Data & Knowledge Engineering 70 (2011) 390–405

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

SyMSS: A syntax-based measure for short-text semantic similarity

Jesús Oliva [*], José Ignacio Serrano, María Dolores del Castillo, Ángel Iglesias

Bioengineering Group, CSIC, Carretera de Campo Real, km. 0,200. La Poveda, Arganda del Rey, CP: 28500, Madrid, Spain

ARTICLE INFO

Article history:
Received 30 July 2009
Received in revised form 12 January 2011
Accepted 14 January 2011
Available online 22 January 2011

Keywords:
Linguistic tools for IS modeling
Text DBs
Natural language processing (NLP)
Semantic similarity
Sentence similarity

ABSTRACT

Sentence and short-text semantic similarity measures are becoming an important part of many natural language processing tasks, such as text summarization and conversational agents. This paper presents SyMSS, a new method for computing short-text and sentence semantic similarity. The method is based on the notion that the meaning of a sentence is made up of not only the meanings of its individual words, but also the structural way the words are combined. Thus, SyMSS captures and combines syntactic and semantic information to compute the semantic similarity of two sentences. Semantic information is obtained from a lexical database. Syntactic information is obtained through a deep parsing process that finds the phrases in each sentence. With this information, the proposed method measures the semantic similarity between concepts that play the same syntactic role. Psychological plausibility is added to the method by using previous findings about how humans weight different syntactic roles when computing semantic similarity. The results show that SyMSS outperforms state-of-the-art methods in terms of rank correlation with human intuition, thus proving the importance of syntactic information in sentence semantic similarity computation.

1. Introduction

Studies of semantic similarity to date have focused on one of two levels of detail, either single words or complete documents. Sentence semantic similarity is also known as short-text semantic similarity (STSS) [1], and is used to measure the similarity of texts that are typically 10–20 words long, not necessarily grammatically complete sentences. The importance of sentence semantic similarity measures in natural language research is increasing due to the great number of applications that are arising in many text-related research fields. For example, in text summarization, sentence semantic similarity is used (mainly in sentence-based extractive document summarization) to cluster similar sentences and then find the sentence that best represents each cluster [2]. In web page retrieval, too, sentence similarity can enhance effectiveness by calculating similarities of page titles [3]. Document retrieval applications can also improve their performance using semantic similarity related techniques [4]. Moreover, conversational agents can also benefit from the use of sentence similarity measures to reduce the scripting process by using natural language sentences instead of sentence patterns [5,6]. These are only a few examples of applications whose effectiveness could improve with sentence semantic similarity calculation.

One of the main problems of existing sentence similarity methods is that most of them are adaptations of long-text similarity methods, and methods of this type are not suitable for coping with the problem of sentence similarity, because long-text similarity methods are not designed to address this problem and the information present at the sentence level and at textual level is not the same. For example, at textual level, word co-occurrence can be an important source of information, but it is less significant when dealing with sentences because of the small number of words present in a sentence. However, the syntactic information that can be extracted at the sentence level is highly relevant in the computation of sentence semantic similarity. Based on the hypothesis

[*] Corresponding author. Tel.: +34 91 8711900; fax: +34 91 8717050.
E-mail addresses: jesus.oliva@car.upm-csic.es (J. Oliva), Ignacio.serrano@car.upm-csic.es (J.I. Serrano), dolores.delcastillo@car.upm-csic.es (M.D. del Castillo), angel.iglesias@car.upm-csic.es (Á. Iglesias).

0169-023X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2011.01.002

that the meaning of a sentence is made up of not only the meanings of its individual words, but also the structural way the words are combined (principle of semantic compositionality [7]), in this paper we propose a novel approach, SyMSS (Syntax-based Measure for Semantic Similarity), to evaluate the semantic similarity between short texts and sentences by taking into account semantic and syntactic information.

SyMSS obtains semantic information from a lexical resource such as WordNet [8],[1] which allows different types of measures of the semantic similarity between concepts to be applied. Syntactic information is obtained through a deep parsing process that obtains the phrases that make up the sentence and their syntactical functions. With this information, our proposed method measures the semantic similarity between concepts that have the same syntactic function. The use of syntactic information and WordNet is also widely spread and used in applications such as text summarization [9] or document clustering [10].

In addition to presenting SyMSS, in this paper we share a comparative study of seven variations of the proposed method, based on seven different concept similarity measures. We also study the importance of adjectives and adverbs in sentence semantic similarity, comparing their importance with that of other major parts of speech, such as nouns and verbs. Furthermore, we endeavor to add psychological plausibility to the method; i.e. we use the results of psychological experiments to pattern our method after the way humans compute sentence semantic similarity. We particularly use, the results obtained by Wiemer-Hastings [11] that pointed out the different values humans give to different syntactic roles when computing sentence semantic similarity. Taking into account this information, we present a second version of SyMSS that weights the different syntactic roles the way humans do.

We use two benchmark data sets to evaluate the method. The first data set, proposed by Li et al. [12], contains human ratings for the similarity of 65 pairs of sentences. We select the same 30 pairs of sentences that they did in order to enable comparisons with human ratings and with the methods proposed by Li et al. [12] and Islam and Inkpen [13].[2] Furthermore, we use the Microsoft Paraphrase Corpus to evaluate the method with a larger corpus and in a much more challenging task. The results obtained by the method proposed in this paper outperform the best results obtained by the Li–McLean method in terms of rank correlation with human intuition. Moreover, the Islam–Inkpen method is also outperformed in terms of rank correlation, highlighting the importance of syntactic structure in semantic similarity.

The next section presents a brief review of the most common approaches used to compute sentence semantic similarity. Section 3 gives the details of the proposed method of measuring sentence semantic similarity. Section 4 covers the experiments, providing the similarity ratings given by the different variations of SyMSS implemented and comparing them with state-of-the-art similarity measures and human ratings. Section 5 is devoted to the study of the effects of the addition of psychological plausibility to the method. Lastly, Section 6 sums up the work and poses some conclusions and future work.

2. Background

Most previous work on sentence or short-text similarity is based on adaptations of long-text similarity methods. The problem is that these methods need adequate information in order to perform well, and in most cases they cannot find adequate information in single sentences or short texts. For example, two long, similar texts are likely to have enough co-occurring words, but, at the sentence level, two very similar sentences might easily fail to share even a single word. Three main kinds of methods are used to compute semantic similarity: corpus-based methods, word co-occurrence methods and hybrid methods.

Corpus-based methods use statistical information on words in a corpus. Perhaps the most important method of this type is Latent Semantic Analysis (LSA). LSA uses a word by passage matrix formed to reflect the presence of words in each of the passages used. This matrix is decomposed by singular value decomposition, and its dimensionality is reduced by removing small singular values. Finally, the sentences to be compared are represented in this reduced space as two vectors containing the meaning of their words. The final similarity is computed as the similarity of these two vectors (See [14,15] for a complete explanation of the method). The main disadvantage of LSA applied to the computation of sentence similarity is the lack of potentially important syntactic information, as will be shown in this paper. For example, the sentences “The dog chased the man” and “The man chased the dog” are viewed as identical by LSA. Also, negations and antonyms are not processed by LSA, which considers that the sentences “He is a doctor” and “He is not a doctor” are very similar. Another well-known method closely related with LSA is Hyperspace Analogs to Language (HAL), although experimental results show that this method is less suitable for computing the semantic similarity of short texts or sentences [16]. Also, Islam and Inkpen [13] present a corpus-based method that finds the similarity of two texts using three similarity functions: a string similarity method between words, a semantic similarity measure between words and a common word order similarity measure.

Word co-occurrence-based methods are the most frequently used methods in applications such as information retrieval, and many such methods have been adapted to compute sentence similarity. Although LSA and HAL do use word co-occurrence, their key feature is their use of corpora, which enables them to find similarity in sentences with no co-occurring words. The main disadvantages of word co-occurrence methods in the sentence domain are their failure to use syntactic information and the sparseness of the vector representation. Although in long texts the vector representation is not sparse because of the great amount of words long texts contain, at the sentence level the number of words is too small. Besides, these methods can overlook very

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

[2] From here, the sentence similarity methods proposed by Li et al. [12] and Islam and Inkpen [13] will be called the Li–McLean method and the Islam–Inkpen method respectively.

similar sentences if the sentences have no words in common; and, vice-versa, they can find a high similarity between sentences with many co-occurring words but not actually very similar, as in:

> "My brother has a dog with four legs" and "My brother has four legs"

However, there have been some proposals for improving word co-occurrence methods, such as pattern matching methods used in text mining [17]. The problem with approaches of this kind is that they require a complete pattern set for each meaning of a word. Manual pattern set compilation is an arduous task, and there is no automated method for doing it.

There are also some hybrid methods that use both corpus-based and knowledge-based techniques. The best-performing method of this kind is the one proposed by Li et al. [12], which endeavors to overcome the limitations of both techniques by forming the word vector entirely on the basis of the words in the compared sentences. Then the method computes the semantic similarity by combining information drawn from a structured lexical database and from corpus statistics. Also Mihalcea et al. [18], proposed a combined unsupervised method that uses six WordNet-based measures and two corpus-based measures and combine the results to show how these measures can be used to derive a short texts similarity measure. The main drawback of this method is that it computes the similarity of words using eight different methods, which is not computationally efficient.

One of the problems that all the approaches discussed above share is that they do not take syntactic information into account. There are few methods that consider pseudo-syntactic information, such as word order in the sentence. The scheme proposed by Achananuparp et al. [19] and the Li–McLean and Islam–Inkpen methods use this kind of information; theirs are the best results reported in the literature in terms of correlation with human intuition, thus confirming that syntactic information is of great importance in the computation of sentence semantic similarity. The method proposed here aims to outperform these approaches by using high-level syntactic information, such as complete parse trees of the compared sentences, thus revealing the influence of in-depth syntactic information.

A very related task to the computation of semantic similarity is recognizing textual entailment (RTE) (see Dagan et al. [20] for an explanation of the PASCAL RTE challenge). However, RTE has many differences with semantic similarity computation. RTE targets the identification of a directional inferential relation between two texts. This means that recognizing textual entailment is an asymmetric task. For example, "I have two dogs" entails "I have a dog" but "I have a dog" does not entail "I have two dogs". However, semantic similarity computation is a symmetric task given that the semantic similarity between two sentences does not depend on the order of processing of the sentences. Moreover, semantic similarity is not enough to compute textual entailment. A high semantic similarity score is not enough to assess textual entailment. For example, the sentences "I have a dog" and "I have a cat" are very similar but none of them entails the other. These two main differences show us that recognizing textual entailment and computing semantic similarity are two different tasks and that sentence semantic similarity measures could be an important step in the textual entailment recognition task but not always sufficient.

Despite the differences pointed, there are many things that can be learned from one task to the other. For example, Corley and Mihalcea [21] define a directional measure of similarity to avoid the problem of asymmetry that we commented before. Their approach is based on a similarity measure between words using WordNet. As expected, the results obtained by Corley and Mihalcea show that semantic similarity measures could represent a first approach to the entailment problem but it is necessary to combine them with other techniques to achieve better results. One of these techniques could be the use of different kinds of representations of sentence structure. For example syntactic information can be very useful in both semantic similarity measures and textual entailment recognition methods. Vanderwende and Dolan [22] showed that 390 pairs of sentences out of the 800 of the test set used on the first PASCAL challenge can be classified using solely syntactic cues. Despite this and other similar observations about the importance of syntactic information to measure semantic similarity, not much work has been done to use deep syntactic information on semantic similarity measures. The combination of semantic similarity measures between words and syntactic information seems to be a promising research line for RTE. Following this approach we can find the work of Bar-Haim et al. [23] which states that lexical and syntactic levels are complementary for RTE. Also Zanzotto et al. [24] follow this line presenting a supervised machine learning algorithm that uses syntactic and shallow semantic feature spaces. Moreover, in the different PASCAL challenges can be found many systems that combine syntax and semantics. For example the UNED-NLP Group obtained great results in most of these challenges (Herrera et al. [25]; Herrera et al. [26], Rodrigo et al. [27]) using dependency trees and WordNet-based semantic similarity measures.

Given the promising results obtained on the entailment task by the combination of semantic and syntactic information, our method captures and combines syntactic and semantic information to compute the semantic similarity of two sentences. The way of combining this information is by measuring the semantic similarity between concepts that play the same syntactic role.

3. Sentence similarity method

The SyMSS method captures how the syntactic structure of the compared sentences influences the calculation of semantic similarity. This is based on the notion that the meaning of a sentence is made up of not only the meanings of its individual words, but also the structural way the words are combined. As we have seen in the example above, the meanings of two words may be exactly the same (or the words may even, be the same), but the different syntactical role the words play in two different sentences may denote two very different meanings.

Semantic information is obtained from WordNet [8], whose structure enables different types of measures of semantic similarity between concepts to be calculated. Syntactic information is obtained through a deep parsing process that finds the phrases (i.e., groups of words that function as a single unit in the syntax of a sentence) that make up the sentence as well as the phrases' syntactic functions. With this information, the proposed method measures the semantic similarity between concepts that perform the same syntactic function.

3.1. Semantic similarity between concepts

As many other methods do (see [11,17]), SyMSS finds the semantic similarity between concepts drawn from WordNet, using its hierarchical structure and the different glosses associated with each term. Similarity between concepts is the basic unit of similarity used by the sentence similarity method, so using a poor similarity measure at this point could reduce the overall performance of the proposed method. Thus, a comparative study of different measures of the semantic similarity between concepts was carried out to find the most suitable measure for our method. Three types of measures were compared (see [19] for a detailed explanation of these measures):

- Path-based measures
  Taking advantage of the hierarchical structure of WordNet (or any other taxonomy with a similar structure), the path length between concepts can be used to measure the similarity between concepts. Two different measures of this type were used:

  Path measure [PATH][3] [29]: this uses the length of the path between two concepts to measure the similarity of the concepts. It is important to note that this particular implementation only takes into account "is-a" relations.
  Hirst and St. Onge measure [HSO] [30]: this takes into account many other WordNet relations, beyond the "is-a" relation (antonyms, synonyms…).
- Information content measures
  Information content measures the specificity of a concept, which is higher for more specific concepts. Three different measures of this type were used:

  Resnik measure [RES] [31]: the idea is that two concepts are semantically similar in proportion to the amount of information they share. So the measure is calculated as the information content of their lowest common subsumer in the hierarchy:

  $$sim_{res}(c_1, c_2) = IC(lcs(c_1, c_2))$$

  Jiang and Conrath measure [JCN] [32]: the idea is that if the sum of the individual information contents is similar to that of their lowest common subsumer, then the concepts are close together in the hierarchy. The calculation is as follows:

  $$sim_{jcn}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 * IC(lcs(c_1, c_2))}$$

  Lin measure [LIN] [33]: this measures the ratio of the information content of the lowest common subsumer to the information content of each of the concepts:

  $$sim_{lin}(c_1, c_2) = \frac{2 * IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)}$$

- Gloss-based measures
  This kind of measures makes use of the glosses associated with each concept in WordNet. Two different measures of this type were used:

  Extended Gloss Overlap measure [LESK-E] [34]: this calculates the similarity of two concepts from the overlapping of the glosses associated with each concept and with their related concepts in WordNet.
  Gloss Vectors measure [VECTOR] [35]: this represents each concept as a gloss vector by averaging gloss co-occurrence data and calculates the similarity by finding the cosine between these vectors.

Because each word may have many senses, in the comparison of two words, all the senses associated with each word (taking into account its corresponding part of speech) are used. It is also important to note that the gloss-based measures and HSO are the only measures that can compute the similarity between two words that are different parts of speech. Furthermore, the particular implementation of these measures used by our method makes that gloss-based measures are the only measures capable of computing the similarity between adjectives and adverbs. In WordNet, adjective structure has some kinds of

[3] In brackets the abbreviation by which we will often refer to each measure.

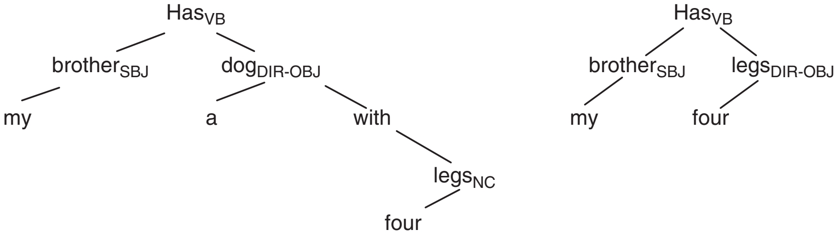

Fig. 1. Parse trees of the sentences: "My brother has a dog with four legs" and "My brother has four legs". Below each head of a phrase, the syntactic function of the phrase is shown.

relations (e.g. "similar-to", "antonyms", "see-also") that could be used by a PATH-like similarity measure. However, the particular characteristics of these relations make gloss-based measures much more adequate to compute semantic similarity between adjectives and adverbs.

3.2. Semantic similarity between sentences

The semantic similarity method proposed in this paper explores the importance of syntactic structure in the calculation of sentence semantic similarity. The similarity between two sentences is calculated as a sum of the similarities between the heads of the phrases that have the same syntactic function in both sentences, following the formula below:

$$sim(s_1, s_2) = \frac{1}{n}\sum_{i=1}^{n} sim(h_{1i}, h_{2i}) - l \cdot PF \quad (1)$$

Let us assume that sentence s_1 is made of n phrases and their heads are $h_{11}, \ldots, h_{1n}$ and sentence s_2 is made of n phrases, and $h_{21}, \ldots, h_{2n}$ are their heads; and moreover the phrases of h_{1i} and h_{2i} have the same syntactic function. Also let us assume that the sentences have l syntactic roles that are only present in one of the sentences. In this case, if one sentence has a phrase not shared by the other, a penalization factor (PF) is introduced to reflect the fact that one of the sentences has extra information. The heads that are not present in WordNet (for example, pronouns) are ignored in the calculation unless the same word shares the same syntactic role in both sentences. This way our method overcomes some of the limitations of WordNet such as the absence of proper nouns.

In the example presented above, "My brother has a dog with four legs" and "My brother has four legs", the method would work as follows: first of all, the method extracts the syntactical structure of the two sentences, obtaining the parse trees shown in Fig. 1. In this method, the syntactic analysis of the sentences is carried out by the system of Johansson and Nugues [36],[4] which yielded the best results in the CoNLL-2008 shared task, concerning syntactic dependency parsing and semantic role labeling. This system carries out the dependency-syntactic and the semantic analysis jointly. The syntactic submodel uses a statistical approach to obtain a list of candidate syntactic structures while the semantic submodel uses a classifier pipeline in order to obtain the list of candidate predicate-argument structures. Finally, the system carries out a reranking on the candidate set of syntactic–semantic structures using information from both syntactic and semantic sources.

Once the syntactic information has been extracted, the method computes the semantic similarity between the words that share the same syntactic functions in both sentences. In this case, the method would calculate the similarity between the principal verbs. "Have" and "have" would get a similarity of 1, because they are the same; the heads of the subjects would again get a similarity of 1; and the heads of the direct objects "dog" and "leg", would get 0.1414 with LIN. Since the second sentence does not have a noun complement of the direct object, the penalization factor is used to take into account this extra information. Then, supposing a penalization factor of 0.03, the semantic similarity obtained for these two sentences according to formula (1) would be:

$$\frac{1}{3}(1 + 1 + 0.1414) - 0.03 = 0.6838$$

Note that these two sentences bear a certain semantic similarity, since both of them refer to my brother having something, and this is exactly the similarity captured by our method. However, word-co-occurrence-based methods or word-order-based methods would give a higher similarity because of the presence of the words "legs" and "four" in the same order in both sentences, although actually these two words do not contribute to increase the semantic similarity.

Note also that, because of our method's use of word semantic similarity measures, our method considers the sentence "My brother has four cats" much more similar to the sentence "My brother has a dog with four legs" than the sentence "My brother

[4] Executables available at: http://nlp.cs.lth.se/lth_srl/.

Table 1
Example sentence pairs from the first dataset.

1. Cord is strong, thick string.
A smile is the expression that you have on your face when you are pleased or amused, or when you are being friendly.
42. A bird is a creature with feathers and wings, females lay eggs and most birds can fly.
A crane is a large machine that moves heavy things by lifting them in the air.
56. The coast is an area of land that is next to the sea.
The shores or shore of a sea, lake or wide river is the land along the edge of it.
62. A cemetery is a place where dead people's bodies or their ashes are buried.
A graveyard is an area of land, sometimes near a church, where dead people are buried.

has four legs". This is due to the fact that "cat" is more similar to "dog" (0.8968 with LIN) than "leg" (0.1414 with LIN). And this seems to be a logical inference since now both sentences refer to my brother having an animal (or animals) and not just having something. This higher similarity is achieved with fewer words in common, contrary to the result that would be given by word co-occurrence methods, which would find a lower similarity between the two sentences of this second example.

It is worth saying that our method's deep parsing is the key to detecting the possible orderings of phrases in the same sentence. Our method thus outperforms some other methods that make use only of primary syntactic information, such as word order. Taking into account only the word order information, the sentences "I will go to your house tomorrow" and "I will go tomorrow to your house" could be different, when they are not, while the complete syntactic information used by our method would allow it to detect this real similarity.

Talking about complex sentences with multiple clauses, our method works as follows: the syntactic functions of coordinate clauses are considered equivalent. For example, the sentences "I took a book and a pencil" and "I took a pen and a book" are considered to have two direct objects. In these cases, all the possible comparisons are done and the highest similarity score is taken for each of them. In our example, the term "book" of the first sentence is compared with the terms "pen" and "book" of the second sentence, obtaining similarity scores of 0.83 and 1 respectively (using, for example, LIN measure). This way, a similarity score of 1 is selected given that "book–book" is the most similar pair. In the same way, the term "pencil" of the first sentence is compared to the terms "pen" and "book" of the second sentence, obtaining similarity scores of 0.9 and 0.76 respectively. This way, a similarity score of 0.9 is selected given that "pen–pencil" is the most similar pair. Regarding subordinate clauses, SyMSS only uses one level of subordination (i.e., a subordinate clause embedded on another subordinate clause is not taken into account). This means that if the two sentences to compare have a subordinate clause with the same syntactic function, the terms of these subordinate clauses are compared attending to its syntactic functions in the subordinate clause. For example, in the sentences: "This is the book I read last month" and "This is the computer he gave me", SyMSS compares the terms "is" and "is" and "book" and "computer" since they are the verbs and subjects of the two main clauses and it also compares "read" and "gave" since they are the verbs of the subordinate clauses that have the same function in both sentences.

4. Experimental results

4.1. Data sets

Two data sets were used to evaluate the performance of the SyMSS method. As stated before, there is not much work about semantic similarity at the sentence level, so there are no benchmark data sets for evaluating methods of this kind. We therefore used the data set proposed by Li et al. [12] to enable comparison with the Li–McLean method and with the similarity scores given by human evaluators. Also, in order to evaluate our sentence similarity method with a larger dataset and in a much more challenging task, we used the Microsoft Paraphrase Corpus [37].[5]

The first data set contained 65 pairs of sentences created from the 65 noun pairs from [38], replaced by the nouns' definitions from the Collins Cobuild dictionary (see [1,11] for a full understanding of how the data set was collected and what method was employed in using the data set to compare sentence semantic similarity measures). In order to enable comparisons with the Li–McLean and the Islam–Inkpen methods, the same 30 pairs of sentences used by Li et al. [12] were selected for the evaluation (you can see some example sentence pairs extracted from this dataset on Table 1. For a full listing of the sentence pairs you can see www2.docm.mmu.ac.uk/STAFF/J.Oshea/TRMMUCCA20081_5.pdf). This data set contained the average similarity scores given by 32 human judges collected by Li et al. [12].

The Microsoft Paraphrase Corpus consists of 4076 training and 1725 test sentence pairs collected from thousands of news sources on the web over a period of 18 months, and labeled by two human annotators who determined whether the two sentences in a pair were semantically equivalent paraphrases or not. The agreement between human evaluators was approximately 83%, which can be considered as an upper bound for the automated task.

[5] Available from http://research.microsoft.com/nlp/msr_paraphrase.htm.

Table 2
Pearson's and Spearman's coefficients between human scores and the similarities given by the seven variations of our method and its corresponding baselines. Also the average difference in ranks and its standard deviations are given.

	Pearson's correlation	Spearman's correlation	Diff.	
			Avg.	Std. dev.
PATH–SyMSS	**0.76**	**0.71**	5.3	4.00
HSO–SyMSS	0.70	0.68	5.6	4.42
JCN–SyMSS	0.70	0.67	5.73	4.62
RES–SyMSS	0.73	0.70	4.73	4.81
LIN–SyMSS	0.69	0.69	4.93	4.85
LESK-E–SyMSS	0.40	0.37	8.33	5.23
VECTOR–SyMSS	0.51	0.46	7.13	5.66
PATH–baseline	0.52	0.49	5.07	4.57
HSO–baseline	0.49	0.48	5.47	4.29
JCN–baseline	0.51	0.46	5.40	4.83
RES–baseline	0.50	0.50	5.33	4.21
LIN–baseline	0.43	0.51	5.20	4.20
LESK-E–baseline	0.38	0.32	7.07	5.78
VECTOR–baseline	0.46	0.40	5.80	4.57

Bold entries show the best performing variation in Pearson's and Spearman's correlation.

4.2. Evaluation measures

Three different measures were used to evaluate the performance of the seven variations of SyMSS (see Table 2 for an enumeration of the seven variants or Section 3.1 for an explanation of the concept similarity measures used by each variant):

1) Pearson's correlation coefficient for the normalized similarities yielded by each method and the normalized similarities given by humans.
2) Spearman's rank correlation coefficient was also used to evaluate the different proposed variations. This coefficient measures the correlation between the ranks of two variables. Note that the correlation between the similarity values may be very different while the ordering of the sentence pairs is the same. Thus, the ranks yielded by each method are another point of view to be taken into account when comparing semantic similarity methods with human intuition. In fact, this qualitative measure could prove more important than correlation in terms of scores. Note that for many applications such as information retrieval, the rank of the different options is much more important than their absolute similarity value.
3) The average difference between ranks was also used in the evaluation. For each sentence pair, the difference of its position in the ranking given by each method and by human evaluators was computed, and then the average difference and the standard deviation were found for each method.

4.3. Evaluation of semantic measures between concepts

4.3.1. Experiments and results

The first thing we wanted to do was compare the performance of each of the seven measures of word semantic similarity studied in this article.[6] In order to do so, we used each of the measures with the SyMSS algorithm and computed the similarities of the 30 pairs of sentences in the second data set. Given these similarities, we found the rank of each pair of sentences and computed the Pearson's and Spearman's correlation coefficients of the similarities given by each measure and the similarities given by human evaluators. Also, the average and standard deviations of the differences of ranks given by each measure were computed to show how different the ranks given by humans and by our method were.

In order to show the contribution of syntactic information in the calculation of semantic similarity, we computed baselines that do not exploit syntactic information. The baseline method just compares each word of the first sentence with all the words of the second sentence and averages the maximum similarity score of each word. We used seven different variations of the baseline method using each of the seven concept similarity measures used by SyMSS.

Table 2 shows the Pearson's and Spearman's correlation coefficients between the similarities of the semantic similarity method proposed in this paper and those of each of the seven measures compared and human scores, and also the average and standard deviation of the difference of ranks given by each method and the human evaluators. The value of the penalization factor used in this and the following experiments was set empirically at 0.03. In order to set the penalization factor we used as training data the 35 out of 65 sentences of the second data set that are not used to evaluate the method. We obtained the correlation between the scores given by our method and the human scores using 10 different penalization values ranging from 0 to 0.1 with increments of 0.01. Remember that the PF is introduced to reflect the fact that one of the sentences has extra information. So, if the sentences have *l* syntactic roles that are only present in one of the sentences, PF is subtracted *l* times from the final similarity score.

[6] The implementation of these metrics was taken from JWordNet-Similarity (http://www.h-its.org/english/research/nlp/download/jwordnetsimilarity.php), a Java interface to access WordNet.

Table 3
Similarities and ranks given by humans and SyMSS with PATH.

Pair	R and G word pair	Human evaluators			SyMSTeSS (path version)			
		Avg.	Avg. norm.	Rank	Sim.	Sim. norm.	Rank	Diff.
1	1.cord:smile	0.01	0.01	27	0.32	0.32	23	4
2	5.autograph:shore	0.01	0.01	28	0.28	0.28	26	2
3	9.asylum:fruit	0.01	0.01	29	0.27	0.27	27	2
4	13.boy:rooster	0.11	0.11	24	0.27	0.27	28	4
5	17.coast:forest	0.13	0.14	22	0.42	0.42	13	9
6	21.boy:sage	0.04	0.04	26	0.37	0.37	18	8
7	25.forest:graveyard	0.07	0.07	25	0.53	0.53	8	17
8	29.bird:woodland	0.01	0.01	30	0.31	0.31	24	6
9	33.hill:woodland	0.15	0.16	20	0.43	0.43	12	8
10	37.magician:oracle	0.13	0.14	23	0.23	0.23	30	7
11	41.oracle:sage	0.28	0.29	19	0.38	0.38	17	2
12	47.furnace:stove	0.35	0.36	17	0.24	0.24	29	12
13	48.magician:wizard	0.36	0.38	15	0.42	0.42	14	1
14	49.hill:mound	0.29	0.30	18	0.39	0.39	15	3
15	50.cord:string	0.47	0.49	13	0.35	0.35	20	7
16	51.glass:tumbler	0.14	0.15	21	0.31	0.31	25	4
17	52.grin:smile	0.49	0.51	11	0.54	0.54	7	4
18	53.serf:slave	0.48	0.50	12	0.52	0.52	9	3
19	54.journey:voyage	0.36	0.38	16	0.33	0.33	22	6
20	55.autograph:signature	0.41	0.43	14	0.33	0.33	21	7
21	56.coast:shore	0.59	0.61	6	0.43	0.43	11	5
22	57.forest:woodland	0.63	0.66	5	0.50	0.50	10	5
23	58.implement:tool	0.59	0.61	7	0.64	0.64	5	2
24	59.cock:rooster	0.86	0.90	2	1.00	1.00	1	1
25	60.boy:lad	0.58	0.60	8	0.63	0.63	6	2
26	61.cushion:pillow	0.52	0.54	10	0.39	0.39	16	6
27	62.cemetery: graveyard	0.77	0.80	3	0.75	0.75	4	1
28	63.automobile:car	0.56	0.58	9	0.78	0.78	3	6
29	64.midday:noon	0.96	1.00	1	1.00	1.00	2	1
30	65.gem: jewel	0.65	0.68	4	0.36	0.36	19	15
		Avg.	0.38 ± *0.28*		Avg.	0.46 ± *0.20*		
					Pearson's r	0.76		
					Spearman's ρ	0.71		

The results showed that the best performance measure was PATH. Table 3 gives the detailed results obtained by the method proposed here using PATH with the 30 pairs of selected sentences. The similarities, normalized similarities, ranks and difference of ranking with human evaluators, are included.

4.3.2. Discussion

The very first conclusion that can be extracted from the first experiment is the importance of taking into account deep syntactic information in the process of semantic similarity computation. Five of the seven variations of the method outperform clearly its corresponding baselines in terms of Pearson's and Spearman's correlation. And it is important to note that, as we will see below, LESK and VECTOR variants have some peculiarities that make them not suitable for this task. So the fact that these two variations only slightly outperform its corresponding baseline does not weaken the conclusion.

Analyzing the differences among the different variations of SyMSS studied, we can see that PATH yields the best results in terms of correlation with human scores and ranks. This could seem surprising, given that path lengths are more appropriate when they are consistent throughout the taxonomy used. Nevertheless, in WordNet concepts at a higher level in the hierarchy are more general, so a path of distance equal to one suggests a larger difference than a path of the same distance placed at a lower level in the hierarchy. For example, "mouse" and "rodent" are separated by a path of length one, and "fire iron" and "implement" are also separated by a path of length one. However, due to the characteristics of this particular corpus, this disadvantage is not very influential because most compared words in the assessed data set of this experiment are at similar levels of the hierarchy.

We can also see that other path-based measures and information content measures also yielded good results (all showing around 0.7 in Pearson's and Spearman's correlations with human similarity scores). Gloss-based measures, however, obtained poorer results. This might seem unexpected since gloss-based measures are the best-performing measures when applied to other problems, such as word sense disambiguation (see Pedersen et al., [28]). However, the characteristics of the problem addressed in this paper make measures of this kind unsuitable. This could be due to factors like the following:

1) Gloss-based measures make use of the overlap of the glosses associated with each concept, so the similarity score computed for two words that are exactly the same varies depending on the word. Recall that in the comparison of two words all the senses associated to each word (taking into account its corresponding part of speech) are used. For example, LESK-E gives a similarity of 703 to the pair of words "chicken" and "chicken" but gives a similarity of 2529 to the pair of words "dog" and "dog", so the

pair of sentences "It is a dog"–"It is a dog" is much more similar than the pair "It is a chicken"–"It is a chicken", even though the two pairs of sentences should have exactly the same similarity. This fact is also present in JCN and RES, but it has less harmful effects there because the differences given by RES are quite low and, the differences in JCN are only between different parts-of-speech (JCN gives a similarity of $2.72 \cdot 10^{-7}$ to a pair of exactly the same nouns and a similarity of $1.22 \cdot 10^{-7}$ to a pair of exactly the same verbs).

2) Another factor that reduces the performance of gloss-based measures is synonym processing. When other types of measures have to rate the similarity between two synonyms (i.e., two words pertaining to the same synset in WordNet), they find the same similarity as if the two words were exactly the same (and this is correct because two words of a same synset are semantically identical). But words of the same synset have different associated glosses, so gloss-based measures yield a different semantic similarity for words in the same synset.
3) Lastly, gloss-based measures allow words pertaining to different parts of speech to be compared. This might seem advantageous but we have found that similar syntactic functions are usually performed by words acting as the same part of speech. So, comparisons of two words belonging to different parts of speech are often due to errors in the syntactic analysis. Other types of measures filter out these errors, thus yielding better results.

Considering the three above characteristics of gloss-based measures, we tried to improve on the performance of gloss-based measures simply by avoiding comparisons of words acting as different parts-of-speech and normalizing the similarity of two words pertaining to the same synset to 1. As expected, these solutions worked well with VECTOR, which improved its performance from the initial Pearson's and Spearman's correlations with human similarities of 0.51 and 0.46 to correlation coefficients of 0.74 and 0.71, respectively. LESK-E, however, did not improve much due to the normalization used. Since LESK-E does not score similarity on a scale between 0 and 1 (note that VECTOR does score on a scale between 0 and 1), giving a similarity score of 1 to two words pertaining to the same synset is pointless. In fact, normalizing LESK-E using its maximum value produces no major improvement, so LESK-E does not seem to offer a good solution to the problem of computing sentence semantic similarity.

Besides the comparison of these seven measures of concept semantic similarity, Table 2 shows promising results in terms of human correlation. SyMSS using PATH, the best-performing variation, yielded Pearson's and Spearman's correlations of 0.761 and 0.705, respectively. Also, when using other path-based or information content measures, SyMSS yielded similar results, always around 0.7 for Pearson's and Spearman's correlation with human intuition in terms of normalized similarities. These results support the observation that syntax is a very important source of information for computing semantic similarity between sentences. Given the great importance of deep syntactic information in the computation of semantic similarity by our method, improving syntactic analysis would seem likely to lead to an improvement of our method's overall results.

Finally, it is important to note that the characteristics of this data set could have influenced the results obtained by the methods and also the similarity values given by the human evaluators. Because each sentence was a definition, the presence of the defined term in the sentence could have been a bias point in the evaluation process.

4.3.3. Error analysis

In order to understand better how the semantic similarity measure works and which are its strengths and weaknesses, we have conducted an error analysis. In the previous point we analyzed the problems of the gloss-based variants. Here we will focus on the rest of the variations. Given that all of them have similar errors, we will deeply analyze the errors of the best performing variation, PATH–SyMSS. The analysis will be focused both on terms of semantic similarity scores and rankings. Tables 2 and 3 show the difference between the similarity scores and rankings given by PATH–SyMSS and human evaluators.

There are two main kinds of errors: overestimation and underestimation errors. The first kind is mostly present with low similarity sentence pairs. This fact is mainly due to the particular characteristics of this data set. As stated before, the data set is made of definitions. Thus, the main verb of all the sentences is the verb 'to be'. This fact implies that all the sentence pairs have at least a common pair of phrase's heads. This way, a pair of sentences with two totally dissimilar phrases according to formula (1) would result in a score of $(0+0+1)/3=0.33$, which is surely much higher than expected. For example, SyMSS ranks sentence pairs 6 and 7 (see Table 4) 8 and 17 positions above its corresponding human ranking. This problem would be reduced with a

Table 4
Example sentence pairs that present over and underestimation problems.

Overstim.	6.boy:sage A boy is a child who will grow up to be a man. A sage is a person who is regarded as being very wise. 7.forest:graveyard A forest is a large area where trees grow close together. A graveyard is an area of land, sometimes near a church, where dead people are buried.
Understim.	12.furnace:stove A furnace is a container or enclosed space in which a very hot fire is made, for example to melt metal, burn rubbish or produce steam. A stove is a piece of equipment which provides heat, either for cooking or for heating a room. 30.gem: jewel A gem is a jewel or stone that is used in jewellery. A jewel is a precious stone used to decorate valuable things that you wear, such as rings or necklaces.

wider data set made up not only by definitions. Underestimation errors are mainly due to complex structures present in the sentences of this dataset. Complex sentences usually have many different phrases. Thus, the denominator *n* in formula (1) is very high, leading to the underestimation. Besides when the difference in length between the two compared sentences is very high, the penalization factor is also very high, leading to a very low similarity. Two examples of this kind of errors are sentence pairs 12 and 30 (see Table 4) which are ranked 12 and 15 positions below its corresponding human ranking.

There are some possible solutions to these errors that will be tested as future work. Overestimation errors could be overcome by weighting the similarity of each pair of heads with the inverse of its generality. This way, a very general verb such as "to be" would have less importance than other more specific terms. The generality of each term could be measured, for example, attending to its level in the WordNet hierarchy. Also, as stated before, it would be very useful to have datasets made up not only by definitions.

In order to deal with underestimation errors, it would be necessary to smooth both the number of phrases *n* and the penalization factor *PF* used by the semantic similarity method. When the compared sentences present complex structures, the denominator in formula (1) is usually very high. Thus, using *log(n)* instead of *n* could be helpful. The penalization factor is also high when the compared sentences have very different lengths. Thus, it could be useful to calculate it as a function not only of the number of different phrases but also of the difference of length between the two sentences.

4.4. Influence of adjectives and adverbs

4.4.1. Experiments and results

Another issue studied in this research was the influence of adjectives and adverbs on the computation of sentence semantic similarity, an issue that had not been studied previously. Gloss-based measures are the only methods suitable to deal with adjectives and adverbs but, as stated before, there are many reasons why gloss-based measures should not be used, so other experiments should be performed to study the significance of adjectives and adverbs. The good results of path-based and information content measures show that the main role in terms of semantic similarity belongs to nouns and verbs. However, adjectives and adverbs do give sentences some shade of meaning that should be taken into account in the computation of semantic similarity. See for example, that using only path-based or information content measures, the similarity score for the pair of sentences "I have a big dog"–"I have a big dog" was 1, and the score obtained for the pair of sentences "I have a big dog"–"I have a little dog" was also 1, while a human evaluator would have given a lower similarity score to the second pair of sentences. Thus, it was posited that taking into account adjectives and adverbs should improve SyMSS's previous results. To prove this, a combination of gloss-based and non-gloss-based measures was used. In the experiment discussed below, we used the five path-based and information content variations of SyMSS together with VECTOR. The only modification in each variation was that, when the method had to measure the semantic similarity between adjectives or adverbs, the similarity given by VECTOR was used. It is important to note that this similarity was previously normalized depending on the variation of SyMSS used. For example, HSO gave a score from 0 to 16, so the similarity found by VECTOR for a pair of adjectives was multiplied by 16 (note that VECTOR gives similarity scores from 0 to 1).

The results obtained by each of the five variations of SyMSS are shown in Table 5. To show how the results were improved by taking into account adjectives and adverbs, Table 5 shows the correlations with human scores (Pearson's correlation) and human ranks (Spearman's correlation) obtained by each variation before and after the use of adjectives and adverbs in semantic similarity computation.

4.4.2. Discussion

The results shown in Table 5 confirm the intuitive idea that adjectives and adverbs play a significant role in sentence semantics and so ought to be used in computing sentence semantic similarity. The five variations we studied yielded better results in terms of rank correlation with human evaluators, with an improvement of 8.68 points in the best case (HSO) and an average improvement of 6.59 points.

The Pearson's correlation between the scores given by each of the variations and the scores given by humans shows that the path-based measures showed improved performance, while the information content measures yielded slightly worse results.

Table 5
Pearson's and Spearman's correlation with and without taking into account adjectives and adverbs.

	Pearson's correlation		Spearman's correlation	
	Without adj. and adv.	With adj. and adv.	Without adj. and adv.	With adj. and adv.
PATH–SyMSS	0.76	**0.79**	0.71	**0.78**
HSO–SyMSS	0.70	**0.72**	0.68	**0.76**
JCN–SyMSS	**0.70**	0.67	0.67	**0.67**
RES–SyMSS	**0.73**	0.69	0.70	**0.78**
LIN–SyMSS	**0.69**	0.64	0.69	**0.76**

Bold entries show, for each variation, the best performing option: with or without adjectives and adverbs.

Table 6
Pearson's and Spearman's correlation of the improved SyMSS method and the Islam–Inkpen method.

Measure	Pearson's correlation	Spearman's correlation
Liu–SyMSS	0.79	0.85
Pirró–Seco–SyMSS	0.79	0.84
Path–SyMSS	0.79	0.78
Li–McLean	0.81	0.81
Islam–Inkpen	0.85	0.83

This was due to a problem of differences in the scale used by the three different types of measures. In our experiment, VECTOR usually scored the similarity of words between 0.45 and 0.65 (we obtained a mean of 0.56 with a standard deviation of 0.10), which was very similar to the scoring scale used in path-based measures. HSO gave a mean score of 0.56 and a standard deviation of 0.18 and PATH yielded a mean of 0.60 with a standard deviation of 0.14. However, the information-content-based measures yielded very different results in similarity scores. The averages and standard deviation values for JCN, RES and LIN were: 0.48 ± 0.22, 0.66 ± 0.22 and 0.67 ± 0.24, respectively. Because the similarity between adjectives and adverbs was calculated with VECTOR, and its typical scores are different from the typical scores given by information-content-based measures, some degree of correlation with human evaluations was lost. However, the improvements made in terms of rank correlation showed that the nuances introduced by adjectives and adverbs are of great importance in sentence semantic similarity.

4.5. Improving word semantic similarity

The proposed system focuses on two fundamental points: the syntactic analysis and the concept semantic similarity measure used. So, improving these points would lead to an overall improvement of system performance. The concept semantic similarity measures used in the experiments discussed above are some of the most regularly used in the literature. However, there are many other measures that outperform them in terms of correlation with human ratings. We selected the measures proposed by Liu et al. [39] and Pirró and Seco [40] (which will be referred to as the Liu metric and the Pirró–Seco metric respectively), to evaluate the improvements brought about by a better concept semantic similarity measure. The Li–McLean corpus was used to evaluate the performance of SyMSS with these two measures, and the results shown in Table 6 were obtained. In this table, we also show the results obtained by the Islam–Inkpen measure and the best of the approaches of SyMSS used thus far, the path measure approach.

The results shown in Table 6 confirmed the hypothesis that an improvement in the calculation of semantic similarity between words would lead to an overall improvement of the system. The SyMSS variation that used the Pirró–Seco metric outperformed all of the variations in terms of scores and rank correlation with human ratings. Moreover, both the Liu and the Pirró–Seco approaches outperformed the measures proposed by Li et al. [12] and Islam and Inkpen [13] in terms of Spearman's correlation. As stated in Subsection 4.2, Spearman's correlation is a more useful measure of evaluation than score correlation. The absolute value given by score correlation is in most cases not as useful as the relative information given by rank correlation.

4.6. Paraphrasing identification

4.6.1. Experiments and results

In order to evaluate our sentence similarity measure with a larger dataset and in a much more challenging task, we used the Microsoft Paraphrase Corpus [37]. This corpus consists of 4076 training pairs and 1725 test pairs collected from thousands of news sources on the web over a period of 18 months, which have been labeled by two human annotators who determined whether the two sentences in a pair were semantically equivalent paraphrases or not. The agreement between human evaluators was approximately 83%, which can be considered as an upper bound for the automated task. For this paraphrase identification task, we

Table 7
Results with the Microsoft Paraphrase Corpus. SyMSS variations, similar methods and baselines.

Measure	Best similarity threshold	Prec.	Rec.	Rej.	F1	f1	Acc.
PATH	0.35	71.94	89.82	21.62	79.89	34.85	69.16
JCN	0.4	74.04	88.63	21.47	80.68	34.57	70.42
RES	0.45	73.23	89.82	26.03	80.68	40.36	69.48
LIN	0.35	73.18	89.43	27.18	80.49	41.69	70.10
HSO	0.55	70.47	81.71	**45.45**	75.67	**58.41**	68.43
VECTOR	0.5	73.65	90.37	26.11	81.16	40.51	70.52
LIU	0.4	73.20	91.58	23.57	80.74	37.49	70.11
Islam and Inkpen	0.6	**74.65**	89.13	39.97	81.25	55.19	**72.64**
Mihalcea et al.	0.5	69.6	**97.7**	–	**81.3**	–	70.3
Random	–	68.3	50.0	–	57.8	–	51.3
Vector-based	0.5	71.6	79.5	–	75.3	–	65.4

Bold entries show the best performing method for each evaluation measure.

used SyMSS as a supervised method, using the training set to earn the best similarity threshold score and the test set to check the method against this similarity threshold. To determine whether a pair is a paraphrase or not we used different similarity thresholds ranging from 0 to 1 with an interval of 0.05. For each candidate paraphrase pair in the training set, the system found the semantic similarity score and then labeled the candidate pair as a paraphrase if the similarity score exceeded each of the thresholds used. After the evaluation with the training test, we selected the best similarity threshold in terms of accuracy for each of the variations of SyMSS evaluated. These similarity thresholds were then used in the evaluation process with the test set. The results obtained with this set for each of the variations tested are shown in Table 7.

According to Mihalcea et al. [18], two baselines were used: Random simply makes a random decision for each candidate pair and Vector-based uses a cosine similarity measure as traditionally used in information retrieval, with tf-idf weighting.

The evaluation metrics used to measure the performance of the different variations of SyMSS were the ones proposed by Achananuparp et al. [41] Precision is the proportion of correctly predicted paraphrase sentences to all predicted paraphrase sentences. Recall is the proportion of correctly predicted paraphrase sentences to all paraphrase sentences. Rejection is the proportion of correctly predicted non-paraphrase sentences to all non-paraphrase sentences. Accuracy is the proportion of all correctly predicted sentences compared to all sentences. F1 and f1 are the uniform harmonic mean of precision–recall and rejection–recall respectively. Table 7 shows the results obtained by each of the different variations of SyMSS for this task. Baseline results and the results obtained by Mihalcea et al. [18] and Islam and Inkpen [13] are also shown, for the sake of comparison. It is important to note that the similarity thresholds shown for these methods were obtained by the authors using the training set in the same way as we did.

4.6.2. Discussion

SyMSS applied to the Microsoft Paraphrase Corpus showed good, promising results. All the variations obtained better results than the baselines. The improvement of the semantic similarity measures over the two baselines was found to be statistically significant ($p<0.001$) in all the experiments, using a parametric paired t-test once we confirmed that data were normally distributed by a Chi-Square test for the goodness of fit ($p<0.05$). Furthermore, all approaches have a similar performance to the one proposed by Mihalcea et al. [18] in terms of accuracy. In fact, it is worth noting that the results included in Table 7 for Mihalcea et al. [18] are the best scores recorded for that method which were achieved by combining different WordNet-based similarity measures. A comparison between the different approaches tested by Mihalcea et al. [18] using a single similarity measure and the different variants of SyMSS method was done. It was found that all the SyMSS variants were significantly better (two-tailed paired t-test, $p<0.01$) than the corresponding variants proposed by Mihalcea et al. [18]

In the light of evaluation, VECTOR and JCN can be seen to be the better variations of SyMSS regarding accuracy. In spite of the poor results that these similarity measures yielded when evaluated on the Li–McLean corpus, the results obtained with the Paraphrase Corpus are consistent with the ones reported by Mihalcea et al. [18] and Fernando and Stevenson [42]. Also, the superiority of JCN is consistent with other evaluations of WordNet similarity measures [43]. We think that the differences between these results and the results reported with the Li–McLean corpus are due to many reasons, for example, the higher complexity of the sentences included in the Microsoft Paraphrase Corpus or the fact that the tasks concerned are very different. As acknowledged by Corley and Mihalcea [44] and Islam and Inkpen [13], sentence semantic similarity measures are a necessary step in the paraphrase recognition task, but are not always sufficient. For example, the Microsoft Paraphrase Corpus contains the following sentence pair:

> “I notice a mood change in their priorities”, one politician said.
> “I notice a mood change in their priorities”, said one Iraqui politician after meeting with Mr. Bremer.

These two sentences are highly related, but they are not considered a paraphrase. Many other examples could be found to demonstrate the different nature of the two tasks. Thus, the evaluation of semantic similarity measures with human similarities and ranks is much more suitable than the evaluation that takes paraphrasing detection into account. However, given the lack of big human-rated sentence semantic similarity corpus, the Microsoft Paraphrase Corpus stands as a very useful tool for comparing different measures.

5. W-SyMSS: weighted SyMSS

Wiemer-Hastings [11] points that human judges tend to ignore similarities between segments that are playing different functional roles, whose tendency denotes the importance of deep syntactic structure analysis in the computation of semantic

Table 8
Weights assigned to the different syntactic roles.

Syntactic role	Weight
Verb	0.433
Subject	0.347
Object	0.347
Adverbial complement	0.250

Table 9
Results with the Microsoft Paraphrase Corpus. W-SyMSS variants, similar methods and baselines.

Measure	Best similarity threshold	Prec.	Rec.	Rej.	F1	f1	Acc.
PATH	0.4	70.75	91.74	28.32	79.89	43.27	69.81
JCN	0.45	74.47	84.17	41.61	79.02	55.68	70.87
RES	0.35	71.56	90.69	24.46	80.00	38.53	69.32
LIN	0.35	74.34	87.46	34.08	80.37	48.98	70.63
HSO	0.5	71.16	77.94	**43.61**	74.40	**55.92**	68.72
VECTOR	0.45	74.15	90.32	36.53	81.44	52.02	70.82
LIU	0.45	72.87	93.49	22.04	**81.91**	35.67	70.58
Islam and Inkpen	0.6	**74.65**	89.13	39.97	81.25	55.19	**72.64**
Mihalcea et al.	0.5	69.6	**97.7**	–	81.3	–	70.3
Random	–	68.3	50.0	–	57.8	–	51.3
Vector-based	0.5	71.6	79.5	–	75.3	–	65.4

Bold entries show the best performing method for each evaluation measure.

similarity, and he claims that different syntactic roles are of different importance in humans' calculation of semantic similarity. In view of Wiemer-Hastings' results, we accept the hypothesis that different syntactic roles are of different importance in humans' calculation of semantic similarity, and we take the additional step forward of trying to add psychological plausibility to our method by weighting the different syntactic roles the way humans do.

The results reported by Wiemer-Hastings [11] show that, when computing semantic similarity between sentences, humans tend to give the highest importance to verbs. The subject and direct object are of great importance also, but there is no significant difference between them. Based on these observations and the weights obtained empirically by Wiemer-Hastings, we assigned the following weights to each syntactic role: Table 8.

Because there are no human data about other syntactic roles, the weight for the rest of the possible functions present in a sentence was set at $w_R = 0.15$ on the assumption that other syntactic roles are of lesser importance. This assumption was partially proved in the work of Wiemer-Hastings, who showed that humans give a lower importance to indirect objects. With all this information, W-SyMSS computed sentence similarity as follows:

$$sim(s_1, s_2) = \frac{w_S \cdot s_S + w_V \cdot s_V + w_O \cdot s_O + w_A \cdot s_A + \sum_{i=1}^{n} w_R \cdot sim(h_{1i}, h_{2i})}{w_S + w_V + w_O + w_A + w_R \cdot n} - l \cdot PF$$

Let us assume that sentence s_1 and s_2 are made of a subject, a verb, an object, and an adverbial complement whose semantic similarities are s_S, s_V, s_O and s_A respectively. Also each sentence has n other syntactic roles whose heads are $h_{11}, \ldots, h_{1n}$ and $h_{21}, \ldots, h_{2n}$ respectively for sentence s_1 and s_2. Moreover, let phrases of h_{1i} and h_{2i} have the same syntactic function. Also let us assume that the sentences have l syntactic roles that are present in only one of the sentences. Thus, the semantic similarity between both sentences is calculated as shown by the formula above.

We evaluated this new approach of the proposed system, with the Microsoft Paraphrase Corpus, obtaining the results shown in Table 9. Baseline results and the results found by Mihalcea et al. [18] and Islam and Inkpen [13] are also shown, for the sake of comparison. Moreover, Table 10 shows the F-values of the differences between each of the measures studied in order to figure out which results differ significantly from each other.

The results showed that adding psychological plausibility to the method by weighting the different syntactic roles improved the method's overall accuracy. All of the variations of W-SyMSS, except the one using RES were improvements on the corresponding variations of SyMSS. However, only three of these improvements (PATH, LIN and LIU) are statistically significant at the 0.05 level, so it would be necessary to optimize the weights used by the method. Once again, all of the variants have a

Table 10
F-values for the differences between the seven variants of W-SyMSS and the methods of Islam and Inkpen, Mihalcea et al. and the proposed baselines. $^{*}p<0.05$, $^{**}p<0.01$.

	PATH	JCN	RES	LIN	HSO	VECTOR	LIU
JCN	-3.88^{**}						
RES	1.77	5.66^{**}					
LIN	-2.99^{*}	0.88	-4.77^{**}				
HSO	3.95^{**}	7.82^{**}	2.16^{*}	6.94^{**}			
VECTOR	-3.69^{**}	0.18	-5.47^{**}	−0.70	-7.64^{**}		
LIU	-2.81^{**}	1.07	-4.59^{**}	0.18	-6.78^{**}	0.88	
I and I	-7.67^{**}	-4.57^{**}	-8.62^{**}	-4.98^{**}	-8.34^{**}	-4.75^{**}	-5.13^{**}
Mihal.	−1.79	2.09^{*}	-3.57^{**}	1.21	-4.73^{**}	1.91	1.03
Random	34.42^{**}	38.43^{**}	32.59^{**}	37.52^{**}	30.35^{**}	38.25^{**}	37.33^{**}
Vec-bas	10.46^{**}	14.64^{**}	9.67^{**}	13.75^{**}	8.80^{**}	14.45^{**}	13.58^{**}

similar performance to the one proposed by Mihalcea et al. [18] in terms of accuracy. Moreover, the improvement achieved by JCN is statistically significant at the 0.05 level. These results supported the hypothesis first voiced by Wiemer-Hastings and repeated in this paper, stating that human judgments are affected to different extents by different syntactic roles.

6. Conclusions and future work

This paper has presented SyMSS, a syntax-based method to measure the semantic similarity between sentences or very short texts. The key notion on which the method is based is that the meaning of a sentence is made up of not only the meanings of its individual words, but also the structural way the words are combined. SyMSS takes into account semantic and syntactic information to compute sentence semantic similarity. Semantic information is obtained from a lexical knowledge base such as WordNet, which models common human knowledge about words in natural language and allows different types of measures of semantic similarity between concepts to be calculated. Syntactic information is obtained through a deep parsing process that finds the phrases that make up the sentence and their syntactic functions. With this information, the proposed method measures the semantic similarity between concepts that share the same syntactic function.

Our proposed method outperforms the results by Li et al. [12] and Islam and Inkpen [13] for a data set of 30 sentence pairs in terms of Spearman's correlation. For the paraphrase recognition task, our proposed method outperforms the combined unsupervised method of Mihalcea et al. [18].

The influence of adjectives and adverbs in the calculation of semantic similarity has also been studied in this paper. Nouns and verbs play the main role in terms of semantic similarity, but the results obtained for the data set of 30 sentence pairs showed that adjectives and adverbs give the sentence some shade of meaning that should be taken into account in the computation of semantic similarity. For this data set, all five variants of SyMSS showed improved performance in terms of rank correlation when adjectives and adverbs were taken into account.

The proposed method has two main critical points: the syntactic analysis and the word semantic similarity measure. So, improving these points would lead to an overall improvement of the whole system. To evaluate the improvements produced by a better lexical semantic similarity measure, we selected the measures proposed by Liu et al. [39] and Pirró and Seco [40]. The SyMSS variations using these two measures outperformed all the previous variations tested. Moreover, both variations outperformed the methods proposed by Li et al. [12] and Islam and Inkpen [13] in terms of rank correlation. This is an important improvement because for many applications, such as information retrieval, the different options' rank is much more important than their similarity value.

We also added psychological plausibility to the proposed method by weighting the different syntactic roles the way humans do. The different weights were drawn from the findings of Wiemer-Hastings [11]. The results showed that the addition of psychological plausibility to our method led to a statistically significant improvement of the overall accuracy in the paraphrase detection task for three of the studied variants. These results show that the weighting strategy is a promising one, but it would be necessary to optimize the weights used by the method.

Future work will include testing the method with different syntactic parsers. As stated before, syntactic analysis is one of the critical points of the proposed method, so improvements at this point would lead to overall improvements. Moreover, it would be interesting to test our method with different syntactic information such as semantic roles or using some other techniques such as frame semantics. Another line of future work will be the improvement of the semantic similarity measure formula to overcome underestimation errors by taking into account the generality when both sentences present identical terms and overestimation errors by analyzing the contribution of the number of different phrases to the sentence similarity.

Also, comparisons between methods are currently difficult to set up because of the lack of a suitable benchmark data set, so another important point to work on is the construction of a broader corpus of sentence pairs with different kind of sentences and not only definitions, since the presence of the defined term can bias the measuring process. Moreover, due to the limitations of the Wiemer-Hastings [11] experiment, it would be of interest to conduct more experiments to ascertain what weights humans assign to different syntactic roles. Other possibilities of future work are related with the application of the proposed method to different NLP tasks, such as text summarization and conversational agents, to check the method's performance in different application domains.

References

[1] J. O'Shea, Z. Bandar, K. Crocket, D. McLean, A comparative study of two short text semantic similarity measures, proceedings of the agent and multi-agent systems: technologies and applications, Second KES International Symposium, 2008, pp. 172–181.

[2] R.M. Aliguliyev, A new sentence similarity measure and sentence based extractive technique for automatic text summarization, Expert Systems with Applications 36 (4) (2009) 7764–7772.

[3] E.K. Park, D.Y. Ra, M.G. Jang, Techniques for improving web retrieval effectiveness, Information Processing and Management 41 (5) (2005) 1207–1223.

[4] L.C. Yu, C.H. Wu, F.L. Jang, Psychiatric document retrieval using a discourse-aware model, Artificial Intelligence 173 (7–8) (2009) 817–829.

[5] J. Allen, Natural Language Understanding, Benjamin Cummings, Redwood City, Calif., 1995

[6] J. O'Shea, Z. Bandar, K. Crockett, Towards a new generation of conversational agents based on sentence similarity, Advances in Electrical Engineering and Computational Science 39 (2009) 505–514.

[7] F.J. Pelletier, The principle of semantic compositionality, Topoi 13 (1994) 11–24.

[8] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, 1998.

[9] C. Bouras, V. Tsogkas, Noun retrieval effect on text summarization and delivery of personalized news articles to the user's desktop, Data & Knowledge Engineering 69 (7) (2010) 664–677.

[10] C.L. Chen, F.S.C. Tseng, T. Liang, Editorial: an integration of WordNet and fuzzy association rule mining for multi-label document clustering, Data & Knowledge Engineering 69 (11) (2010) 1208–1226.

[11] P. Wiemer-Hastings, All Parts are not Created Equal: SIAM-LSA, Proceedings of 26th Annual Conference of the Cognitive Science Society, 2004.

[12] Y. Li, D. McLean, Z. Bandar, J. O'Shea, K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Transactions on Knowledge and Data Engineering 18 (8) (2006) 1138–1149.

[13] A. Islam, D. Inkpen, Semantic text similarity using corpus-based word similarity and string similarity, ACM Transactions on Knowledge Discovery from Data 2 (2) (2008), article 10.

[14] T.K. Landauer, P.W. Foltz, D. Laham, Introduction to latent semantic analysis, Discourse Processes 25 (2–3) (1998) 259–284.

[15] P.W. Foltz, W. Kintsch, T.K. Landauer, The measurement of textual coherence with latent semantic analysis, Discourse Processes 25 (2–3) (1998) 285–307.

[16] C. Burgess, K. Livesay, K. Lund, Explorations in context space: words, sentences, discourse, Discourse Processes 25 (2–3) (1998) 211–257.

[17] J.H. Chiang, H.C. Yu, Literature extraction of protein functions using sentence pattern mining, IEEE Transactions on Knowledge and Data Engineering 17 (8) (2005) 1088–1098.

[18] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and Knowledge-based Measures of Text Semantic Similarity, Proceedings of the American Association for Artificial Intelligence, 2006, pp. 775–780.

[19] P. Achananuparp, X. Hu, X. Zhou, X. Zhang, Utilizing sentence similarity and question type similarity to response to similar questions in knowledge-sharing community, Proceedings of QAWeb 2008 Workshop, 2008.

[20] I. Dagan, O. Glickman, B. Magnini, Proceedings of the PASCAL Workshop 2006. The PASCAL recognizing textual entailment challenge, Machine Learning Challenges. Lecture Notes in computer Science 3944 (2006) 177–190.

[21] C. Corley, R. Mihalcea, Measuring the Semantic Similarity of Texts, Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment 2005, Ann Arbor, MI, 2005, p. 1318.

[22] L. Vanderwende, W.B. Dolan, What Syntax can Contribute in the Entailment Task, in: J.Q. Candela, I. Dagan, B. Magnini, F. d'Alché Buc (Eds.), Machine Learning Challenges Workshop, Springer, Berlin, 2006, pp. 205–216.

[23] R. Bar-Haim, I. Szpecktor, O. Glickman, Definition and Analysis of Intermediate Entailment Levels, Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment 2005, Ann Arbor, MI, 2005, pp. 55–60.

[24] F.M. Zanzotto, M. Pennacchiotti, A. Moschitti, A machine learning approach to textual entailment recognition, Natural Language Engineering 15 (4) (2009) 551–582.

[25] J. Herrera, A. Peñas, F. Verdejo, Textual Entailment Recognition Based on Dependency Analysis and WordNet, in: J.Q. Candela, I. Dagan, B. Magnini, F. d'Alché Buc (Eds.), Machine Learning Challenges Workshop, Springer, Berlin, 2006, pp. 231–239.

[26] J. Herrera, A. Peñas, A. Rodrigo, F. Verdejo, UNED at PASCAL RTE-2 Challenge, Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment (2006) 38–43.

[27] A. Rodrigo, A. Peñas, J. Herrera, F. Verdejo, Experiments of UNED at the third recognizing textual entailment challenge, Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing (2007) 89–94.

[28] T. Pedersen, S. Banerjee, S. Patwardhan, Maximizing Semantic Relatedness to Perform Word Sense Disambiguation, University of Minnesota, Supercomputing Institute, Research Report UMSI 2005/25, , 2005.

[29] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, IEEE Transactions on Systems, Man and Cybernetics 19 (1) (1987) 17–30.

[30] G. Hirst, D. St-Onge, Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database, MIT Press, 1998, pp. 305–332.

[31] P. Resnik, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995, pp. 448–453.

[32] J. Jiang, D. Conrath, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, Proceedings on International Conference on Research in computational Linguistics, 1997, pp. 19–33.

[33] D. Lin, Using Syntactic Dependency as a Local Context to Resolve Word Sense Ambiguity, Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, 1997, pp. 64–71.

[34] S. Banerjee, T. Pedersen, Extended Gloss Overlaps as a Measure of Semantic Relatedness, Proceedings of the Eighteenth International Joint conference on Artificial Intelligence, 2003, pp. 805–810.

[35] S. Patwardhan, Incorporating dictionary and corpus information into a context vector measure of semantic relatedness, Master's thesis, University of Minnesota, Duluth, 2003.

[36] R. Johansson, P. Nugues, Dependency-based Syntactic–Semantic Analysis with PropBank and NomBank, Proceedings Of CoNLL-2008 Shared Task, 2008, pp. 183–187.

[37] W. Dolan, C. Quirk, C. Brockett, Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources, Proceedings of the 20th International Conference on Computational Linguistics, 2004.

[38] H. Rubenstein, J.B. Goodenough, Contextual correlates of synonymy, Communications of the ACM 8 (10) (1965) 627–633.

[39] X. Liu, Y. Zhou, R. Zheng, Measuring Semantic Similarity in Wordnet, Proceedings of ICMLC2007 Conference, 2007, pp. 123–128.

[40] G. Pirro, A semantic similarity metric combining features and intrinsic information content, Data & Knowledge Engineering 68 (11) (2009) 1289–1308.

[41] P. Achananuparp, X. Hu, X. Shen, The Evaluation of Sentence Similarity Measures, Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery, 2008, pp. 305–316.

[42] S. Fernando, M. Stevenson, A Semantic Similarity Approach to Paraphrase Detection, Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2008, pp. 291–297.

[43] A. Budanitsky, G. Hirst, Evaluating WordNet-based measures of lexical semantic relatedness, Computational Linguistics 32 (1) (2006) 13–47.

[44] C. Corley, R. Mihalcea, Measuring the Semantic Similarity of Texts, Proceedings of the ACL workshop on Empirical Modeling of Semantic Equivalence, 2005, pp. 13–18.

Jesús Oliva was born in Guadalajara, Spain, in 1984. He received the BSc degree in Computer Science and the BSc degree in Mathematics by the Universidad Autónoma of Madrid (UAM), Spain, in 2007. He also obtained the MSc degree in Artificial Intelligence in 2008, by the Universidad Nacional de Educación a Distancia (UNED). Since 2007 he is with the Instituto de Automática Industrial (CSIC), as PhD student of the Bioengineering Group, in the Computer Science Department. His research interests include Cognitive and Computational Linguistics, Language Acquisition, Information Extraction and Probabilistic Reasoning.

J. Ignacio Serrano was born in Madrid, Spain, in 1977. He received the BSc degree in Computer Science by the Complutense University of Madrid (UCM), Spain, in 2001, and the MSc and PhD degrees in Artificial Intelligence and Software Engineering in 2004 and 2007, respectively, by UCM too. Since 2002 he is with the Instituto de Automática Industrial (CSIC), as PhD student and then as postdoctoral researcher, in the Computer Science Department, Bioengineering Group. His research interests include Cognitive Science, Computational Models of Cognition and Human Behavior, Computational and Cognitive Linguistics, Evolutionary Computation and Knowledge Discovery.

M. Dolores del Castillo was born in Madrid, Spain, in 1962. She got a BSc and MSc in Physics by Complutense University of Madrid (UCM), Spain, in 1984 and 1986, respectively. She got the PhD in Physics by UCM in 1990, although she has mostly worked in Knowledge Discovery, Machine Learning and Cognitive Science since then. She is a senior researcher of the scientific staff of the Instituto de Automática Industrial, Spanish National Research Council (CSIC), since 2005, when she joined the Bioengineering Group. Her research interests include Cognitive Science and Neuroscience, Knowledge Discovery, Machine Learning and Philosophy of Science.

Ángel Iglesias was born in Madrid, Spain, in 1982. He received the BSc and MSc degrees in Computer Science engineering by the Complutense University of Madrid (UCM), Spain, in 2005 and 2007 respectively. Since 2005 he is with the Instituto de Automática Industrial (CSIC), as granted PhD student of the Bioengineering Group, in the Computer Science Department. His research interests include Cognitive Decision Making, Fuzzy Reasoning, and Data Mining and Knowledge Discovery in Security applications.