COMPUTATION OF THE SEMANTIC RELATEDNESS BETWEEN WORDS USING CONCEPT CLOUDS

Swarnim Kulkarni and Doina Caragea
Department of Computing and Information Sciences, Kansas State University, Manhattan, Kansas, U.S.A.

Keywords: Semantic relatedness, Automatic concept extraction, Concept cloud, PageRank, Information retrieval.

Abstract: Determining the semantic relatedness between two words refers to computing a statistical measure of similarity between those words. Word similarity measures are useful in a wide range of applications such as natural language processing, query recommendation, relation extraction, spelling correction, document comparison and other information retrieval tasks. Although several methods that address this problem have been proposed in the past, effective computation of semantic relatedness still remains a challenging task. In this paper, we propose a new technique for computing the relatedness between two words. In our approach, instead of computing the relatedness between the two words directly, we propose to first compute the relatedness between their generated concept clouds using web-based coefficients. Next, we use the obtained measure to determine the relatedness between the original words. Our approach heavily relies on a concept extraction algorithm that extracts concepts related to a given query and generates a concept cloud for the query concept. We perform an evaluation on the Miller-Charles benchmark dataset and obtain a correlation coefficient of 0.882, which is better than the correlation coefficients of all other existing state of art methods, hence providing evidence for the effectiveness of our method.

1 INTRODUCTION

Building a system that can effectively determine the similarity between two words has been a problem of interest to many researchers in artificial intelligence and information retrieval areas. A robust solution to this problem would not only help in a wide range of applications such as document comparison, spell checking, community mining, but more importantly such semantic metrics would help the computers to gain a “common sense” of the information on the web.

Semantic metrics between words have been used by researchers to define *semantic relatedness, semantic similarity* and *semantic distance*, as described in (Gracia and Mena, 2008). For completeness, we provide brief definitions of these concepts here. *Semantic relatedness* considers any type of relationship between two words (including hypernymy, hyponymy, synonymy and meronymy relationships, among others) and is usually a statistical similarity measure between the two words. *Semantic similarity* is a more specialized version of semantic relatedness that considers only synonymy and hypernymy relationships between words. *Semantic distance* is a distance-based measure of semantic relatedness. That is, the more related two words are, the smaller is the semantic distance between them.

Compared to machines, humans are able to accurately determine the similarity between two words based on their common sense knowledge. For example, in order to determine that the words <*apple*, *computer*> are more closely related than the words <*apple*, *car*>, humans would use their knowledge that *apple* is the name of a company that manufactures computer hardware and software, to determine that *apple* is semantically more related to *computer* than to *car*. Our goal is to provide machines with such power by using an automated *concept cloud* based approach to determine semantic relatedness.

More precisely, our approach computes the semantic relatedness between two words by computing the similarity between their concept clouds. To automatically generate concept clouds, we use a **C**oncept **E**xtractor (CE) tool that we recently proposed in (Kulkarni and Caragea, 2009). The Concept Extractor takes as input a word query and generates a concept cloud for the query, by extracting its associated concepts from the web. Thus, for a given pair of words, we first extract the concepts associated with each word in the pair and generate their concepts

Kulkarni S. and Caragea D. (2009).
COMPUTATION OF THE SEMANTIC RELATEDNESS BETWEEN WORDS USING CONCEPT CLOUDS.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 183-188
DOI: 10.5220/0002303701830188

clouds. We then compute the semantic relatedness between the two concept clouds and use it to determine the relatedness between the initial word pair.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 gives a detailed description of the our approach for calculating semantic relatedness and its underlying algorithms. Section 4 contains the experimental results and an evaluation of our method by comparison with other similar methods. We conclude the paper in Section 5.

2 RELATED WORK

The problem of determining the semantic relatedness between two words has been an area of interest to researchers from several areas for long time. Some very preliminary approaches (Rada et al., 1989) calculated the similarity between two words on the basis of the number of edges in the term hierarchy created by indexing of articles. Similar edge-counting based methods were also applied on existing knowledge repositories such as Roget's Thesaurus (Jarmasz and Szpakowicz, 2003) or WordNet (Hirst and St-Onge, 1998) to compute the semantic relatedness.

To improve the preliminary approaches to calculating the semantic relatedness between words, more sophisticated methods have been proposed. Instead on simply relying on the number of connecting edges, Leacock and Chodorow (1998) have proposed to take the depth of the term hierarchy into consideration. Others groups have proposed to use the description of words present in dictionaries (Lesk, 1986) and techniques such as LSA (Deerwester et al., 1990) to compute semantic relatedness. However, due to the very limited size of WordNet as a knowledge base and the absence of well known named entities (e.g., *Harry Potter*) in WordNet, researchers have started to look for more comprehensive knowledge bases.

The advent of Wikipedia in 2001 has fulfilled the need for a more comprehensive knowledge base. Many techniques that use Wikipedia to compute semantic relatedness have been developed in the recent years. Among others, Strube and Ponzetto (2005) have used Wikipedia to determine semantic relatedness. Their results outperform those obtained using WordNet, hence showing the effectiveness of Wikipedia in determining the similarity between two words. Gabrilovich and Markovitch (2007) have developed a technique, called Explicit Semantic Analysis (ESA), to represent the meaning of words in a high dimensional space of concepts derived from Wikipedia. Experimental results show that ESA outperforms the method given by (Strube and Ponzetto, 2005). Chernov et al. (2006) have suggested to make use of the links between categories present on Wikipedia to extract semantic information. Milne and Witten (2008) have proposed the use of links between articles of Wikipedia rather than its categories to determine semantic relatedness between words. Zesch et al. (2008) have proposed to use Wiktionary, a comprehensive wiki-based dictionary and thesaurus for computation of semantic relatedness. Although Wikipedia has proven to be a better knowledge base than WordNet, many terms (e.g., *1980 movies*) are still unavailable on Wikipedia. This has motivated the use of the whole web as the knowledge base for calculating semantic relatedness.

Bollegala et al. (2007) have proposed to use page counts and text snippets extracted from result pages of web searches to measure semantic relatedness between words. They achieve a high correlation measure of 0.83 on the Charles-Miller benchmark dataset. Sahami and Heilman (2006) have used a similar measure. Cilibrasi et al. (2007) have proposed to compute the semantic relatedness using the normalized google distance (NGD), in which they used Google^{TM} to determine how closely related two words are on the basis of their frequency of occurring together in web documents. Chen et al. (2006) have proposed to exploit the text snippets returned by a Web search engine as an important measure in computing the semantic similarity between two words.

The approach in (Salahli, 2009) is the closest to our approach, as it uses the related terms of two words to determine the semantic relatedness between the words. However, the major drawback of the approach proposed in (Salahli, 2009) is that the related terms are manually selected. As opposed to that, our approach automatically retrieves the most relevant terms to a given word query. Furthermore, Salahli compares the related terms to the original query. In our approach, we compute the semantic relatedness between two words using the semantic similarity between their generated concept clouds. To the best of our knowledge, such an approach has not been proposed yet.

3 PROPOSED APPROACH

The steps of out proposed approach are shown in Figure 1. We use a two-phase procedure to compute the semantic relatedness between two words. The first phase involves the use of a *Concept Extractor* (Kulkarni and Caragea, 2009) to identify concepts related to the given pair of words and to generate their concept clouds. In the second phase, we use web-based coefficients (Cosine, Jaccard, Dice, Overlap) to compute

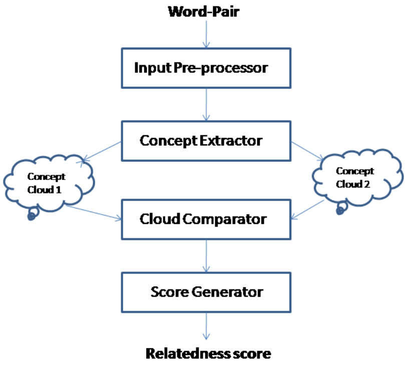

Figure 1: Steps of our proposed approach.

the semantic relatedness between the generated concept clouds, and use the resulting score to determine the relatedness between the original words. We will next describe the precise steps in our approach, and in particular, the algorithm that determines the semantic relatedness between two concept clouds.

3.1 Input Pre-Processor

The *Input Pre-Processor* module takes the given word-pair as input and divides it into two separate words. Each word is then pre-processed by converting it to lower case letters. Furthermore, any special characters such as “@”, if present, are removed from the words. Finally, the two words are provided as input to the concept extractor (one at a time).

3.2 Concept Extractor

The *Concept Extractor* module takes as input words from the *Input Pre-Processor* module and, for each word, it extracts its related concepts and forms the concept cloud by including the top k related concepts. More details about the *Concept Extractor* module can be found in (Kulkarni and Caragea, 2009).

In this paper, we use the top five related concepts to generate the concept cloud of a term. Please note that before generating the concept cloud, we perform a filtering of the related concepts. Filtering involves removal of long concepts (more than 3 words), concepts containing special characters such as “@”, and general concepts such as dictionary, web, etc.

3.3 Cloud Comparator

The function of the *Cloud Comparator* module is to perform a statistical comparison of the concept clouds corresponding to two words, using web based coefficients such as Dice and Jaccard coefficients. To achieve that, the *Cloud Comparator* computes a statistical similarity measure between all pairs of concepts $< A, B >$, where A belongs to the cloud of one term and B belongs to the cloud of the second term, and calculates the average of all similarity scores to determine the relatedness between the original words.

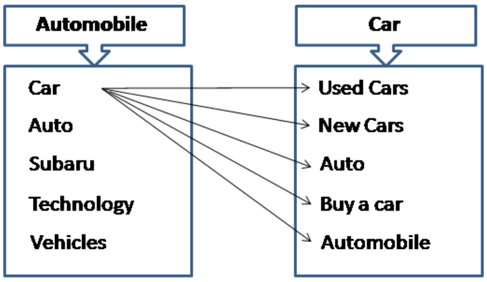

Figure 2: An example illustrating the functionality of the *Cloud Comparator* module.

We will use the example in Figure 2 to illustrate the functionality of the *Cloud Comparator* module. We assume that we want to find the semantic relatedness between the words *automobile* and *car*. After executing the first two modules in our procedure, the concept cloud for *automobile* is $\{$*Car, Auto, Subaru, Technology, Vehicles*$\}$, while the concept cloud for *car* is $\{$*Used Cars, New Cars, Auto, Buy a Car, Automobile*$\}$. The comparator takes each concept from the first cloud, e.g., *Car* and finds its relatedness to concepts in the second cloud using web-based coefficients. Preliminary experiments have shown that the Jaccard’s coefficient produces the best results. Hence, we have used the Jaccard’s coefficient to compute semantic relatedness between two concept clouds.

Traditionally, the Jaccard’s coefficient is used to determine the similarity between two given sets, A and B, by taking the ratio between the size of the intersection of the two sets and the size of the union of the two sets. That is: $Jaccard(A,B) = \dfrac{|P \cap Q|}{|P \cup Q|}$.

However, Bollegala et al. (2007) have modified the Jaccard’s coefficient definition to make it possible to compute the relatedness between two words, P and Q, using web search results. Thus:

$$Jaccard(P,Q) = \frac{H(P \cap Q)}{H(P) + H(Q) - H(P \cap Q)},$$

where $H(P)$ and $H(Q)$ refer the number of pages retrieved when the query *“P”* and the query *“Q”* are posted to a search engine, respectively; and $H(P \cap Q)$ is the number of pages retrieved when the query *“P” “Q”* is posted to a search engine.

To compute the relatedness between a term i from the first concept cloud A, denoted $con_A(i)$, and the second concept cloud B, we compute the Jaccard’s coefficient between $con_A(i)$ and all concepts $con_B(j)$ in the concept cloud B and then take the average of all scores obtained. That is:

$$rel(con_A(i), cloud(B)) = \frac{\sum_{j=1}^{n} Jaccard(con_A(i), con_B(j))}{n},$$

where n is the total number of concepts in $cloud(B)$.

Consider the example in Figure 2. The similarity between the concept *car* and the cloud *car* is computed as:

$$rel(car, cloud(car)) = \frac{\sum_{j=1}^{5} Jaccard(car, con_{car}(j))}{5}.$$

We calculate this score for each concept in first cloud A and then pass on the array of scores to the *Score Generator* module.

3.4 Score Generator

The *Score Generator* is the simplest module in our framework. It takes as input the array of scores received from the *Cloud Comparator* and computes the average of the scores to obtain a final score for the two initial words. That is,

$$score(A,B) = \frac{\sum_{i=1}^{m} rel(con_A(i), cloud(B))}{m},$$

where m refers to the total number of concepts in $cloud(A)$. It can be seen that:

$$score(A,B) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} Jaccard(con_A(i), con_B(j))}{m*n}.$$

The calculated score is reported to the user as the semantic relatedness score between the given words.

4 EXPERIMENTAL RESULTS AND EVALUATION

As the effectiveness of our method is highly dependent on the results of the *Concept Extractor* module, we start by briefly presenting the results of the *Concept Extractor* module on several types of queries. Then, we evaluate the procedure for calculating the semantic relatedness on the Miller-Charles benchmark dataset. Similar to previous work (Bollegala et al., 2007), we compute the correlation coefficient between our relatedness scores and benchmark scores and we compare the resulting correlation coefficient with the correlation coefficients reported for other similar methods. The comparison shows that our approach outperforms previous approaches reported in the literature.

4.1 Evaluation of Concept Extractor

We conducted five experiments on the *Concept Extractor* module. In the first experiment, we used a single well-defined concept as the query. In the second experiment, we provided the system with similar keywords from a specific domain. In the third experiment, we tested the system by providing the name of a person as the query. In the fourth experiment, we provided a misspelled query as input. The goal of the fifth experiment was to test the ability of the system to perform word sense disambiguation. The results for the experiments are summarized in Table 1.

4.2 Evaluation on Miller-Charles Data

We evaluated the *Cloud Comparator* module on the Miller-Charles benchmark dataset. The Miller-Charles dataset is a data set of 30 word pairs, which have been evaluated for semantic relatedness, on a scale of 0-4, by a group of 38 human subjects. Table 2 summarizes our results as well as results of previous approaches obtained from (Bollegala et al., 2007). Please note that all scores, except for the Miller-Charles scores, have been scaled to [0,1] by dividing them by the maximum score (such that the best score becomes 1). The page count and the concept data collected is as of May 4th, 2009.

The correlations between the scores obtained with each method and the benchmark scores are also reported in Table 2. As can be seen, our approach outperforms similar existing methods by achieving a high Pearson correlation coefficient of 0.882. The highest scoring pair is *midday-noon* while the lowest scoring pair is *bird-crane*. In addition to being more accurate, another advantage of our approach is that it is not dependent on a single knowledge source such as WordNet or Wikipedia and hence, has the capability to determine semantic relatedness between almost any word-pair.

5 SUMMARY AND DISCUSSION

In this paper, we have proposed a method for computing the semantic relatedness between two given words. Our approach relies on a *Concept Extractor* procedure (Kulkarni and Caragea, 2009) for finding related concepts based on web searches (i.e., concept clouds for the two words). We compute the semantic relatedness between the clouds using the web-based Jaccard coefficient. Experimental results on the Miller-Charles benchmark dataset show that our approach outperforms similar existing approaches in

Table 1: *Concept Extractor* results: related concepts extracted for a given query.

Query Type	Query	Extracted Concepts
General	Flu	Influenza, Flu, Flu Shot, Flu vaccine, Avian flu, Cold, Acetaminophen
	Kansas State University	K-State, Kansas State, CIS, Ahearn field house , Kansas, Powercat, Courses
Set-based	Mars Venus Earth	Mars, Earth, Moon, Sun, Jupiter, Mercury, Neptune
Name-based	Christopher Manning IR	Chris Manning, Christopher d. Manning, Computer Science, Data mining, Ergativity: argument structure and grammatical relations, Foundations of statistical natural language processing, Introduction to information retrieval
	Dan Brown	Author, Da vinci code, Angels and Demons, Biography, Deception point
Spelling-error	Contectiivtis	Conjunctivitis, Pinkeye, Allergic conjunctivitis, Chlamydia, Eye infection, Infectious, Bacterial conjunctivitis
Ambiguous word	Leopard OS	Apple, Mac OS X, Mac OS History, Operating System, 32-bit, 64-bit, Software
	Leopard Animal	Animals, Leopard, Snow Leopard, Mammals. Wildlife, Aardwolf

Table 2: Semantic relatedness results obtained with our approach and several related methods, by comparison with the Miller-Charles scores. The results of the methods called Web Dice and Web Overlap (Bollegala et al., 2007), Sahami (Sahami and Heilman, 2006), CODC (Chen et al., 2006) and Bollegala (Bollegala et al., 2007) are obtained from (Bollegala et al., 2007). Pearson correlation coefficients between the scores obtained with each method and the Miller-Charles scores are also reported.

Word Pair	Miller-Charles	Web Dice	Web Overlap	Sahami	CODC	Bollegala	Our approach
cord-smile	0.13	0.108	0.036	0.090	0	0	0.023
rooster-voyage	0.08	0.012	0.021	0.197	0	0.017	0.027
noon-string	0.08	0.133	0.060	0.082	0	0.018	0.034
glass-magician	0.11	0.124	0.408	0.143	0	0.180	0.027
monk-slave	0.55	0.191	0.067	0.095	0	0.375	0.029
coast-forest	0.42	0.870	0.310	0.248	0	0.405	0.078
monk-oracle	1.1	0.017	0.023	0.045	0	0.328	0.052
lad-wizard	0.42	0.077	0.070	0.149	0	0.220	0.012
forest-graveyard	0.84	0.072	0.246	0	0	0.547	0.062
food-rooster	0.89	0.013	0.425	0.075	0	0.060	0.121
coast-hill	0.87	0.965	0.279	0.293	0	0.874	0.010
car-journey	1.16	0.460	0.378	0.189	0.290	0.286	0.186
crane-implement	1.68	0.076	0.119	0.152	0	0.133	0.035
brother-lad	1.66	0.199	0.369	0.236	0.379	0.344	0.307
bird-crane	2.97	0.247	0.226	0.223	0	0.879	0.009
bird-cock	3.05	0.162	0.162	0.058	0.502	0.593	0.518
food-fruit	3.08	0.765	1	0.181	0.338	0.998	0.566
brother-monk	2.82	0.274	0.340	0.267	0.547	0.377	0.460
asylum-madhouse	3.61	0.025	0.102	0.212	0	0.773	0.849
furnace-stove	3.11	0.417	0.118	0.310	0.928	0.889	0.502
magician-wizard	3.5	0.309	0.383	0.233	0.671	1	0.493
journey-voyage	3.84	0.431	0.182	0.524	0.417	0.996	0.596
coast-shore	3.7	0.796	0.521	0.381	0.518	0.945	0.649
implement-tool	2.95	1	0.517	0.419	0.419	0.684	0.524
boy-lad	3.76	0.196	0.601	0.471	0	0.974	0.911
automobile-car	3.92	0.668	0.834	1	0.686	0.980	0.898
midday-noon	3.42	0.112	0.135	0.289	0.856	0.819	1.000
gem-jewel	3.84	0.309	0.094	0.211	1	0.686	0.884
Correlation	**1**	**0.267**	**0.382**	**0.579**	**0.693**	**0.834**	**0.882**

terms of the correlation coefficient computed with respect to the Miller-Charles benchmark scores. More precisely, we obtained a high correlation coefficient of 0.882.

The success of our approach can be explained as follows: The *Concept Extractor* forms the basis for the proposed method. The *Concept Extractor* works by extracting concepts using the top links returned by a search engine. Therefore, the extracted concepts are related to the most popular meaning of the term. If we analyze the Miller-Charles dataset, we note that the word pairs that are related through the most popular meaning of the words get a higher rating by the human subjects. For example, between the word pairs *magician-wizard* and *glass-magician*, *magician-wizard* gets a higher rating by the human subjects (3.5) as compared to *glass-magician* (0.11). This is because *wizard* is a more popular meaning of the term *magician* as compared to *glass*. Our system also assigns a higher score to *magician-wizard*, as the concepts extracted for *magician* are closer to concepts extracted for *wizard* than to concepts extracted for *glass*.

Other advantages of our approach include the following: it is not limited to a particular knowledge base such as WordNet or Wikipedia and, as a consequence, it is capable of handling a larger set of inputs; it is robust to the errors made by the user in inputs (as the search engine usually corrects misspellings); and it is able to handle multi-word queries.

Future work includes further improvements of the correlation coefficient, by improving the concept extraction process. We believe that more precise concept clouds will results in more accurate estimates for the semantic relatedness between words. Extensions of the approach to natural language processing applications are also part of our planned future work.

ACKNOWLEDGEMENTS

This work was funded by National Science foundation grant number NSF 0711396 to Doina Caragea.

REFERENCES

Bollegala, D., Matsuo, Y., and Ishizuka, M. (2007). Measuring semantic similarity between words using web search engines. In *Proc. of WWW 2007*.

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

Chernov, S., Iofciu, T., Nejdl, W., and Zhou, X. (2006). Extracting semantic relationships between wikipedia categories. In *Proc. of SemWiki2006 Workshop, colocated with ESWC2006*.

Cilibrasi, R. and Vitanyi, P. (2007). The google similarity distance. In *IEEE Transactions on Knowledge and Data Engineering*, pages 370–383.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Hashman, R. (1990). Indexing by latent semantic indexing. In *Journal of the Amer. Soc. for Inf. Science*.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI-07)*, pages 1606–1611.

Gracia and Mena (2008). Web-based measure of semantic relatedness. In *Proc. of the 9th Int. Conf. on Web Information Systems Engineering*, pages 136–150.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In *C. Fellbaum Ed.*, pages 305–332. MIT Press.

Jarmasz, M. and Szpakowicz, S. (2003). Rogets thesaurus and semantic similarity. In *Proc. of RANLP-03*, pages 212–219.

Kulkarni, S. and Caragea, D. (2009). Towards bridging the web and the semantic web. In *Proc. of WI/IAT 2009*.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In *C. Fellbaum Ed.*, pages 265–283. MIT Press.

Lesk, M. (1986). Automatic sense disambiguation using dictionaries. In *Proc. of the 5th Int. Conf. on Systems Documentation*.

Milne, D. and Witten, I. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proc. of AAAI08 Workshop on Wikipedia and Artificial Intelligence*, Chicago,IL.

Rada, R., Milli, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric to semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 17–30.

Sahami, M. and Heilman, T. (2006). A web-based kernel function for measuring the similarity of short text snippets. In *Proc. of 15th Int. WWW Conf.*

Salahli, M. A. (2009). An approach for measuring semantic relatedness via related terms. In *Mathematical and Comp. Applications*, volume 14, pages 55–63.

Strube, M. and Ponzetto, S. (2005). Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proc. of the ACL HLT Conf.*

Zesch T., Mller Christof, G. I. (2008). Using wiktionary for computing semantic relatedness. In *Proceedings of AAAI 2008*, pages 861–868.