TALN 2011, Montpellier, 27 juin – 1er juillet 2011

Corpus-Based methods for Short Text Similarity

Prajol Shrestha
LINA-UFR Sciences, 44322 Nantes Cedex 3
prajol.shrestha@etu.univ-nantes.fr

Résumé. Cet article concerne la détermination de la similarité entre des textes courts (phrases, paragraphes, ...). Ce problème est souvent abordé dans la littérature à l'aide de méthodes supervisées ou de ressources externes comme le thesaurus Wordnet ou le British National Corpus. Les méthodes que nous proposons sont non supervisées et n'utilisent pas de connaissances à priori. La première méthode que nous présentons est basée sur le modèle vectoriel de Salton auquel nous avons apporté des modifications pour prendre en compte le contexte, le sens et la relation entre les mots des textes. Dans un deuxième temps, nous testons les mesures de Dice et de ressemblance pour résoudre ce problème ainsi que l'utilisation de la racinisation. Enfin, ces différentes méthodes sont évaluées et comparées aux résultats obtenus dans la littérature.

Abstract. This paper presents corpus-based methods to find similarity between short text (sentences, paragraphs, ...) which has many applications in the field of NLP. Previous works on this problem have been based on supervised methods or have used external resources such as WordNet, British National Corpus etc. Our methods are focused on unsupervised corpus-based methods. We present a new method, based on Vector Space Model, to capture the contextual behavior, senses and correlation, of terms and show that this method performs better than the baseline method that uses vector based cosine similarity measure. The performance of existing document similarity measures, Dice and Resemblance, are also evaluated which in our knowledge have not been used for short text similarity. We also show that the performance of the vector-based baseline method is improved when using stems instead of words and using the candidate sentences for computing the parameters rather than some external resource.

Mots-clés : Similarité, Modèle Vectoriel, Mesure de Similarité.

Keywords: Similarity, Vector Space Model, Similarity metric.

1 Introduction

Many natural language processing applications use similarity between short text (e.g. sentences, paragraphs) such as text summarization (Lin & Hovy, 2003), which works on sentence or paragraph level ; question answering, which uses similarity between the question answer pairs (Mohler & Mihalcea, 2009) ; and image retrieval, where an image is retrieved by finding the similarity between the query and the image caption (Coelho *et al.*, 2004).

In general, existing methods view the short text similarity problem as a classification task, where one-to-one text similarity decision is made, or as an alignment task, where many-to-many text similarity decision is also made. Most of the existing methods treat this problem as a classification task which use similarity metrics (Abdalgader & Skabar, 2011)(Cordeiro *et al.*, 2007)(Mihalcea & Corley, 2006)(Hatzivassiloglou *et al.*, 1999). These similarity metrics give a value of similarity between pairs of short text which can then classify the pairs as similar or not using a threshold. This threshold value is usually empirically fixed for each similarity metric. These existing methods use external knowledge like WordNet (Miller *et al.*, 1990) to find lexical similarity or some corpora like the British National Corpus for optimizing parameters. These methods are not suitable for languages that do not have resources like WordNet or large corpora. This leads to find similarity measures that use no resources or resources that are easily buildable. There are few methods that treat the similarity problem as an alignment task (Barzilay, 2003)(Nelken & Shieber, 2006). These methods also use similarity metrics like the classification methods but the value from these metrics are not used directly for alignment. The alignment in these methods are based on supervised methods and use dynamic programming which includes the context of the sentences and are designed for comparable monolingual text.

We take the similarity problem between short text as a classification task. This task is based on corpus-based unsupervised methods which do not use external resources to compute the similarity value unlike existing classification methods. One of the earliest and well known classification method for text similarity is the Vector Space Model (VSM) for information retrieval, where similar documents in a collection is chosen by a similarity value computed using a similarity metric, the cosine similarity, between the vectors of term weights representing documents (Salton *et al.*, 1975). This measure is based on the overlap of terms in the document pair whose similarity is being measured.

The VSM assumes that the vectors of terms are independent, pairwise orthogonal, to each other which is unrealistic. There exist other vector space models like the Generalized Vector Space Model (GVSM) (Wong *et al.*, 1987) which does not assume this independence and although this model claims to be more effective than the standard implementation of the VSM, it is computationally expensive and therefore VSM is widely used despite its unrealistic assumption. The VSM for information retrieval is modified and used for short text similarity by treating the short text as documents and computing the idf value using external resources like the British National Corpus. This VSM cosine similarity measure is the baseline for most of the similarity studies (Mihalcea & Corley, 2006).

Our similarity measure is based on VSM but is adopted in such a way that the assumption of term independence is excluded and the short text vectors incorporates the sense and correlation of the terms. This is done by taking into account the overlap of the terms in all the short text of the corpus rather than only the short text pair between which the similarity is measured. Along presenting a new method to find similar short text we evaluate the performance of two other information retrieval methods which use term overlaps namely Dice measure (Manning & Schütze, 1999) and Resemblance measure [1] (Lyon *et al.*, 2001). We also show that using stems instead of words can improve the baseline VSM model for short text similarity.

2 Related Works

In information retrieval, there are many methods to find similarity between documents and one of the most well known method is the VSM which uses cosine similarity measure (Barron-Cedeno *et al.*, 2009). This vector based method is also used to measure similarity between sentences as done by Barzilay et al. (Barzilay, 2003). They view the problem of finding similar sentences as an alignment problem, where they align similar sentences between two monolingual comparable documents. In their method, the paragraphs are first aligned by a trained classifier and once the paragraphs are aligned the sentences within them are aligned using vector based cosine similarity and

1. also known as the Jaccard or Tanimoto coefficient (Manning & Schütze, 1999)

dynamic programming. Rani et al. (Nelken & Shieber, 2006) took the same problem as Barzilay et al. and proposed an improved robust method. This method also uses dynamic programming for alignment but uses cosine similarity measure in a logistic regression to provide a score to aid the alignment. Both of these methods use context around the sentences, the following and preceding sentences, to aid in alignment. These methods for alignment make many-to-many sentence alignment and do not provide a similarity value between sentences indicating that these methods are suitable to find similar sentences only in comparable monolingual corpora.

Another method that uses the concept of overlap like the cosine measure is the fingerprinting method. It takes into account the overlap of bigrams or trigrams between the text to calculate a value of resemblance as shown in Lyon et al.(Lyon *et al.*, 2001) which is the basis of classifying similar text and has been used to detect plagiarism. Cordeiro et al. (Cordeiro *et al.*, 2007) has also proposed a similarity metric to identify similarity between texts and to identify paraphrase based on word overlap. It computes a similarity value by combining the ratio of common words in each sentence and is focused on capturing paraphrases which makes it unsuitable to find other types of similar sentences for example, this metric gives a similarity value zero to identical sentences.

Linguistic features has also been used to find similarity between short text as in Hatzivassiloglou et al. (Hatzivassiloglou *et al.*, 1999). They build linguistic feature vectors to build rules in a supervised manner to classify paragraph pairs. The features used to build rules are noun phrase matching, WordNet synonyms, common word class of verbs, shared common noun and their combinations. Even though it performs better than the vector based cosine similarity measure, it requires resources like Wordnet which are not present and are hard to build for other resource less languages.

Recent researches are focused on finding the similarity between lexical items in short text to find the similarities between these text. There exist corpus-based approaches to find the lexical similarity, some of which use text pattern analysis, Pointwise mutual information (PMI) and Latent Semantic Analysis (LSA). We will not focus on WordNet based approaches to find lexical similarity (Abdalgader & Skabar, 2011). Mihalcea et al. (Mihalcea & Corley, 2006) use PMI and LSA to compute the text semantic similarity using a wrapper given in equation 1.

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\Sigma_{w \in T_1} maxSim(w, T_2) * idf(w)}{\Sigma_{w \in T_1} idf(w)} + \frac{\Sigma_{w \in T_2} maxSim(w, T_1) * idf(w)}{\Sigma_{w \in T_2} idf(w)}\right) \quad (1)$$

$maxSim(w, T_{2(1)})$ is the maximum lexical similarity between the word w in sentence $T_{1(2)}$ and all the words in sentence $T_{2(1)}$ and $idf(w)$ is the inverse document frequency of the word w calculated from the British National Corpus. The similarity metric, STS, proposed by Islam et al. (Islam & Inkpen, 2008), unlike other metrics, use string similarity along with corpus-based word similarity. Corpus-based word similarity is measured using two measures that includes second order co-occurrence pointwise mutual information and common word order similarity. The string similarity is measured using the concept of Longest Common Sequence. All these three measures are combined to determine the similarity between two short text. All the mentioned corpus-based method have the same drawback of using external resources.

3 Short Text Similarity

In this section, we present a new method to find similarity between short text. For simplicity reasons, we explain the method using sentence as our short text. Our similarity method is based on VSM but is different in the way the sentence vectors are created. The dimensions in sentence vectors do not represent the terms in the collection of sentences as in the bag of words model (Baeza-Yates & Ribeiro-Neto, 1999) but rather created from term vectors. Given a corpus C of n sentences and m unique terms, the term vector for term t_j is created with n number of dimensions in which the presence and absence of the term in each sentence is indicated by a boolean value x :

$$\vec{t_j} = [x_1, x_2, x_3, x_4, ..., x_n] \quad x_i \in 0, 1; i \in 1 \text{ to } n; 0 = absent, 1 = present \quad (2)$$

This term vector representation is similar to the wordspace model (Schutze, 1998) where the distribution of the terms are stored. These representation of term vectors together will form a m x n term-sentence matrix and as the number of sentence increases the size of the matrix will also increase. This huge dimension of the matrix can be reduced to some extent by removing stopwords and stemming [2]. There are also mathematical procedures for the reduction of the matrix like Latent Semantic Analysis (Deerwester *et al.*, 1990) which uses singular value decomposition, SVD, or Principle Component Analysis (Jolliffe, 1986) which represents the matrix in different

2. http ://snowball.tartarus.org/

coordinates. We have used the simple technique of removing the stopwords and stemming words but none of the mathematical procedures to reduce the dimension during our experiments. This representation of term vector will consist of many zero values which will take a lot of memory. To reduce this space, we represent the vector in a reduced form where only the dimensions having value 1 are kept as shown in Equation 3 where we assume that the term t_j is present in sentence numbers 1,5, and 8 :

$$\vec{t_j} = [(S_1, 1), (S_5, 1), (S_8, 1)] \quad S_i \text{ is the sentence number } i \text{ where the term } t_j \text{ is present ; } i \in 1 \text{ to } n \tag{3}$$

This term vector shows the different senses that the term may have. Here, the sense of the term means the idea with which it can be related to. Our assumption is that sentences are independent to each other making each sentence presenting a unique idea and therefore, each term present in a sentence is related to this idea. This assumption like the assumption of VSM is unrealistic but the effect of this assumption can be reduced using clustering techniques like hierarchical clustering (Han & Kamber, 2006) to group sentences that give the same idea or in other words similar sentences. Clustering has not been used in the experiments. Once we have the term vectors we can create sentence vectors by adding the term vectors of the terms present in the sentence making the number of dimension of their sentence vector equal to the term vector. The term vector consists of only the boolean value to be added which doesn't provide much information about the term so while adding the term vector we add the inverse document frequency, idf, value of the term which in our case is the inverse sentence frequency. This idf value is computed from the sentences present in the corpus. For a sentence consisting of terms $t_1, t_2, .., t_n$, the dimension, i, corresponding to the sentence S_i of the sentence vector will be :

$$d_i = \Sigma_{j=1;t_j \in S_i}^{n} idf_j \quad idf_j \text{ is the idf value of the term } j \text{ ; } i \in 1 \text{ to } n \tag{4}$$

This method is similar to the method of second-order similarity (Kaufmann, 2000) and includes more information other than cohesion of text by encoding three different information in the sentence vector which are $i)$ the importance of each term using its idf $ii)$ the co-occurrence of terms by adding up the idf values of all the terms that occur in a sentence and $iii)$ the distribution of term along various sentences as the dimensions of the sentence vector is equal to the number of sentences present in the corpus. Using these sentence vectors we can now compute the similarity value between two sentences using the cosine similarity measure. We name our method Short text based Vector Space Model, SVSM, to distinguish it from the other vector based models. This method can be easily used to find similarity between other types of short text by directly using the new type of short text instead of sentences.

4 Experiments and Results

We used the Microsoft Research Paraphrase Corpus(MSRPC) (Dolan *et al.*, 2004) to evaluate our sentence similarity method which consists of 5801 pairs of sentences collected from a range of online newswire over a period of 18 months for experiments. This dataset is divided into 4076 training pairs and 1725 test pairs. The training pairs consist of 3900 paraphrases and the test pairs consist 1147 paraphrase. The remaining sentence pairs in the corpora are not paraphrases. We test our method on these test pairs and compare results with other methods which are tested on the same corpus. We also evaluated the performance of Dice measure, Resemblance measure and an adaptation of the VSM cosine similarity measure on the same test corpus. Resemblance is the method explained in section 2 and the Dice measure follows the same principle of term overlaps whose similarity value is given by the ratio between twice the number of term overlaps and the total number of terms in both the sentences (Manning & Schütze, 1999). The adopted VSM cosine similarity measure, vector-based (A), is explained in section 1 and uses stems instead of words and calculating the idf value from the given corpus rather than using some external one. The evaluations of these methods are given in Table 1 where the evaluation value named accuracy represents the number of correctly identified true or false classifications (Mihalcea & Corley, 2006) and the rest of the evaluation values bare their traditional meaning. In Table 1, the first two section of the table presents the best results according to the highest accuracy achieved by increasing the threshold by 0.1. The remaining results from the other three sections are taken from Abdalgader et al.(Abdalgader & Skabar, 2011).

Table 1 shows two baseline methods. The random method is the method which randomly assigns similarity values to the sentence pairs and the vector-based method is the VSM based cosine similarity measure between two sentences with tf*idf term weights computed using external corpus. All our experiments were done with stems as terms and without stopwords. The results for our SVSM shows improvement over the baselines with higher recall but are not better than existing methods. Our SVSM method uses the distribution of terms across sentences from which it captures the sense of the term and the correlation between other terms which leads us to believe that

Methods	Threshold	Accuracy	Precision	Recall	F-measure
Proposed Method					
SVSM	0.7	68.9	71.7	87.9	79.0
IR Methods					
Vector-based (A)	0.4	71.0	71.0	**95.4**	**81.4**
Dice	0.5	70.6	72.7	89.2	80.2
Resemblance	0.1	68.1	70.6	89.1	78.8
Islam & Inkpen (2008) Corpus-based					
STS	0.6	72.6	74.7	89.1	81.3
Mihalcea et al. (2006) Corpus-based					
PMI-IR	0.5	69.9	70.2	95.2	81.0
LSA	0.5	68.4	69.7	95.2	80.5
Baselines					
Vector-based	0.5	65.4	71.6	79.5	75.3
Random	0.5	51.3	68.3	50.0	57.8

TABLE 1 – Different methods to detect similarity between sentences and their performance according to the accuracy, precision, recall, and F-score on the MSR paraphrase detection test set are shown for the given thresholds.

the more sentences we have in our corpus the better it will perform. We evaluated this method on a larger set of sentence pairs by using the complete MSRP corpus and found that the method does perform better. The result is shown in the Table 2.

Methods	Threshold	Accuracy	Precision	Recall	F-measure
Proposed Method					
SVSM	0.6	68.3	70.25	91.7	79.6
IR Methods					
Vector-based (A)	0.4	70.8	71.2	94.9	81.4

TABLE 2 – SVSM and VSM based cosine similarity results on the complete MSRP corpus with their performance evaluated according to the accuracy, precision, recall, and F-score for the given thresholds.

The Dice measure and Resemblance measure perform better than the baseline methods and have similar F-measure values with the existing methods. The evaluation of the vector-based (A) method shows that this method is among the best corpus-based sentence similarity methods with higher precision and recall values and Table 2 shows that even with larger collection of text this method performs equally well.

5 Conclusion and Discussion

In this paper we introduce a new method (SVSM) to compute the similarity between short text which takes the similarity problem as a classification task. This new method is a modified version of VSM and is similar to the second-order similarity method (Kaufmann, 2000). This method is able to capture the similarity between short text by using short text vectors which encodes three corpus-based information which are the importance of the term as idf, the distribution of terms in the short text of the corpus which represent the sense of the terms and the correlation between terms present in the pair of short text. SVSM performs better than the baseline methods with high recall and has the potential to perform better with more text available to be able to model the language by encoding the three information it utilizes. Even though this method assumes that the short text are independent of each other, which is unrealistic, we believe that the effect of this assumption can be reduced by using stems and clustering techniques. This belief has not been tested and will be incorporated in our future work.

We also show that stemming increases the performance of vector-based baseline and is one of the best corpus-based sentence similarity method that exist at least for english. We also use two other information retrieval measures, Dice and Resemblance, to find similar sentences and see that they do perform better than the baseline methods. All these experiments have been done on the MSR paraphrase corpus which does not contain other types of similar short text other than paraphrase and therefore, the results only partially represent the ability of the techniques to determine similarity. Further experiments of other types of short text pairs must be done to understand the full extent of the ability of our method.

Références

ABDALGADER K. & SKABAR A. (2011). Short-text similarity measurement using word sense disambiguation and synonym expansion. *AI 2010 :Advances in Artificial Intelligence, Lecture Notes in Computer Science*, **6464**, 435–444.

BAEZA-YATES R. & RIBEIRO-NETO B. (1999). *Modern Information Retrieval*, volume 463. Addison Wesley.

BARRON-CEDENO A., EISELT A. & ROSSO P. (2009). Monoligual text similarity measures : A comparison of models over wikipedia articles revisions. *Proceedings of ICON-2009 : 7th International Conference on Natural Language Processing*.

BARZILAY R. (2003). Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, p. 25–32.

COELHO T. A. S., CALADO P. P., SOUZA L. V., RIBEIRO-NETO B. & MUNTZ R. (2004). Image retrieval using multiple evidence ranking. *TKDE*, **16**(4), 408–417.

CORDEIRO J., DIAS G. & BRAZDIL P. (2007). Learning paraphrases from wns corpora. In *Proceedings of the 20th Int. FLAIRS Conf. AAAI Press*, p. 193–198.

DEERWESTER S., DUMAIS S. T., FURNAS G. W., LANDAUER T. K. & HARSHMAN R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, **41**(6), 391–407.

DOLAN B., QUIRK C. & BROCKETT C. (2004). Unsupervised construction of large paraphrase corpora : Exploiting massively parallel news sources. *20th International Conf. on Computational Linguistics*, p. 350–356.

HAN J. & KAMBER M. (2006). *Data Mining : Concepts and Techniques*. Number Edition, Second. Morgan Kaufmann.

HATZIVASSILOGLOU V., KLAVANS J. L. & ESKIN E. (1999). Detecting text similarity over short passages : Exploring linguistic feature combinations via machine learning. In *Proceedings of the 1999 joint sigdat conference on empirical methods in natural language processing and very large corpora*, p. 203–212.

ISLAM A. & INKPEN D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. In *ACM Transactions on Knowledge Discovery from Data Vol. 2, Article 10*.

JOLLIFFE I. T. (1986). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, **2**(1-3), 37–52.

KAUFMANN S. (2000). Second-order cohesion. *Computational Intelligence*, **16**(4), 511–524.

LIN C.-Y. & HOVY E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. *Proceedings of Human Language Technology Conference*.

LYON C., MALCOLM J. & DICKERSON B. (2001). Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, p. 118–125.

MANNING C. D. & SCHÜTZE H. (1999). *Foundations of Statistical Natural Language Processing*, volume 26. MIT Press.

MIHALCEA R. & CORLEY C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI 06*, p. 775–780.

MILLER G. A., BECKWITH R., FELLBAUM C., GROSS D. & MILLER K. (1990). Wordnet : An on-line lexical database. *International Journal of Lexicography*, **3**, 235–244.

MOHLER M. & MIHALCEA R. (2009). Text-to-text semantic similarity for automatic short answer grading. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on EACL 09*, (April), 567–575.

NELKEN R. & SHIEBER S. M. (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

SALTON G., WONG A. & YANG C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, **18**(11), 613–620.

SCHUTZE H. (1998). Automatic word sense discrimination. *Journal of Computational Linguistics*, **24**, 97–123.

WONG S. K. M., ZIARKO W., RAGHAVAN V. V. & WONG P. C. N. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems TODS*, **12**(2), 299–321.