Smart combination of web measures for solving semantic similarity problems

Jorge Martinez-gil, José F. Aldana-montes

Smart Combination of Web Measures for Solving Semantic Similarity Problems

Jorge Martinez-Gil, Jose F. Aldana-Montes

Abstract

Purpose

Semantic similarity measures are very important in many computer related fields. Previous works on applications such as data integration, query expansion, tag refactoring or text clustering have used some semantic similarity measures in the past. Despite the usefulness of semantic similarity measures in these applications, the problem of measuring similarity between two text expressions remains a key challenge.

Design/methodology/approach

In this article, we propose an optimization environment to improve the existing techniques that use the notion of co-occurrence and the information available on the Web to measure similarity between terms.

Findings

Experimental results on Miller & Charles and Gracia & Mena benchmark datasets show that the proposed approach is able to outperform classic probabilistic web-based algorithms by a wide margin.

Originality/value

We present two main contributions:

We propose a novel technique which beats classic probabilistic techniques for measuring semantic similarity between terms. This new technique consists of using not only a search engine for computing web page counts, but a smart combination of several popular web search engines.

We evaluate our approach on the Miller & Charles and Gracia & Mena benchmark datasets and compare it with existing probabilistic web extraction techniques.

Keywords: Similarity measures, Web Intelligence, Web Search Engines, Information Integration

Introduction

The study of semantic similarity between terms is an important part of a lot of computer related fields (Zhu *et al.,* 2010). Semantic similarity between terms changes over time and across domains. The traditional approach to solve this problem has consisted of using manually compiled taxonomies such as WordNet (Budanitsky *et al.*, 2006). The problem is that a lot of terms (proper nouns, brands, acronyms, new words, and so on) are not covered by dictionaries; therefore, similarity measures that are based on dictionaries cannot be used directly in these tasks (Bollegala *et al.*, 2007). However, we think that the great advances in web research have provided new opportunities for developing more accurate solutions.

In fact, with the increase of larger and larger collections of data resources on the World Wide Web (WWW), the study of web extraction techniques has become one of the most active areas for researchers. We consider that techniques of this kind are very useful for solving problems related to semantic similarity because new expressions are constantly being created and also new senses are assigned to existing expressions (Bollegala *et al.*, 2007). Manually maintaining databases to capture these new expressions and meanings is very difficult, but it is, in general, possible to find all of these new expressions in the WWW (*Yadav*, 2010).

Therefore, our approach considers that the chaotic and exponential growth of the WWW is the problem, but also the solution. In fact, we are interested in three characteristics of the Web:

- It is one of the biggest and most heterogeneous databases in the world. And possibly the most valuable source of general knowledge. Therefore, the Web fulfills the properties of Domain Independence, Universality and Maximum Coverage proposed in (Gracia & Mena, 2008).
- It is close to human language, and therefore can help to address problems related to natural language processing.
- It provides mechanisms to separate relevant from non-relevant information or rather the search engines do. We will use these search engines to our benefit.

One of the most outstanding works in this field is the definition of the Normalized Google Distance (NGD) (Cilibrasi & Vitanyi, 2007). This distance is a measure of semantic relatedness derived from the number of hits returned by the Google search engine for a given (set of) keyword(s). The idea behind this measure is that keywords with similar meanings from a natural language point of view tend to be close according to the Google distance, while words with dissimilar meanings tend to be farther apart. In fact, Cilibrasi and Vitanayi (2007) state: “We present a new theory of similarity between words and phrases based on information distance and Kolmogorov complexity. To fix thoughts, we used the World Wide Web (WWW) as the database, and Google as the search engine. The method is also applicable to other search engines and databases''. Our work is about those web search engines; more specifically, we are going to use not only Google, but a selected set of the most popular ones.

In this work, we are going to mine the Web, using web search engines to determine the degree of semantic similarity between terms. It should be taken into account that under no circumstances data from experiments presented in this work can be considered as a

demonstration that one particular web search engine is better than another or that the information it provides is more accurate. In fact, we show that the best results are obtained when weighting all of them in a smart way. Therefore, the main contributions of this work are:

- We propose a novel technique which beats classic probabilistic techniques for measuring semantic similarity between terms. This new technique consists of using not only a search engine for computing web page counts, but a smart combination of several popular web search engines. The smart combination is obtained using an elitist genetic algorithm that is able to adjust the weights of the combination formula in an efficient manner.
- We evaluate our approach on the Miller & Charles (Miller & Charles, 1998) and Gracia & Mena (Gracia & Mena, 2008) benchmark datasets and compare it with existing probabilistic web extraction techniques.

The rest of this work is organized as follows: Section 2 describes several use cases where our work can be helpful. Section 3 describes the preliminary technical definitions that are necessary to understand our proposal. Section 4 presents our contribution which consists of an optimization schema for a weighted combination of popular web search engines. Section 5 shows the data that we have obtained from an empirical evaluation of our approach. Section 6 discusses the related works and finally, Section 7 presents the conclusions and future lines of research.

Use Cases

Identifying semantic similarities between terms is not only an indicator of mastery of a language, but a key aspect in a lot of computer-related fields too. It should be taken into account that semantic similarity measures can help computers to distinguish one object from another, group them based on the similarity, classify a new object into the group, predict the behavior of the new object or simplify all the data into reasonable relationships. There are a lot of disciplines where we can get benefit from these capabilities. For example, data integration, query expansion, tag refactoring or text clustering. Now, we are going to explain why.

Data integration

Nowadays data from a large number of web pages are collected in order to provide new services. In such cases, extraction is only part of the process. The other part is the integration of the extracted data to produce a coherent database because different sites typically use different data formats (Halevy *et al*., 2006). Integration means to match columns in different data tables that contain the same kind of information (e.g., product names) and to match values that are semantically equivalent but represented differently in other sites (e.g., “cars” and “automobiles”). Unfortunately, only limited integration research has been done so far in this field.

Query Expansion

Query expansion is the process of reformulating queries in order to improve retrieval performance in information retrieval tasks (Vechtomova & Karamuftuoglu, 2007). In the context of web search engines, query expansion involves evaluating what terms were typed

into the search query area and expanding the search query to match additional web pages. Query expansion involves techniques such as finding synonyms of words (and searching for the synonyms as well) or fixing spelling errors and automatically searching for the corrected form or suggesting it in the results, for example.

Web search engines invoke query expansion to increase the quality of user search results. It is assumed that users do not always formulate search queries using the most appropriate terms. Appropriateness in this case may be because the system does not contain the terms typed by the user.

Tag refactoring

Nowadays the popularity of tags in websites is increased notably, but its generation is criticized because its lack of control causes it to be more likely to produce inconsistent and redundant results. It is well known that if tags are freely chosen (instead of taken from a given set of terms), synonyms (multiple tags for the same meaning), normalization of words and even, heterogeneity of users are likely to arise, lowering the efficiency of content indexing and searching contents (Urdiales-Nieto *et al*., 2009) . Tag refactoring (also known as tag cleaning or tag gardening) is very important in order to avoid redundancies when labeling resources in the WWW.

Text clustering

Text clustering is closely related to the concept of data clustering. Text clustering is a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval and filtering (Song *et al*., 2009).

A web search engine often returns many web pages in response to a broad query, making it difficult for users to browse or to identify relevant information. Clustering methods can be used to automatically group the retrieved web pages into a list of logical categories.

Text clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Examples of text clustering include web document clustering for search users.

Technical Preliminaries

Given two terms *a* and *b*, the problem which we are addressing consists of trying to measure the semantic similarity between them. Semantic similarity is a concept that extends beyond synonymy and is often called semantic relatedness in literature. According to Bollegala *et al*. (2007); a certain degree of semantic similarity can be observed not only between synonyms (e.g. lift and elevator), but also between meronyms (e.g. car and wheel), hyponyms (leopard and cat), related words (e.g. blood and hospital) as well as between antonyms (e.g. day and night) (Bollegala *et al*., 2007). In this work, we focus on optimizing web extraction techniques that try to measure the degree of synonymy between two given terms using the information available on the WWW.

These are the definitions for the concepts that we are going to use:

Definition 1 (Similarity measure). *A similarity measure sm is a function sm : μ_1 x μ_2 → R that associates the similarity of two input terms μ_1 and μ_2 to a similarity score sc ϵ R in the range [0, 1] which states the confidence for the relation μ_1 and μ_2 to be true .*

In this way, a similarity score of 0 stands for complete inequality of the input terms and 1 for the semantic equivalence of μ_1 and μ_2.

Definition 2 (Hit). *Hit (also known as page count) is an item found by a search engine to match specified search conditions. More formally, we can define a hit as the function hit: expr → N which associates an expression to a natural number which ascertains the popularity of expr in the subset of the WWW indexed by the search engine.*

A value of 0 stands for no popularity and the bigger the value, the bigger its associated popularity. Moreover, we want to remark that the function *hit* has many possible implementations. In fact, every web search engine implements it a different way.

Example 1. (Similarity measures based on web hits). If we look at the literature, we can find a lot of similarity measures. For example, measures based on hits which are a kind of similarity measure which are calculated based on the co-occurrence (Dagan *et al*., 2009) of the terms in the WWW, thus, the number of hits for each individual term to be compared and their conjunction. Some of the most popular measures of this kind are: Pointwise Mutual Information (PMI) (Turney, 2001), Dice, Overlap Coefficient or Jaccard (Manning & Schütze, 1999), to cite some of them. When these measures are used on the Web, it is necessary to add the prefix Web-; WebPMI, WebDice, and so on. All of them are considered probabilistic because given a web page containing one of the terms *a* or *b*, these measures try to compute the probability of that web page also containing the other term. These are their corresponding formulas:

$$WebPMI\ (a,b) = \frac{p(a,b)}{p(a)\cdot p(b)}$$

$$WebDice(a,b) = \frac{2\cdot p(a,b)}{p(a)+p(b)}$$

$$WebOverlap(a,b) = \frac{p(a,b)}{\min\big(p(a),p(b)\big)}$$

$$WebJaccard(a,b) = \frac{p(a,b)}{p(a)+p(b)-p(\mathrm{a},b)}$$

On the WWW, probabilities of term co-occurrence can be expressed by hits. In fact, these formulas are measures for the probability of co-occurrence of the terms *a* and *b* (Cilibrasi & Vitanyi, 2007). The probability of a specific term is given by the number of hits returned when a given search engine is presented with this search term divided by the overall number of web

pages possibly returned. The joint probability *p(a, b)* is the number of hits returned by a web search engine, containing both the search term *a* and the search term *b* divided by the overall number of web pages possibly returned. Estimation about the number of web pages possibly returned by a search engine has been studied in the past (Bar-Yossef & Gurevich, 2006). In this work, we set the overall number of web pages as 10^{10} according to the number of indexed pages reported in (Bollegala *et al*., 2007).

Despite its simplicity, using hits as a measure of co-occurrence of two terms presents several advantages. It is surprisingly good at recognizing synonyms because it seems to be empirically supported that synonyms often appear together in web pages (Cilibrasi & Vitanyi, 2007). Moreover, if the search terms never occur (or almost never) together on the same web pages, but do occur separately, this kind of measure will present very low similarity values.

Contribution

Traditionally web extraction techniques which are based on the notion of web hits have used a given web search engine (frequently Google) in order to collect the necessary information for their purposes. We thought that there was no compelling reason for considering Google the most appropriate web search engine in order to collect information and we began to use other search engines.

When we collected and processed the data from a wide variety of search engines, we saw that Google was the best source. However, we discovered that the average mean of the results was better than the results from Google. So we understood that maybe an optimized combination of those search engines could be even better than the average means. Based on this, we designed an experiment in order to test our hypothesis.

Figure 1 shows the solution proposed for the computation of the optimized web extraction approaches. The key of the architecture is an elitist genetic algorithm (i.e. an algorithm which selects the better individuals in each iteration), with a small mutation rate which generates a vector of numerical weights and associates them to their corresponding web search engine. The final solution can be expressed in the form of a numeric weight vector. It should be taken into account that the output of this architecture is a function that tries to simulate the behavior of the (group of) human(s) who solve the input benchmark dataset.

For the implementation of the function *hit*, we have chosen the following search engines from among the most popular in the Alexa ranking (Alexa, 2010): Google, Yahoo!, Altavista, Bing, and Ask. It should be taken into account that our approach does not include the automatic selection of appropriate web search engines because it assumes that all of them are offered initially, and those which may be associated with a weight of 0 will be automatically deselected.

The vector of numerical weights is encoded in binary format in each of the genes belonging to the domain of the genetic algorithm. For example, a vector of 20 bits can represent 1 weight of 20 bits, 2 weights of 10 bits, 4 weights of 5 bits, or 5 weights of 4 bits. It is necessary to implement a function to convert each binary weight into decimal format before to calculate the fitness for the weight vector. The number of decimal possibilities for each weight is 2^{bits}, for example, in case of choosing a vector of 5 weights of 4 bits, it is possible to represent 5

weights with 2^4 = 16 different values. These different values can range from 0 to 15, from 8 to -7, from 0 to 1 (where ticks have double precision) and so on. This depends of the implementation for the conversion from binary format into decimal format.

The comparison between the benchmark datasets and our results is made using the Pearson's Correlation Coefficient, which is a statistical measure which allows comparing two matrices of numeric values. Therefore the results can be in the interval [-1, 1], where -1 represents the worst case (totally different values) and 1 represents the best case (totally equivalent values).

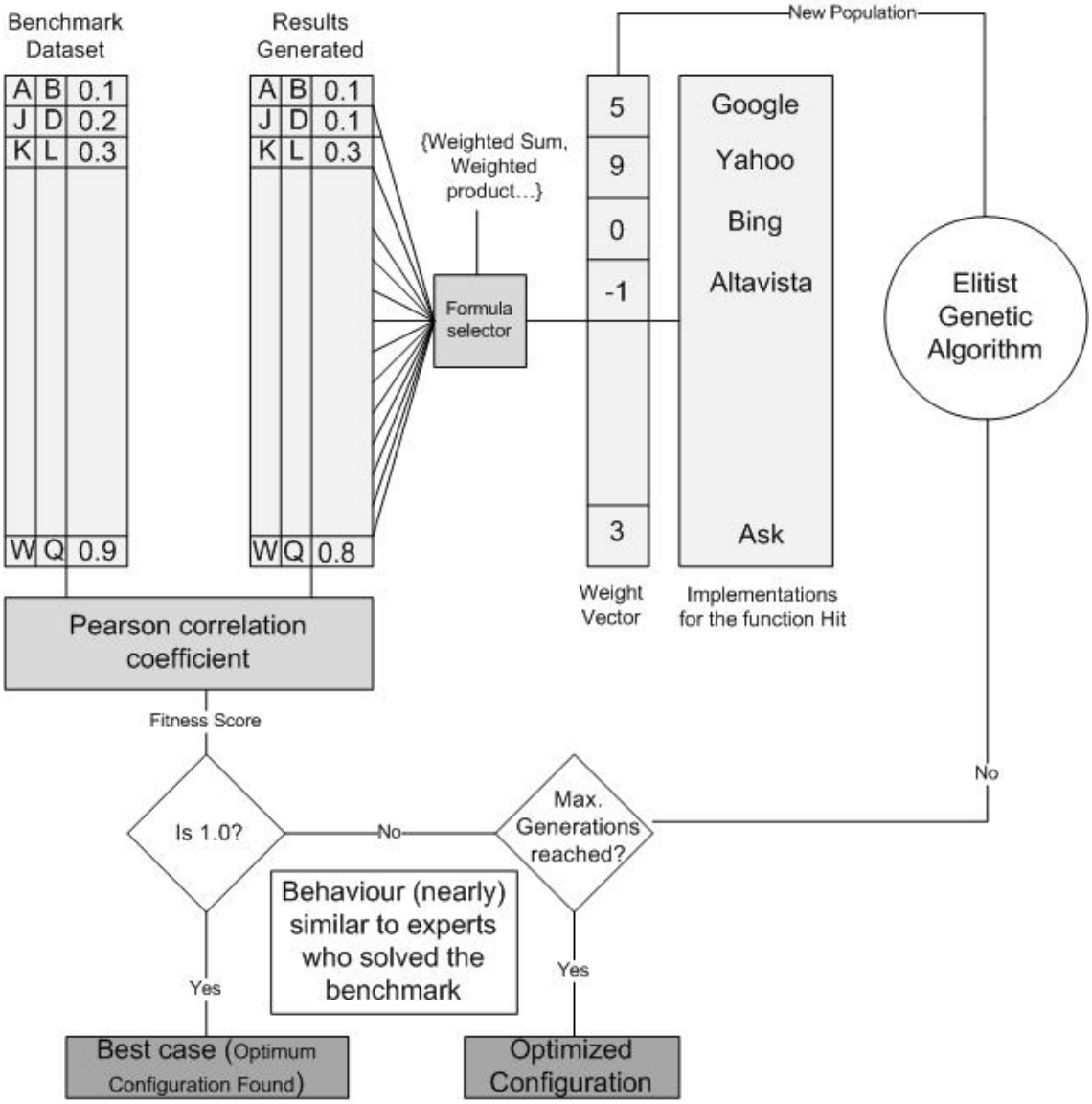

Figure 1. Solution proposed for the computation of the semantic similarity measure. An elitist genetic algorithm generates a vector of numerical weights. The final solution can be expressed in the form of a numeric weight vector.

Our approach is efficient and scalable because it only processes the snippets (no downloading of web pages is necessary) for the result pages from the web search engines. On the other hand, scalability is given by our genetic algorithm which allows us to expand the set of web search engines without causing any kind of technical problem. This is would not be possible if we would use a brute force strategy, for example.

It should be taken into account that problems related to the speed of the process, and more importantly, to the restrictions on the maximum number of allowed queries per engine, are solved by means of a preliminary phase which loads the results retrieved from the search engines into appropriate data structures. We consider that this information is valid for a period of 7 days, because the content indexed by the search engines growths dynamically.

Evaluation

In this section, we are going to show an evaluation of our approach. The section is divided in a) a preliminary study to choose the appropriate parameters for configuring the optimization environment, b) the empirical data that we have collected from the experiments, c) a study on the quality of the optimized function, d) the effect of the operators on the final result and finally, e) the information necessary to repeat the experiments.

Preliminary study

We are going to do a preliminary study of the configuration parameters for the environment.

- For the number of genes per chromosome we have selected such values as 5, 10 and 20. A study using a t-Test distribution has shown us that that the differences between samples obtained by means of these parameters are not statistically significant. Therefore, we have selected 20 genes per chromosome.
- For the number of individuals in the population, we have selected such values as 20, 50 and 100. Again, a t-Test statistical distribution has shown that the differences between these samples are not statistically significant. We have selected a population of 100 individuals.
- Related to crossover and mutation fraction, we have chosen a high value for the crossover between genes and, a small percentage for mutations, because we wish a classical configuration for the genetic algorithm.
- Related to the combination formula, our preliminary study has shown us that the weighted sum is better that the weighted product and the weighted square.

After ten independent executions, we noticed that the genetic algorithm did not improve the results beyond the 200th generation, so we have set a limit of 200 generations for the algorithm. The final results are computed by means of the average mean from these 10 generations. We are going to evaluate our approach using the Miller & Charles benchmark dataset which is a dataset of term pairs rated by a group of 38 human beings (Miller & Charles, 1998). Term pairs are rated on a scale from 0 (no similarity) to 4 (complete similarity). Miller & Charles dataset benchmark is a subset of Rubenstein & Goodenough original benchmark dataset of 65 term pairs (Rubenstein & Goodenough, 1965). Although Miller & Charles experiment was carried out many years later than Rubenstein & Goodenough, Bollegala *et al*. (1965) state that two sets of ratings are highly correlated (Bollegala *et al*., 2007). Therefore, Miller & Charles ratings can be considered as a good benchmark dataset to evaluate solutions that involve semantic similarity measures.

Results

In this subsection we are going to present the empirical results that we have obtained from our experiments. It should be taken into account that all figures, except those for the Miller &

Charles ratings, are normalized into values in [0, 1] range for ease of comparison. Pearson's correlation coefficient is invariant against a linear transformation (Bollegala *et al*., 2007).

Table 1 shows the collected data using several search engines for solving the Miller & Charles benchmark dataset. The measure that we have used is NGD (Cilibrasi & Vitany, 2007). As we commented previously, Google is the best search engine for our purposes; however, the average mean and the median present even better results. Other kinds of statistical measures such as mode, maximum and minimum have been considered because their associated results have not been better.

Table 1. Summary results for Miller & Charles benchmark using several search engines

	Mil.Char.	Google	Ask	Altavista	Bing	Yahoo	Average	Median
cord-smile	0.13	0.05	0.25	1.00	0.00	0.00	0.26	0.05
rooster-voyage	0.08	0.24	1.00	0.00	0.00	0.00	0.25	0.00
noon-string	0.08	0.50	1.00	0.00	0.00	0.00	0.30	0.00
glass-magician	0.11	1.00	1.00	1.00	0.00	0.01	0.60	1.00
monk-slave	0.55	1.00	1.00	1.00	0.00	0.00	0.60	1.00
coast-forest	0.42	1.00	1.00	1.00	0.02	0.01	0.61	1.00
monk-oracle	1.10	1.00	1.00	1.00	0.00	0.00	0.60	1.00
lad-wizard	0.42	0.04	1.00	1.00	0.00	0.00	0.41	0.04
forest-graveyard	0.84	1.00	1.00	1.00	0.00	0.01	0.60	1.00
food-rooster	0.89	1.00	1.00	1.00	0.00	0.00	0.60	1.00
coast-hill	0.87	1.00	1.00	1.00	0.06	0.02	0.62	1.00
car-journey	1.16	0.17	1.00	1.00	0.01	0.00	0.44	0.17
crane-implement	1.68	0.14	1.00	0.00	0.00	0.00	0.23	0.00
brother-lad	1.66	0.18	1.00	1.00	0.00	0.00	0.44	0.18
bird-crane	2.97	1.00	1.00	1.00	0.13	0.09	0.64	1.00
bird-cock	3.05	1.00	1.00	1.00	0.07	0.07	0.63	1.00
food-fruit	3.08	1.00	1.00	1.00	0.03	0.01	0.61	1.00
brother-monk	2.82	1.00	1.00	1.00	0.11	0.04	0.63	1.00
asylum-madhouse	3.61	1.00	1.00	1.00	0.00	0.00	0.60	1.00
furnace-stove	3.11	0.46	1.00	1.00	0.00	1.00	0.69	1.00
magician-wizard	3.50	1.00	1.00	1.00	0.04	0.98	0.80	1.00
journey-voyage	3.84	1.00	1.00	1.00	0.00	0.00	0.60	1.00
coast-shore	3.70	1.00	1.00	1.00	0.02	0.08	0.62	1.00
implement-tool	2.95	1.00	1.00	1.00	0.00	0.02	0.60	1.00
boy-lad	3.76	1.00	1.00	1.00	0.18	0.02	0.64	1.00
automobile-car	3.92	1.00	1.00	1.00	0.01	0.34	0.67	1.00
midday-noon	3.42	1.00	1.00	1.00	0.07	0.00	0.61	1.00
gem-jewel	3.84	1.00	1.00	1.00	0.39	0.05	0.69	1.00
Correlation	**1.00**	**0.47**	**0.26**	**0.35**	**0.43**	**0.34**	**0.61**	**0.54**

Figure 2 shows us the behavior of the average mean in relation to the Miller & Charles benchmark dataset. As it can be seen the behavior is quite similar, however, it can be improved by using more elaborated formulas than the average mean, as we show in our second experiment.

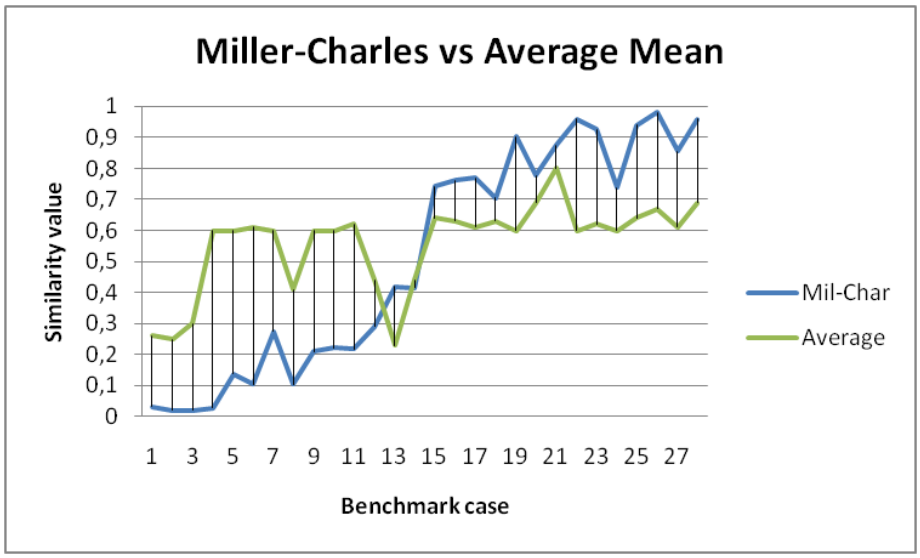

Figure 2. Comparison between Miller & Charles benchmark dataset and average mean from several web search engines

Meanwhile, Table 2 shows us the results from several probabilistic web extraction algorithms described previously. The data represent the numerical values obtained before and after the optimization process. The classic score is obtained using only the Google search engine, while the optimized score is obtained by means of the Pearson's Correlation Coefficient obtained after 200 generations from an elitist genetic algorithm that tries to maximize the weighted sum of some of the most popular web search engines.

Table 2. Results from probabilistic web extraction algorithms before and after the optimization process for Miller & Charles Dataset

Measure	Classic Score	Optimized Score	Improvement
WebPMI	0.548	0.678	23.72 %
WebDice	0.267	0.622	132.95 %
WebOverlap	0.382	0.554	45.02 %
WebJaccard	0.259	0.565	118.14 %

Lastly, Figure 3 represents a histogram with the results that clearly show us that our approach significantly outperforms classic probabilistic web extraction techniques. In some cases, there is an improvement of more than two times the original score.

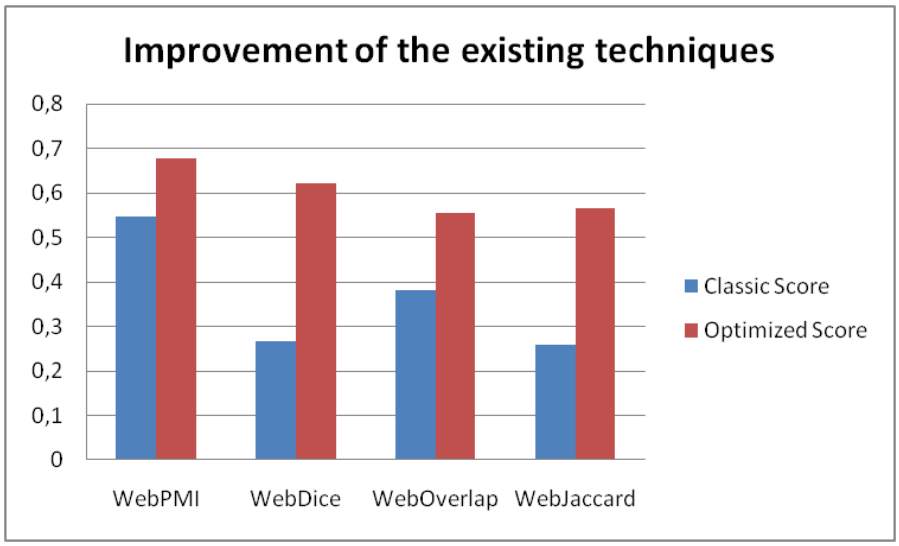

Figure 3. Existing techniques can be significantly improved by using our approach

Although our hypothesis has been tested using measures based on web hits, we see no problem in using other kind of web extraction techniques, and thus, the optimization environment can be used to optimize other kinds of web extraction techniques if these techniques are susceptible to be parameterized.

Quality of the obtained function

Maybe the most important fact of this work is to determine if the trained optimal weights for the previous dataset can maintain the same level of improvement for other term pairs. In order to clarify this, we are going to use the function optimized using the Miller & Charles benchmark dataset for solving a different benchmark.

Table 3. Quality of the optimized function when solving the Gracia-Mena benchmark. Gracia-Mena column shows the results from Google, Gracia-Mena' shows the results from the smart function obtained in the previous experiment

Measure	Gracia-Mena	Gracia-Mena'	Improvement
WebPMI	0.548	0.571	5.35 %
WebDice	0.520	0.534	2.69 %
WebOverlap	0.521	0.538	3.26 %
WebJaccard	0.528	0.596	12.88 %

We have chosen the Gracia & Mena benchmark dataset (Gracia & Mena, 2008). This benchmark dataset is similar to our previous dataset. It contains 30 pairs of English nouns, where some kind of relationship are present in most of them: similarity (person, soul), meronymy (hour, minute), frequent association (penguin, Antarctica), and others.

As it can be seen in Table 3, quality of the optimized function when solving the Gracia-Mena benchmark dataset is very good, or at least, it is better to use the optimized function that the classical approach.

Effect of the operators

In our experiments we have chosen the weighted sum because the results are, in general better, than the results for the weighted product or the weighted square. The effects of the operators in the optimization environment can be seen in the Table 4.

The weighted sum is the most appropriate operator in all cases, the reason is that values are always better that those obtained using a weighted product or a weighted square. This is the reason why we have chosen the weighted sum as the operator for our optimization environment.

Table 4. Effect of the operators in the final results from the optimization environment

Measure	Weighted Sum	Weighted Product	Weighted Square
WebPMI	0.678	0.242	0.348
WebDice	0.622	0.240	0.413
WebOverlap	0.554	0.245	0.278
WebJaccard	0.565	0.372	0.337

As future work, we can try to test the effect of other kinds of statistical measures such as: median, mode, maximum or minimum for the values retrieved from the different search engines, and so on. These measures have not been considered in this work, because they have not an associated numeric value which can be optimized using our approach.

Data to repeat the experiments

Related to the conditions of the experiment, we have used:

As web search engines for implementing the function Hit: *{Google, Ask, Altavista, Bing, Yahoo}*

The elitist genetic algorithm has been configured taking the following parameters into account[1]:

- 20 genes per chromosome
- A population of 100 individuals
- 0.98 for crossover fraction
- 0.05 for mutation fraction
- We allow 200 generations
- The goal is to optimize the weighted sum

We have chosen the weighted sum because the results are, in general better, that the results for the weighted product.

It should be taken into account that in case of repeating the experiments, results from the experiments can vary slightly along the time because the content indexed by the web search engines is not static.

[1] Fitness and search space have been explained in the previous section

Related Work

In addition to its multiple applications in the Natural Language Processing field, it is widely accepted that similarity measures are essential to solve many problems such as classification, clustering, and retrieval problems. For this reason, several works have been developed over the last few years proposing different ways to measure semantic similarity (Budanitsky *et a*l., 2006). According to the sources exploited and the way in which they are used, different families of methods can be identified. These families are: Taxonomies of concepts, Feature-based approaches, and the Web as corpus paradigm.

Related to taxonomies of concepts, the most popular method for calculating similarity between two words consists of finding the length of the shortest path connecting the two words in a taxonomy (Budanitsky *et al*., 2006). If a word has two o more meanings, then multiple paths may exist between the two words. In such cases, it is possible to choose the shortest path between any two meanings of the words (optimistic approach) or the largest path between them (pessimistic approach). A problem frequently acknowledged with this approach is that it relies on the notion that all links in the taxonomy represent uniform distances.

> An advantage of our method compared to the taxonomy based semantic similarity measures is that our method requires no taxonomies for computing the semantic similarity. Therefore, the proposed method can be applied in many tasks where such taxonomies do not exist or are not up-to-date.

Related to feature-based approaches, it is possible to estimate semantic similarity according to the amount of common and non common features (Petrakis *et al*. 2006); by features authors typically consider taxonomical information found in an ontology and concept descriptions retrieved from dictionaries.

> The problem of feature-based approaches is that the feature detection stage requires features to be located accurately and reliably. This is a non-trivial task. On the other hand, it is not easy to determine if two words share a common feature, this is due to the problem of feature matching. Our approach does not require feature detection or matching.

Regarding the Web as a knowledge corpus (Lapata & Keller, 2005), unsupervised models demonstrably perform better when n-gram counts are obtained from the Web rather than from other corpus (Keller & Lapata, 2003). In fact, Resnik and Smith extracted sentences from the Web to create parallel corpora for machine translation (Resnik & Smith, 2003). We have identified two research lines related to the use of web measures: 1) measures based on web hits, and 2) measures based on text snippets.

- Regarding web hits, we have seen that one of the best measures was introduced by Turney and it consists of a point-wise mutual information measure using the number of hits returned by a web search engine to recognize synonyms (Sanchez *et al*., 2010). On the other hand, Bollegala *et al*. have proposed several works which use web hits for measuring semantic similarity (Bollegala et al., 2007), mining personal names aliases (Bollegala *et al*. 2008), or measuring relational similarity (Bollegala *et al*. , 2009).

- Related to text snippets, it should be taken into account that snippets are brief windows of text extracted by a search engine around the query term in a document, provide information regarding the query term. Works addressing this issue were proposed by Sahami and Heiman (2006) who measured the semantic similarity between two queries using snippets returned for those queries. For each query, they collect snippets from a search engine and represent each snippet as a TF-IDF-weighted vector. Or Chen *et al*. who proposed a double-checking model using text snippets returned by a web search engine to compute semantic similarity between words (Chen *et al*., 2006).

Conclusions

In this work, we have presented a novel approach that generalizes and extends previous proposals for determining the semantic similarity between terms using web search engines. The optimization environment consists of several web extraction techniques which use a selected set of simple web search engines (instead of a single web search engine) which are combined in order to achieve satisfactory results.

We have provided an analysis of the most popular algorithms for extracting web information by using web hits, and characterized their relative applicability as black boxes in our optimization environment. It is necessary to bear in mind that the success of our approach depends largely on the kind of the underlying simple web search engines included and the heterogeneity and soundness of the benchmark datasets for determining their associated weights.

We have implemented a prototype for our optimization schema following an approach based on Genetic Algorithms that is highly scalable, thus, it can be expanded with a lot of simple web search engines. Our proposal has been optimized using a widely used benchmark dataset (Miller & Charles, 1998) and applied to solve another widely used but different benchmark dataset (Gracia & Mena, 2008). We have shown us that our approach significantly outperforms the classical probabilistic techniques by a wide margin when solving the two datasets.

As future work, we propose a comparison of the knowledge provided by online encyclopedias and that provided by the web search engines. The idea behind this proposal is to use not only simple measures liked those based on web hits, but to benefit from the structured nature of these encyclopedias in order to use and optimize more complex web extraction techniques. The goal is to further improve semantic similarity detection techniques. In this way, the semantic interoperability between people, computers or simply agents in the WWW might become true.

References

Alexa rank. Available from <http://www.alexa.com> [15-feb-2010].

Bollegala, D., Matsuo, Y., Ishizuka, M. (2007), "Measuring semantic similarity between words using web search engines". *Proceedings of the World Wide Web Conference*, pp. 757-766.

Bollegala, D., Honma, T., Matsuo, Y., Ishizuka, M. (2008), "Mining for personal name aliases on the web". *Proceedings of the World Wide Web Conference*, pp. 1107-1108.

Bollegala, D., Matsuo, Y., Ishizuka, M. (2009), "Measuring the similarity between implicit semantic relations from the web". *Proceedings of the World Wide Web Conference*, pp. 651-660.

Bar-Yossef, Z., Gurevich, M. (2006), "Random sampling from a search engine's index". *Proceedings of the World Wide Web Conference*, pp. 367-376.

Budanitsky, A., Hirst, G. (2006), "Evaluating WordNet-based Measures of Lexical Semantic Relatedness". *Computational Linguistics* vol. 32, no. 1, pp. 13-47.

Chen H., Lin, W. (2006) Novel association measures using web search with double checking. *Proceedings of COLING/ACL*, pp. 1009-1016.

Cilibrasi, R., Vitanyi, PM. (2007), "The Google Similarity Distance". *IEEE Trans. Knowl. Data Eng*, vol. 19, no. 13, pp. 370-383.

Dagan, I., Lee, L., Pereira, F. (1999), "Similarity-based models of word co-occurrence probabilities". *Machine Learning,* vol. 34, no. 1, pp. 43–69.

Gracia, J., Mena, E. (2008), "Web-Based Measure of Semantic Relatedness". *Proceedings of WISE,* pp. 136-150.

Halevy, A., Rajaraman, A., Ordille, J. (2006), "Data integration: The teenage years". *Proceedings of VLDB*, pp. 9-16.

Keller, F., Lapata, M. (2003), "Using the web to obtain frequencies for unseen bigrams". *Computational Linguistics* vol. 29, no. 3, pp. 459-484.

Lapata, M., Keller, F. (2005), "Web-based models of natural language processing". *ACM Transactions on Speech and Language Processing* vol. 2, no. 1, pp. 1-31.

Manning, CD., Schütze, H. (1999), *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts, MIT Press.

Miller, G., Charles, W. (1998), "Contextual correlates of semantic similarity". *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1-28.

Petrakis, EGM., Varelas, G., Hliaoutakis, A., Raftopoulou, P. (2006), "X-Similarity:Computing Semantic Similarity between Concepts from Different Ontologies". *Journal of Digital Information Management* vol. 4, pp. 233-237.

Resnik, P., Smith, NA. (2003), "The web as a parallel corpus". *Computational Linguistics*, vol. 29, no. 3, pp. 349-380.

Rubenstein, H., Goodenough, JB. (1965), "Contextual correlates of synonymy". *Communications of the ACM*, vol. 8, pp. 627–633.

Sahami, M., Heilman, T. (2006), "A web-based kernel function for measuring the similarity of short text snippets". *Proceedings of the World Wide Web Conference*, pp. 377-386.

Sanchez, D., Batet, M., Valls, A., Gibert, K. (2010), “Ontology-driven web-based semantic similarity”. *J. Intell. Inf. Syst*, vol. 35, no. 3, pp. 383-413.

Song, W., Hua Li, C., Cheol Park, S. (2009), “Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures”. *Expert Syst. Appl.* vol. 36, no. 5, pp. 9095-9104.

Turney, PD. (2001), “Mining the web for synonyms: Pmi-ir versus lsa on toefl”. *Proceedings of ECML*, pp. 491-502.

Urdiales-Nieto, D., Martinez-Gil, J., Aldana-Montes, JF. (2009), “MaSiMe: A Customized Similarity Measure and Its Application for Tag Cloud Refactoring”. *Proceedings of OTM Workshops*, pp. 937-946.

Vechtomova, O., Karamuftuoglu, M. (2007), “Query expansion with terms selected using lexical cohesion analysis of documents”. *Inf. Process. Manage*, vol. 43, no. 4, pp. 849-865.

Yadav, SB. (2010), “A conceptual model for user-centered quality information retrieval on the World Wide Web”. *J. Intell. Inf. Syst*, vol. 35, no. 1, pp. 91-121.

Zhu, L., Ma, Q., Liu, C., Mao, G., Yang, W. (2010), “Semantic-distance based evaluation of ranking queries over relational databases”. *J. Intell. Inf. Syst*, vol. 35, no. 3, pp. 415-445.