Comparable Evaluation of Contemporary Corpus-Based and Knowledge-Based Semantic Similarity Measures of Short Texts

[1]Bojan Furlan, [2]Vladimir Sivački, [3]Davor Jovanović, [4]Boško Nikolić

[1](bojan.furlan@etf.bg.ac.rs), School of Electrical Engineering, University of Belgrade

[2](vsivacki@gmail.com), School of Electrical Engineering, University of Belgrade

[3](mdmdavor@yahoo.com), School of Electrical Engineering, University of Belgrade

[4](bosko.nikolic@etf.bg.ac.rs), School of Electrical Engineering, University of Belgrade

Case study

UDC **004.891:004.738.5**

Abstract: This paper presents methods for measuring the semantic similarity of texts, where we evaluated different approaches based on existing similarity measures. On one side word similarity was calculated by processing large text corpuses and on the other, commonsense knowledgebase was used. Given that a large fraction of the information available today, on the Web and elsewhere, consists of short text snippets (e.g. abstracts of scientific documents, image captions or product descriptions), where commonsense knowledge has an important role, in this paper we focus on computing the similarity between two sentences or two short paragraphs by extending existing measures with information from the ConceptNet knowledgebase. On the other hand, an extensive research has been done in the field of corpus-based semantic similarity, so we also evaluated existing solutions by imposing some modifications. Through experiments performed on a paraphrase data set, we demonstrate that some of proposed approaches can improve the semantic similarity measurement of short text.

Keywords: semantic similarity, corpus-based, knowledge-based

Introduction

The use of computers has changed our everyday lives, in a way of accelerated, automated and simplified job execution. Today, the information can be found fastest by using electronic resources, such as web pages. But, the large amount of information can greatly linger the search process. The problem is also in connecting questions in natural language with responses that are presented in electronic form.

This paper presents two methods for measuring the semantic similarity of texts, using corpus-based (CBSS) and knowledge-based (KBSS) measures of similarity. Previous work on this problem has focused mainly on either large documents (e.g. text classification, information retrieval) or individual words (e.g. synonymy tests). Given that a large fraction of the information available today, on Web and elsewhere, consists of short text snippets (e.g. abstracts of scientific documents, imagine captions, product descriptions), in this paper we focus on measuring the semantic similarity of short texts. A short text in typical human dialogue would be a sentence in the range of 10-20 words, bearing in mind that user utterances include other forms that fail to conform to the grammatical rules of sentences. A large number of software applications is based on the use of this kind of communication, for example in automatic processing of text and e-mail messages, natural language interfaces to databases, health care dialogue systems, online customer self-service, real estate sales, phone call routing and intelligent tutoring.

Therefore, this paper analyzes various techniques of short text processing based on existing similarity measures and presents their possible improvements. On one side word similarity was calculated by pro-

cessing large text corpuses and on the other commonsense knowledgebase was used. An extensive research has been done in the field of corpus-based semantic similarity, so we also evaluated existing solutions by imposing some modifications. Also, we focus on computing the similarity between two sentences or two short paragraphs by extending existing measures with information from the ConceptNet knowledgebase. Through experiments performed on a paraphrase data set, we show that by some of those approaches the semantic similarity measurement can be improved.

The rest of this paper is organized as follows: Section 2 considers some relevant features of corpus-based semantic similarity, implementation of discussed algorithms and evaluation of the results; Section 3 describes approach based on the knowledge-based semantic similarity. Section 4 outlines directions for future work.

CORPUS-BASED SEMANTIC SIMILARITY

There is a relatively large number of word-to-word similarity metrics that were previously proposed in literature, ranging from distance-oriented measures computed on semantic networks, to metrics based on models of distributional similarity learned from large text collections. From these, we chose to focus our attention on a corpus-based metrics. Corpus-based measures of word semantic similarity try to identify the degree of similarity between words using information exclusively derived from large corpora [3]. We applied different approach for calculating semantic word similarity that is based on the word-space models.

The general idea behind word-space models is to use distributional statistics to generate high-dimensional vector spaces, in which words are represented by *context vectors* whose relative directions are assumed to indicate semantic similarity. This assumption is motivated by the *distributional hypothesis*, which states that words with similar meanings tend to occur in similar contexts [9].

In the standard word space methodology, the high-dimensional vector space is produced by collecting the data in a co-occurrence matrix F, such that each row F_w represents a unique word w and each column F_c represents a context c, typically a multi-word segment such as a document, or another word. In the former case, where the columns represent documents, we call the matrix a *words-by-documents* matrix, and in the latter case where the columns represent words, we call it a *words-by-words* matrix. LSA [2] is an example of a word space model that uses document-based co-occurrences, and Hyperspace Analogue to Language (HAL, [7]) is an example of a model that uses word-based co-occurrences. COALS (*Correlated Occurrence Analogue to Lexical Semantic*) [4] is a method for deriving, from large text corpora, vectors representing word meanings, such that words with similar meaning have similar vectors and it is inspired by and highly related to the HAL and LSA methodologies. Random Indexing (RI) is word space approach, which presents an efficient, scalable and incremental alternative to standard word space methods [9].

In a corpus, terms co-occurrences is captured by means of a dimensionality reduction operated by singular value decomposition on the term-by-document matrix T representing the corpus. The cells F_{wc} of the co-occurrence matrix record the frequency of co-occurrence of word w and document or word c (Figure 1). The frequency counts are usually normalized and weighted in order to reduce the effects of high frequency words and, in case document-based co-occurrences are used, to compensate for differences in document size. On the Figure 1 we can no-

FIGURE 1 - AN EXAMPLE OF WORDS-BY-DOCUMENTS MATRIX

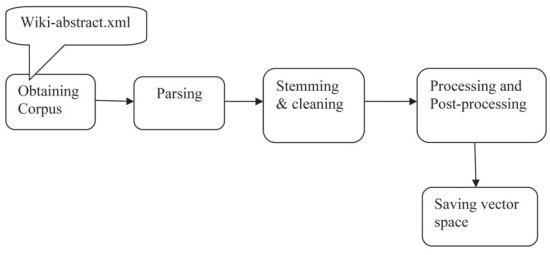
	DOC1	DOC2	DOC3	…	DOCn
W1	3	2	1		0
W2	4	2	1		0
W3	4	4	15		15
…					
Wm	2	2	2		2

tice that W_2 and W_m have very similar row vectors as a consequence of the distributional hypotheses.

CBSS Implementation

Figure 2 represents phases of implementation. The first stage is to obtain a corpus for generation of semantic space. Since Wikipedia abstracts dump (wiki-abstract.xml size of 1.7 GB) is in XML format; it has to be parsed to extract a flat text. The next phase is *stemming* and *cleaning*, which is the part of pre-processing phase. Stemming is a process of singling out a base of a word. For example, words: "fisher, fishing, fished" have the same base word "fish". Also, it is very common that large corpuses contain non-English words; therefore they have to be discarded by cleaning operation. The next phase is processing and post-processing, which is built upon implemented algorithms from SSPACE package. This package is a part of Google airhead open source project [4] and it implements large number of semantic analysis algorithms and provides possibility for developing new ones by using various utility classes (in our case COALS, RI and LSA). At the final stage, constructed semantic space has to be saved. Since it contains large amount of data it is impractical to keep it in a file, so we used database and its indexing functionalities, in order to obtain better performances for retrieving the specific vector for a given word.

Figure 2 – Implementation phases



Finally, for implementation of semantic similarity measure of sentences we applied algorithm explained in [3], by using string similarity and corpus-based word similarity, where for word similarity we used previously built semantic space model.

CBSS Evaluation

For the purpose of evaluation, we used *Microsoft Shared Paraphrase Corpus* (MSPC, [1]). It consists of 5081 pairs of sentences graded with a binary 0 for semantically non-similar and binary 1 for semantically similar. The MSPC itself is divided in two sets: train part (70% of the evaluation corpus) and the test part (other 30%). The train set is used to assess optimal threshold value, where samples with a value above the threshold are classified as similar and below as not similar. The threshold levels were evaluated in a range between 0.4 and 0.8, with a 0.1 increment, and optimal results on train part were found around threshold value of 0.6 with accuracy of 71% as shown at Table 1. Experiments were also carried on test part of the evaluation corpus, and results are shown in Table 2.

Table 1- The results on the train part of the corpus (70%)

Threshold	Accuracy
0.4	67.75%
0.5	69.27%
0.589	71.33%
0.6	71%
0.7	67.72%
0.8	57.4%

Table 2 - The results on the test part of the evaluation corpus (30%)

Threshold	Accuracy
0.4	66.7%
0.5	69.4%
0.589	70.32%
0.6	70.1%
0.7	67.8%
0.8	58%

The evaluation results on the test part were similar to results presented in [3]. However, we used different measure for calculating word similarity and also we processed different text corpus (Wikipedia abstracts dump) that is considerably smaller. Therefore, we assume that processing of larger corpus will increase accuracy of word similarity measure and consequently it will result in overall improvement of algorithm's accuracy. Also, one important algorithm's characteristic is that it showed good re-

sults with proper nouns that represent unique entities (specific names of countries, cities, people etc.), since it combines string similarity measure with semantic word similarity.

KNOWLEDGE-BASED SEMANTIC SIMILARITY

Another approach that we evaluated is based on algorithms that use ConceptNet knowledge base to extract and compare different concepts. ConceptNet is a semantic network that aims at providing common-sense knowledge to computers [5]. Its knowledge base is collected through an open source project called Open Mind Common Sense, where people can freely contribute with new knowledge. It has Python implemented Natural Language Processing (NLP) tools and many built-in tools for extracting valuable information from its knowledge base, such as methods for comparing two concepts, finding concepts that have the highest level of similarity to a given concept, etc. The similarity calculation is done by using Divisi, an implementation of AnalogySpace, which is a way of representing ConceptNet's common-sense knowledge base in a multi-dimensional vector space. MontyLingua [6] is also a Python implemented tool that is used for natural language understanding. Given a sentence, it can extract verb/subject/object tuples, as well as other semantic information.

KBSS IMPLEMENTATION

Our next approach was to combine the features of ConceptNet and MontyLingua to measure the semantic similarity of two text segments. ConceptNet offers a method for measuring similarity between two texts (lists of identified words), without taking into account the importance of a particular concept in the sentence. We tried to improve this method by adding a measure of a weight to each word, so the words with a bigger weight would factor more in the overall evaluation of sentence similarity. Next, we imposed a modification of a text similarity scoring function, defined in [8], where the similarity between the input text segments T_1 and T_2 is determined by using the following scoring function:

As in [8], each word ω from the first sentence T_1 is compared with words from the second sentence T_2, using ConceptNet's similarity function, so we could identify the word in the second sentence that has the highest level of similarity *(maxSim(*ω*,* T_2 *))*. The similarity is then multiplied with the word's weight and the resulting sum is normalized with the total sum of weights for all words from the sentence. The same method is applied to the sentence T_2 and finally the resulting similarity scores are combined using a simple average.

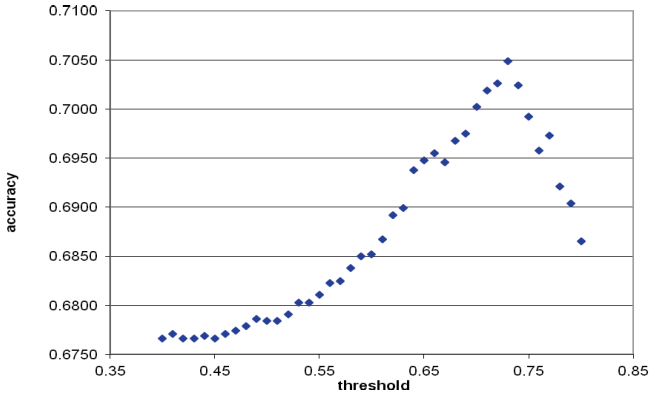
The scoring function originally used is *idf(ω)* instead of *weight(ω),* where *idf(ω)* stands for inverse document frequency, defined as the total number of documents in the corpus divided by the total number of documents that include the word *ω*. In our approach, we replaced inverse document frequency with the word's "weight" that represents its importance in the sentence. Assigning the optimal weight for each word was done by determining its role by extracting verb-subject-object tuples from a sentence with MontyLingua. Each word was then assigned with a weight, including some of the words not recognized by MontyLingua, and the scoring function was evaluated on MSPC corpus.

KBSS EVALUATION

The first step in experiment was to determine an optimal threshold for returned similarity values, where samples with a value above the threshold are classified as similar and below as not similar. The threshold levels were evaluated in a range between 0.4 and 0.8, with a 0.01 increment, and the results on the train part of the corpus (70%) are shown in the following Figure3.

$$sim(T_1,T_2)=\frac{1}{2}\left(\frac{\sum_{\omega\in\{T_1\}}\left(\max Sim(\omega,T_2)*weight(\omega)\right)}{\sum_{\omega\in\{T_1\}}weight(\omega)}+\frac{\sum_{\omega\in\{T_2\}}\left(\max Sim(\omega,T_1)*weight(\omega)\right)}{\sum_{\omega\in\{T_2\}}weight(\omega)}\right)$$

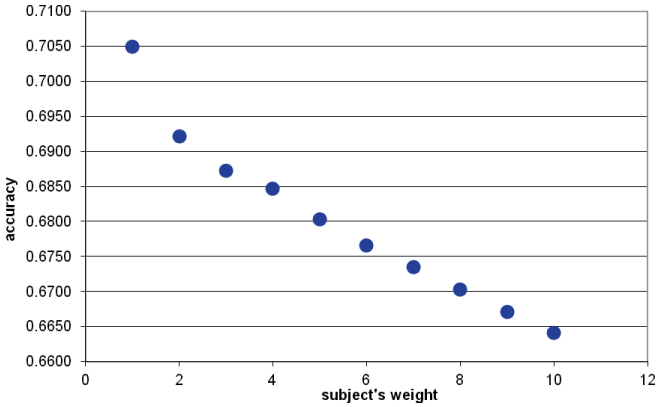
Figure 3 – The results on the train part of the corpus (70%)

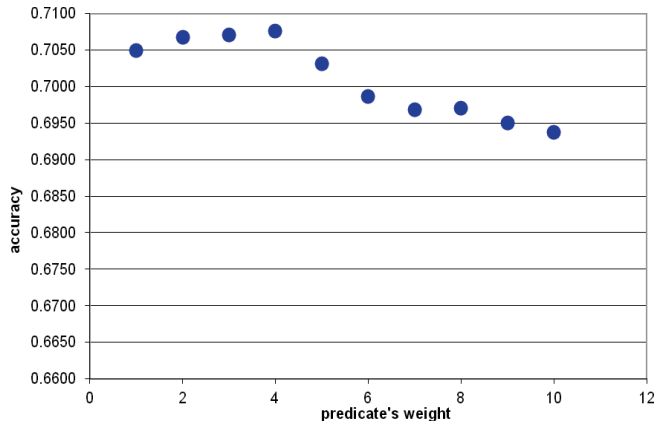


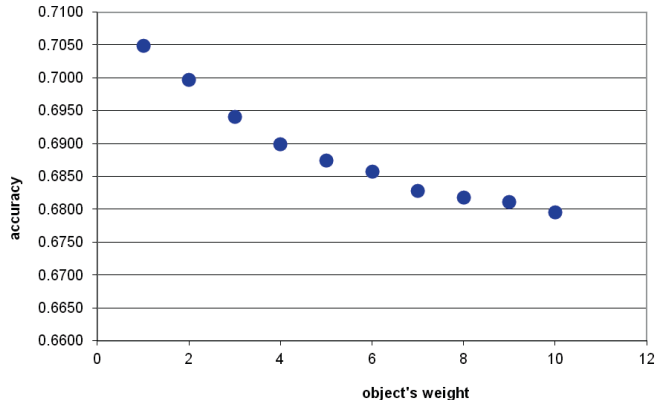
The results were according to expectations, since similar algorithms from literature [3, 8] had optimal threshold values around 0.7. In our case the best results were obtained with a threshold value at 0.73 and the algorithm produced the same results as a human judge in 70.49% of the cases.

Since the threshold evaluation was done by keeping the weight of the words in the sentence at a same level, for the next series of tests, we observed a correlation between the accuracy and the change of relative weight of specific parts of the sentence (predicate, subject and object). First, the weight of the predicate was increased from 1(the same weight as the other words) to 10, while keeping the weight of the other words constant. We repeated the procedure for the subject and object and the results are shown in the following Figure 4.

Figure 4. Correlation between the accuracy and the change of relative weight of specific parts of the sentence (predicate, subject and object)







The evaluation results showed that the measurement accuracy improved with the increase of predicate's weight, when comparing it to the given weight of the subject and object. Further increase of the weight of the subject or object resulted in a constant drop of the algorithm's performance. One of the reasons is that most of the words that appeared in these sentences as subjects or objects are proper nouns representing unique entities (specific names of countries, cities, people etc.). Since concepts can be transformed into vectors in AnalogySpace only if they are represented by four or more features in the database, such concepts were not taken into account when comparing sentence similarity. AnalogySpace as such works better with common nouns simply because it has more information to work with, which is important when generalizing and comparing concepts.

Since any increase of subject's or object's weight, while keeping the weight of the verb at an optimal level of 4, produced worse results, the conclusion was that the algorithm gave the best results with a threshold level of 0.73 and with the weights of verb,

subject and object at 4:1:1 respectively. Using these parameters, the results of the evaluation on the test part (other 30% of the MSPC), are presented in Table 3.

TABLE 3- THE RESULTS ON THE TRAIN PART OF THE CORPUS (OTHER 30% OF THE MSPC)

Number of pairs of sentences tested:	1725
Number of pairs where the algorithm reported an error:	38
Number of pairs where the algorithm gave the same result as the human judge:	1177
Relative accuracy rate (without unrecognized pairs):	1177/1687 = 0.6977 = 69.77%
Absolute accuracy rate:	1177/1725 = 0.6823 = 68.23%

Also, we evaluated ConceptNet's built-in algorithm for calculating semantic similarity of short text against the same corpus and its accuracy rate was 5% lower than the modified algorithm we previously presented.

CONCLUSION

We evaluated corpus-based measure, where we used different measure for calculating word similarity. We gained similar results, but with the considerably smaller processed corpus. Furthermore, since this algorithm, besides semantic word similarity measure, employs string similarity, it showed good results with proper nouns that represent unique entities and this was one of the main weaknesses of knowledge-based measure.

Given that a large fraction of the information available today, on the Web and elsewhere, consists of short text snippets (e.g. abstracts of scientific documents, image captions or product descriptions), where commonsense knowledge has an important role, we experimented on computing the similarity between two sentences or two short paragraphs by extending existing measures with information from the ConceptNet knowledgebase. The evaluation results showed that the measurement accuracy improved with the increase of predicate's weight, when comparing it to the given weight of the subject and object. Further increase of the weight of the subject or object resulted in a constant drop of the algorithm's performance. One of the main reasons is that most of the words that appeared in these sentences as subjects or objects are proper nouns representing unique entities (specific names of countries, cities, people etc.).

Therefore, the idea for further work is to extend the semantic text similarity measure that uses corpus-based word similarity and string similarity, by adding a measure of weight to each word, so the words with a bigger weight (importance) would factor more in the overall evaluation of sentence similarity.

Finally, it is also worth mentioning that the results were compared with those given by two human judges comparing the semantic similarity of the sentences. In some instances, they could not decide the similarity themselves, so a third judge was used to break the tie. This was interesting since the purpose of these and similar evaluations, of implementing and modifying algorithms for measuring the semantic similarity of two sentences, was an attempt to make the algorithms compare sentences the same way a human does when it still seems to be unclear how it is actually done. Thus, the main challenge is how to determine the best measure while a precise definition of that measure still remains unknown.

ACKNOWLEDGMENTS

The work presented here was partially supported by the Serbian Ministry of Education and Science

(projects III 44006 and 32047).

References:

[1] Dolan, W., Quirk, C. and Brockett, C. (2004). "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," *20th International Conference on Computational Linguistics*.

[2] Dumais, S. (2004). "Latent semantic analysis," *Annual Review of Information Science and Technology*, vol. 38, p. 188–230.

[3] Islam, A. and Inkpen, D. (2008). "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, Jul. 2008, pp. 1-25.

[4] Jurgens, D. and Stevens, K. (2010). "The S-Space package: an open source package for word space models," *System Papers of the Association of Computational Linguistics.*

[5] Liu, H. and Singh, P. (2004). "ConceptNet — A Practical Commonsense Reasoning Tool-Kit," *BT Technology Journal*, vol. 22, Oct. pp. 211-226.

[6] Liu, H. and Singh, P. (2004). "Commonsense Reasoning in and over Natural Language," *International Conference on Knowledge-Based Intelligent Information & Engineering Systems*

[7] Lund, C. & Burgess, K. (1996). "Producing high-dimensional semantic spaces from lexical co-occurrence," *Behavior Research Methods, Instrumentation, and Computers*, pp. 203-208.

[8] Mihalcea, R, Corley, C. and Strapparava, C. (2006). "Corpus-based and knowledge-based measures of text semantic similarity, "*Proceedings of the National Conference on Artifi cial Intelligence*, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 775.

[9] Sahlgren, M. (2005). "An introduction to random indexing," *In Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering.*

Submitted: April 28, 2011
Accepted: June 09, 2011