table, as shown in Table 1 using its frequency p_v in snippets for synonymous word-pairs and n_v in snippets for non-synonymous word-pairs. In Table 1, P denotes the total frequency of all patterns in snippets for synonymous word pairs ($P = \sum_v p_v$) and N is the same in snippets for non-synonymous word pairs ($N = \sum_v n_v$).

Using the information in Table 1, we calculate the χ^2 [7] value for each pattern as,

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

We selected the top ranking 200 patterns experimentally according to their χ^2 values. Some selected patterns are shown in Table 2.

3.3 Integrating Patterns and Page Counts

For each word-pair in the set of synonymous word-pairs we compute the four page counts-based similarity scores described in section 3.1. Moreover, we find the frequency of each lexico-syntactic pattern for this word-pair using algorithm 2. Pattern frequencies together with similarity scores form a feature vector representing the word-pair. Such feature vectors are generated for synonymous word-pairs (positive training instances) and non-synonymous word-pairs (negative training instances). We train a two class support vector machine (SVM) with the labelled training instances. Semantic similarity between two given words is defined as the posterior probability that they belong to positive (synonymous word-pairs) class. Being a large-margin classifier, output of an SVM is the distance from the decision hyper-plane. However, this is not a calibrated posterior probability. We use sigmoid functions to convert this uncalibrated distance into a calibrated posterior probability. (see [10] for a detailed discussion on this topic)

4. Experiments

4.1 The Benchmark Dataset

We evaluate the proposed method against Miller-Charles [9] dataset, a dataset of 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). Correlation with Miller-Charles' dataset has been considered as a benchmark evaluation in previous work on semantic similarity.

Features with the highest linear kernel weights are shown in Table 2 alongside with their χ^2 values. The weight of a feature in the linear kernel can be considered as a rough estimate of the influence it imparts on the final SVM output. WebDice has the highest kernel weight followed by a series of pattern-based features. WebOverlap (rank=18, weight=2.45), WebJaccard (rank=66, weight=0.618) and WebPMI (rank=138, weight=0.0001) are not shown in Table 2 because of space limitations. It is noteworthy that the pattern features in Table 2 agree with intuition. Lexical patterns (e.g., *X or Y, X and Y are, X of Y*) as well as syntax patterns (e.g., bracketing, comma usage) are extracted by our method.

Table 2: Features with the highest SVM linear kernel weights

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

We score the word pairs in Miller-Charles' dataset using the page-count-based similarity scores defined in section 3.1, Web-based semantic similarity measures proposed in previous work; Sahami [13], CODC [1], and the proposed method [*3]. Results are shown in Table 3. All figures, except those for the Miller-Charles ratings, are normalized into values in $[0, 1]$ range for ease of comparison [*4]. Proposed method earns the highest correlation of 0.834 in our experiments. Our reimplementation of Co-occurrence Double Checking (CODC) measure [1] indicates the second-best correlation of 0.6936. Similarity measure proposed by Sahami et al. [13] is placed third, reflecting a correlation of 0.5797. This method does not use page counts. Among the four page-counts-based measures, WebPMI garners the highest correlation ($r = 0.5489$). Overall, the results in Table 3 suggest that similarity measures based on snippets are more accurate than the ones based only on page counts in capturing semantic similarity.

5. Conclusion

We proposed a measure that uses both page counts and snippets to robustly calculate semantic similarity between words. Our method integrates page-counts-based similarity scores with automatically extracted lexico-syntactic patterns using support vector machines. Training data were automatically generated using WordNet synsets. Proposed method outperformed all baselines including previously proposed web-based semantic similarity measures on a benchmark dataset achieve a high correlation coefficient of 0.834 with human ratings. Proposed method does not require manually compiled taxonomies. Therefore, the proposed method can be applied in many tasks where such taxonomies do not exist or are not up-to-date. We employed the proposed method in community clustering and entity disambiguation. Experimental results indicate that the proposed method can robustly capture semantic similarity between named entities. In future research, we intend to apply the proposed semantic similarity measure in automatic

∗3 We did not use any of the words in the benchmark dataset or their synsets for training

∗4 Pearson's correlation coefficient is invariant against a linear transformation

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

Word-Pair	Miller-Charles'	Web Jaccard	Web Dice	Web Overlap	Web PMI	Sahami et al.	CODC	Proposed WebSim
automobile-car	3.92	0.654	0.668	0.834	0.427	1	0.686	0.980
journey-voyage	3.84	0.415	0.431	0.182	0.467	0.524	0.417	0.996
gem-jewel	3.84	0.295	0.309	0.094	0.687	0.211	1	0.686
boy-lad	3.76	0.186	0.196	0.601	0.631	0.471	0	0.974
coast-shore	3.7	0.786	0.796	0.521	0.561	0.381	0.518	0.945
asylum-madhouse	3.61	0.024	0.025	0.102	0.813	0.212	0	0.773
magician-wizard	3.5	0.295	0.309	0.383	0.863	0.233	0.671	1
midday-noon	3.42	0.106	0.112	0.135	0.586	0.289	0.856	0.819
furnace-stove	3.11	0.401	0.417	0.118	1	0.310	0.928	0.889
food-fruit	3.08	0.753	0.765	1	0.448	0.181	0.338	0.998
bird-cock	3.05	0.153	0.162	0.162	0.428	0.058	0.502	0.593
bird-crane	2.97	0.235	0.247	0.226	0.515	0.223	0	0.879
implement-tool	2.95	1	1	0.517	0.296	0.419	0.419	0.684
brother-monk	2.82	0.261	0.274	0.340	0.622	0.267	0.547	0.377
crane-implement	1.68	0.071	0.076	0.119	0.193	0.152	0	0.133
brother-lad	1.66	0.189	0.199	0.369	0.644	0.236	0.379	0.344
car-journey	1.16	0.444	0.460	0.378	0.204	0.189	0.290	0.286
monk-oracle	1.1	0.016	0.017	0.023	0	0.045	0	0.328
food-rooster	0.89	0.012	0.013	0.425	0.207	0.075	0	0.060
coast-hill	0.87	0.963	0.965	0.279	0.350	0.293	0	0.874
forest-graveyard	0.84	0.068	0.072	0.246	0.494	0	0	0.547
monk-slave	0.55	0.181	0.191	0.067	0.610	0.095	0	0.375
coast-forest	0.42	0.862	0.870	0.310	0.417	0.248	0	0.405
lad-wizard	0.42	0.072	0.077	0.070	0.426	0.149	0	0.220
cord-smile	0.13	0.102	0.108	0.036	0.207	0.090	0	0
glass-magician	0.11	0.117	0.124	0.408	0.598	0.143	0	0.180
rooster-voyage	0.08	0.011	0.012	0.021	0.228	0.197	0	0.017
noon-string	0.08	0.126	0.133	0.060	0.101	0.082	0	0.018
Correlation	1	0.259	0.267	0.382	0.548	0.579	0.693	0.834

synonym extraction, query suggestion and name alias recognition.

References

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

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

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

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

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

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

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

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

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

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

[11] 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.

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

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