Comparison of Statistical and Semantic Similarity Techniques for Paraphrase Identification

Savitha Sam Abraham
Department of Computer Science, CUSAT
savithasam88@gmail.com

Sumam Mary Idicula
Department of Computer Science, CUSAT
sumam@cusat.ac.in

Abstract

This paper compares statistical technique of paraphrase identification to semantic technique of paraphrase identification. The statistical techniques used for comparison are word set and word-order based methods where as the semantic technique used is the WordNet similarity matrix method described by Stevenson and Fernando in [3].

1. Introduction

Paraphrase is an expression of the same message in different words. Paraphrase identification has many applications in the areas of information retrieval, information extraction, natural language processing and etc.

There are several paraphrase identification techniques-both statistical measures and semantic techniques. In statistical methods, the similarity between sentences is measured based only on the statistical information of sentences. Some of the statistical measures include similarity based on edit distance, word vector, word set, word order etc. Semantic similarity approach makes extensive use of information about similarities between word meanings. Semantic measures can be corpus based or knowledge based. Corpus based measures try to identify the similarity between words using information exclusively derived from large corpora like Brown Corpus. Knowledge based measures use information drawn from semantic networks like WordNet. WordNet provides many similarity metrics.

The purpose of this project is to compare statistical technique to semantic technique. The statistical techniques selected for comparison in this paper are word set based and word order based sentence similarity techniques since they are the most successful among the statistical measures as described by Zhang in [2]. Similarly the most successful among the semantic measures is the WordNet similarity matrix method described by Stevenson and Fernando in [3].

The outline of the paper is as follows. Section 2 describes the resources used namely the dataset used and WordNet. Section 3 describes some of the previous statistical and semantic approaches for paraphrase identification. Section 4 describes the experiments performed and section 5 gives the result of these experiments. Conclusions and suggestions for future work are presented in section 6.

2. Resources

2.1. Microsoft Research Paraphrase Corpus (MSRPC)

The dataset used is a subset of MSRPC. MSRPC originally contains 5801 sentence pairs. It is available as training set which contains 4076 sentence pairs and a testing set which contains the remaining 1725 sentence pairs. These sentence pairs are taken from web news sources. Every sentence pair has a binary classification associated with it which says whether the pair is considered a paraphrase or not by the human judges. Over 100,000 articles were collected and were clustered into 11,000 clusters based on the topic. Two strategies were used to decide which of the sentence pairs would be useful examples. The first strategy used to filter out the sentences was the edit distance metric. Each sentence was converted to lower case and was paired with every other sentence. Identical sentences or those only different in punctuation were removed. Edit distance used was the Levenshtein distance. Levenshtein distance was calculated for every pair and if the distance n<=12, then they were selected. This formed the L12 dataset which contained 139K sentence pairs. The second strategy used was based on the tendency of journalists to summarize the content of an article in the first two sentences. Hence the first two sentences of each article was taken and paired with first two sentences of every other article. Certain

978-1-4673-2149-5/12/$31.00 ©2012 IEEE

heuristics are then used to filter out useful sentence pairs from these. One example for heuristic is, we find the number of common words in the pair. If at least three words of greater than four characters are same, then the sentence pairs were considered similar. Based on this, 214K sentence pairs were selected and were called the F2 dataset. Word Alignment Error Rate (AER) was then used to measure the quality of the sentence pairs. It was seen that L12 dataset had more number of paraphrases but F2 dataset had richer paraphrase examples. Next, human raters were made to classify the sentence pairs based on whether they are semantically equivalent or not. Further filtering techniques reduced the dataset to 5801 pairs.

2.2. WordNet

WordNet is a lexical database. A synset is a set of lexical items which are synonymous. WordNet consists of many such synsets which are interlinked by many relations like hypernym or is-a relation, meronym or part-of relation etc. There are methods for determining the similarity of pairs of words using the WordNet hierarchy. WordNet only contains is-a hierarchies for verbs and nouns. Hence similarity between a pair of words can be found only if either both the words are nouns or both the words are verbs. Some of the best performing WordNet similarity metrics are lch metric (Leacock and Chodorow, 1998), wup metric (Wu and Palmer, 1994), res metric (Resnik, 1995), lin metric (Lin, 1998) and jcn (Jiang and Conrath, 1997).

The lch metric finds the similarity between two nodes by calculating the path length between the nodes in a is-a hierarchy.

$$Sim_{lch} = - \log (N_p / 2D)$$

where N_p is the distance between the nodes and D is the maximum depth in is-a taxonomy.

The wup metric finds similarity of nodes as a function of the path length from the least common subsumer (LCS) of the nodes. If there are two concept nodes C1 and C2, then LCS of these two nodes is defined as the most specific node which both shares as an ancestor.

$$Sim_{wup} = (2 * N_3) / (N_1 + N_2 + 2 * N_3)$$

where N_1 is the number of nodes from LCS to C1, N_2 is the number of nodes from LCS to C2 and N_3 is the number of nodes from root node to LCS.

The res metric makes use of information content (IC) of the LCS of two nodes whose similarity is to be determined. Information content of a concept says how informative the concept is. If the concept occurs very frequently, then it will have lower IC and vice versa. Hence IC of a concept c is:

$$IC(c) = - \log P(c)$$

where P(c) is the probability of finding c in a large corpus. The resnik metric uses IC of the LCS based on the notion that two nodes that are more similar will share more amount of information and IC of LCS of two nodes represents the amount of information that the two nodes share.

$$sim_{res} = IC (LCS(C1, C2))$$

The lin metric builds on res metric by normalizing using the information content of two nodes themselves.

$$Sim_{lin} = (2 * IC (LCS (C1,C2))) / (IC(C1) + IC(C2))$$

The jcn metric uses the same information combined in a different manner:

$$Sim_{jcn} = \frac{1}{IC(C1) + IC(C2) + 2 * IC(LCS(C1,C2))}$$

3. Previous works

There has been many works on paraphrase detection and evaluation of different paraphrase identification techniques. Section 3.1 describes the different statistical measures available and section 3.2 describes the different semantic techniques available.

3.1. Statistical measures

Statistical measures consider only the spellings and ignore the semantic meanings of words.[2] describes statistical measures like similarity based on word set, word order, word vector, edit distance and word distance.

Word set requires constructing the word set of each of the sentences and then using either jaccard similarity or dice similarity to compute similarity. Word order requires constructing the word order vector of the two sentences. According to positions of words in a sentence, the orders between the word pairs such as before and after could be established. In word distance based sentence similarity the distance between word pairs is considered and it takes the form (w1, w2, d) where w1 and w2 are two words and d is the distance between them. Similarity based on word vector is one

of the earliest methods where the word vectors of the sentences will be constructed and cosine similarity between sentences is calculated. A weight will be assigned to each word which can be based on the number of all the words or on TF-IDF obtained from a corpus. Sentence similarity based on edit distance makes use of spellings of words in the two sentences. There are different edit distances: Levenshtein, Hamming, Jaro-Winkler distance etc. [2] evaluates the performances of these methods and they are given in Table 1.

Measure	Precision	Recall	F-measure
Word set	1	1	1
Averaged weight vector	0.9797	0.9797	0.9797
Edit distance	0.8786	0.8786	0.8786
Word order	0.9932	0.9932	0.9932
Word distance	0.9730	0.9730	0.9730
TF-IDF weighted vector	0.8952	0.8952	0.8952

Table 1: Results of evaluation across statistical measures

Since word set and word order outperforms the rest of the methods they will be used for comparison in this paper.

3.2. Semantic Techniques

An approach that goes beyond simple lexical matching is described in [4] where for each word in first sentence, the most similar word to it is found in second sentence and then these maximal scores are added up. Word-to-word similarity measures and a word specificity measure were used to estimate the semantic similarity. Word-to-word similarity was calculated using corpus based or knowledge based measures. Knowledge based measures made use of the WordNet similarity metric described in section 2.2. The corpus based measures like pointwise mutual information (PMI) use information gained from large corpora. The evaluation in [4] showed that best results were obtained by combining the different knowledge based measures.

The method proposed in [3] considers similarity scores between all word pairs unlike in [4] which consider only maximal scores. This method is also called the similarity matrix method. This method outperforms the one described in [4].

The approach developed in [5] is a two phase process where in the first phase all information nuggets in each sentence are identified and identical ones are paired off. If there are unpaired nuggets, their significance is calculated. The approach in [6] was based on converting text into canonicalized forms which are then compared. It was based on the concept that similar sentences will have identical surface texts.

Semantic similarity based on similarity matrix is one of the most successful techniques and hence it will be used in our experiments for comparison.

Metric	Precision	Recall	Fmeasure
matrixLin	74.9	91.2	82.2
Mihalcea, 2006	69.6	97.7	81.3
Qiu ,2006	72.5	93.4	81.6
Zhang, 2005	74.3	88.2	80.7

Table 2: Results of evaluation across semantic measures

4. Experiments

For comparing statistical and semantic techniques, most successful among statistical and semantic techniques were taken. Among statistical techniques, word set and word order measures were taken and among semantic techniques similarity matrix method was selected. All the methods are implemented in PERL in these experiments.

4.1. Word set and word order based similarity

The only preprocessing needed for statistical technique is tokenization. String::Tokenizer module from CPAN is used for tokenizing the text.

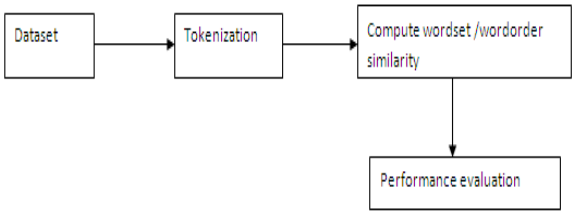

Figure 1: Word set and word order schematic diagram

In sentence similarity based on word set, the word sets of the two sentences are formed first. There are two ways here – one is to form the word set of the sentences with the original words in the sentences,

second is to used stemmed words. The first method is used since it takes into consideration the tense form and voice information. Let $w(S_a)$ be the set of words in first sentence Sa and $w(S_b)$ be the set of words in second sentence S_b. After the word set is formed, jaccard similarity or dice similarity is computed as:

$$\text{Jaccard}(S_a,S_b) = \frac{|\, w(S_a) \cap w(S_b) \,|}{|w(S_a) \cup w(S_b)|}$$

$$\text{Dice}(S_a, S_b) = \frac{2\,|\, w(S_a) \cap w(S_b) \,|}{|w(S_a)| + |w(S_b)|}$$

Sentence similarity based on word order requires constructing the order vectors of the two sentences first. If the sentence S_a has words (w_{a1}, w_{a2}…w_{ai}) and sentence S_b has words (w_{b1},w_{b2},….w_{bj}) then word order vectors for S_a and S_b are:

$$L(S_a) = \{(w_{a1}, w_{a2}), (w_{a1}, w_{a3})\ldots(w_{a(i-1)}, w_{ai})\}$$
$$L(S_b) = \{(w_{b1}, w_{b2}), (w_{b1}, w_{b3})\ldots(w_{b(i-1)}, w_{bi})\}$$

Where (w_x, w_y) means w_x is before w_y in the sentence. The similarity between the sentences is then given as:

$$\text{Word order similarity}(S_a,S_b) = \frac{|\, L(S_a) \cap L(S_b) \,|}{|L(S_a) \cup L(S_b)|}$$

The training set is used to find the threshold in both the cases. Here the threshold is found with the training set. Hence if the score obtained is greater than this threshold value, the sentence pair in the test set is classified as a paraphrase, else it is considered a non paraphrase.

4.2. Similarity Matrix Technique

This is one of the most successful semantic techniques. In this method, the two sentences are represented as binary vectors (with element equal to 1 if a word is present in the sentence, else 0). While constructing binary vectors, only the keywords in the sentences are considered. Next the similarity matrix is constructed. Similarity matrix will contain the similarity scores between every pair of keywords from both the sentences. The similarity between the words is computed using the Lin WordNet similarity metric described in section 2.2. If the similarity matrix is W, and the sentence vectors are a and b, the score is given as:

$$\text{Similarity} = \frac{a.W.b^T}{|a|.|b|}$$

This method requires more preprocessing. The text should be tokenized first. String::Tokenizer module from CPAN is used for tokenizing the text. The WordNet similarity metric used here is the Lin similarity metric which allows only comparison of words that has the same POS tag. Hence after tokenization, tagging should be done. Lingua::TreeTagger module from CPAN is used for this purpose. Stopwords are also eliminated using Lingua::TreeTagger::Filter.

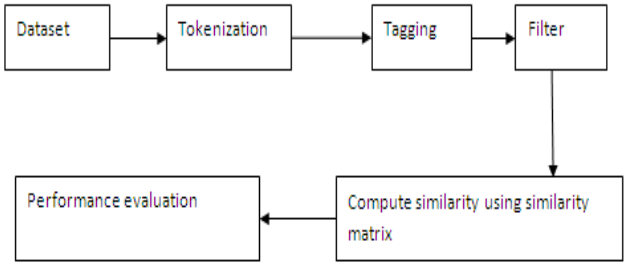

Figure 2: Similarity matrix schematic diagram

4.3. Performance Metrics

The performance metrics used are accuracy, precision, f-measure and recall. If a sentence pair is a paraphrase and the score indicates it is a paraphrase, then it is a true positive (TP). If a sentence pair is a non-paraphrase and the score also indicates it to be a non-paraphrase, then it is a true negative (TN). These are the cases of true classification. If a sentence pair is a paraphrase but the score indicates it to be a non-paraphrase, then it is a false negative (FN). If a sentence pair is a non-paraphrase but the score indicates it to be a paraphrase, then it is a false positive (FP).

Accuracy= (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall = TP / (TP + FN)
F-measure =
(2* precision * recall) / (precision + recall)

5. Results

The results of the experiments are shown in the table below:

Method	Accuray	Precision	Recall	Fmeasure
Jaccard	0.85	0.87	0.93	0.90
Dice	0.95	1	0.93	0.96
Word order	0.83	0.88	0.90	0.89
Semantic matrix	0.80	0.83	0.94	0.88

Table 3: Results of the experiment

As per the results, it is clear that the statistical measure, dice similarity, performs the best. Statistical techniques take all the words in the sentences into consideration. In semantic technique, the WordNet::Similarity methods can be used to compute the similarity between words only if either both the words are nouns or both the words are verbs because the adjectives and adverbs in WordNet are not organized into is-a hierarchy. Also, when the time taken is considered, semantic technique takes a lot more time for computation than statistical techniques because WordNet::Similarity methods takes a lot of time for computation.

6. Conclusion and future scope

The aim of this project was to compare statistical and semantic techniques for paraphrase identification. These were tested on the same dataset. Though there is a need for more computation in semantic techniques compared to the simple statistical measures, it is seen that their performances are comparable, with dice similarity measure performing the best.

7. References

[1] Samuel Fernando and Mark Stevenson, "Paraphrase Identification", University of Sheffield, August 2007, pages 6-26.

[2] Zhang, Sun, Wang and He, "Calculating Statistical Similarity between Sentences", Journal of Convergence of Information Technology, February 2011.

[3] Samuel Fernando and Mark Stevenson, "A semantic Similarity Approach to Paraphrase Identification", University of Sheffield, 2007.

[4] Mihalcea, Corley and Strapparava, "Corpus-based and Knowledge-based Measures of Text Semantic Similarity", Proc. American Association for Artificial Intelligence. 2006.

[5] Qiu, Kan and Chua, "Paraphrase Recognition via Dissimilarity Significance classification", Proc. Association for Computational Linguistics-Empirical Methods in Natural Language Processing, 2006.

[6] Zhang and Patrick, "Paraphrase Identification by Text Canonicalization", Proc. Australasian Language Technology Workshop, 2005.