Learning Paraphrases from WNS Corpora

João Cordeiro
CLT and Bioinformatics
University of Beira Interior
Covilhã, Portugal
Email: jpaulo@di.ubi.pt

Gaël Dias
CLT and Bioinformatics
University of Beira Interior
Covilhã, Portugal
Email: ddg@di.ubi.pt

Pavel Brazdil
LIACC
University of Porto
Porto, Portugal
Email: pbrazdil@liacc.up.pt

Abstract

Paraphrase detection can be seen as the task of aligning sentences that convey the same information but yet are written in different forms. Such resources are important to automatically learn text-to-text rewriting rules. In this paper, we present a new metric for unsupervised detection of paraphrases and apply it in the context of clustering of paraphrases. An exhaustive evaluation is conducted over a set of standard paraphrase corpora and real-world web news stories (WNS) corpora. The results are promising as they outperform state-of-the-art measures developed for similar tasks.

Introduction

Monolingual text-to-text generation is an emerging research area in Natural Language Processing. One reason for the interest in such generation systems is the possibility to automatically learn text-to-text generation models from aligned monolingual corpora (Jing & McKeown 2000; Knight & Marcu 2002; Shi 2002; Barzilay & Lee 2003; Structuring 2003; M. Le Nguyen & Ho 2004; Marsi & Krahmer 2005). Such text collections are usually called paraphrase corpora. In fact, text-to-text generation is a particularly promising research direction given that there are naturally occurring examples of comparable texts that convey the same information yet are written in different styles. Web news stories are an obvious example. So, presented with such texts, one can pair sentences that convey the same information, thereby building a training set of rewriting examples i.e. a paraphrase corpus. These pairs of sentences are called paraphrases and share almost the same meaning, but contain different lexical elements and possibly different syntactical structure. However, the unsupervised methodologies proposed so far (Barzilay & Lee 2003; W.B Dolan & Brockett) show a major drawback by extracting quasi-exact or even exact match pairs of sentences as they rely on classical string similarity measures such as the Edit Distance in the case of (W.B Dolan & Brockett) and word overlap for (Barzilay & Lee 2003). Such pairs are obviously useless. As a consequence, we first propose a new metric - named the *Sumo-Metric* - that presents a solution to these limitations and outperforms all state-of-the-art metrics both in the general case where exact and quasi-exact pairs do not occur and in the real-world case where exact and quasi-exact pairs occur like in web news stories. Second, (Barzilay & Lee 2003) show that clusters of paraphrases can lead to better learning of text-to-text rewriting rules compared to just pairs of paraphrases. For that purpose, they use the complete-link hierarchical algorithm but do not provide any evaluation. We fulfill this lack by proposing a comparison of three clustering algorithms and show that improved results can be obtained with the QT algorithm (L.J. Heyer & Yooseph).

Related Work

Three different approaches have been proposed for paraphrase detection: unsupervised methodologies based on lexical similarity (Barzilay & Lee 2003; W.B Dolan & Brockett), supervised methodologies based on context similarity measures (Bar 2003) and methodologies based on linguistic analysis of comparable corpora (V. Hatzivassiloglou & Eskin). (W.B Dolan & Brockett) endeavored a work to find and extract monolingual paraphrases from massive comparable news stories. They use the Edit Distance (also known as *Levenshtein Distance* (Levenshtein 1966)) and compare it with an heuristic derived from Press writing rules. The evaluation shows that the data produced by the Edit Distance is cleaner and more easily aligned than by using the heuristic. However, using word error alignment rate, results show that both techniques perform similarly. (Barzilay & Lee 2003) use the simple word n-gram ($n = 1, 2, 3, 4$) overlap measure in the context of paraphrase lattices learning. In particular, this string similarity measure is used to produce clusters of paraphrases using hierarchical complete-link clustering. More deepening techniques rely on context similarity measures such as (Bar 2003). They find sentence alignments in comparable corpora by considering sentence contexts (local alignment) after semantically aligning equivalent paragraphs. Although they show interesting results, this methodology relies on supervised learning techniques, which need huge quantities of training data that may be scarce and difficult to obtain. Others, such as (V. Hatzivassiloglou & Eskin), go further by exploring harvesting linguistic features combined with machine learning techniques to propose a new text similarity metric. Once again it is a supervised approach

and also heavily dependent on valuable linguistic resources which is usually not available for the vast majority of languages.

Metrics Overview

In the literature, we can find the *Levenshtein Distance* also known as the *Edit Distance* and the *Word N-Gram Overlap Family* of similarity measures. In this section, we review all existing metrics and propose a new n-gram overlap metric based on the Longest Common Prefix paradigm.

The Levenshtein Distance

The Levenshtein Distance (Levenshtein 1966) is a well-known metric that may be adapted for calculating *Sentence Edit Distance* upon words instead of characters (W.B Dolan & Brockett). Considering two strings, it computes the number of character/words insertions, deletions and substitutions that would be needed to transform one string into the opposite. A problem, when using the Edit Distance for the detection of paraphrases, is the possibility that there exist sentence pairs that are true paraphrases but are not identified as such. In fact, if the sentences show high lexical alternations or different syntactical structures, they are unlikely defined as similar.

The Word N-Gram Family

Two metrics are usually found in the literature: the word simple n-gram overlap and the BLEU metric. In order to be complete, we also propose a new metric based on the Longest Common Prefix paradigm.

Word Simple N-gram Overlap: For a given sentence pair, the metric counts how many 1-grams, 2-grams, 3-grams, ..., N-grams overlap. Usually N is chosen equal to 4 or less (Barzilay & Lee 2003). Let's name this counting function $Count_{match}$(n-gram). So, for a given $N \geqslant 1$, this normalized metric evaluates the similarity between sentences S_a and S_b, as given in Equation 1:

$$sim_o(S_a, S_b) = \frac{1}{N} * \sum_{n=1}^{N} \frac{Count_{match}(\text{n-gram})}{Count(\text{n-gram})} \quad (1)$$

where the function $Count$(n-gram) counts the maximum number of n-grams in the shorter sentence.

Exclusive LCP N-gram Overlap: In most work in Natural Language Processing, the longest a string is, the more meaningful it should be (G. Dias & Lopes). Based on this idea, we propose an extension of the word simple n-gram overlap metric. The difference between simple and exclusive n-gram overlap lays on the fact that the exclusive form counts prefix overlapping 1-grams, 2-grams, 3-grams, ..., N-grams, regarding the Longest Common Prefix (LCP) paradigm proposed by (Yamamoto & Church 2001). For example, if some maximum overlapping 4-gram is found then its 3-grams, 2-grams and 1-grams prefixes will not be counted. Only the 4-gram and its suffixes will be taken into account. This is based on the idea that the longer the match the more significant the match will be. Therefore smaller matches are discarded. If one wants to normalize the n-gram overlap then a particular difficulty rises due to the LCP n-gram considerations i.e. the maximum number of overlapping n-grams depends on the number of (n+1)-gram overlaps that exist. For that purpose, we introduce a normalized function expressed in Equation 2

$$sim_{exo}(S_a, S_b) = \max_{\text{n}} \left\{ \frac{Count_{match}(\text{n-gram})}{Count(\text{n-gram})} \right\} \quad (2)$$

where S_a and S_b are two sentences and the functions $Count_{match}$(n-gram) and $Count$(n-gram) are the same as above with the new matching strategy i.e. we first calculate $sim_{exo}(S_a, S_b)$ for 4-grams and then for the remaining 3-grams and so on and so forth, and then choose the maximum ratio.

The BLEU Metric: The BLEU metric was introduced by (K. Papineni 2001) for automatic evaluation of machine translation but can easily be adapted to calculate similarity between two sentences as it is based on the calculation of string overlaps between texts. The adapted formula is given below in Equation 3:

$$BLEU = \frac{1}{N} * exp[\sum_{n=1}^{N} log \sum_{\text{n-gram}} \frac{Count_{match}(\text{n-gram})}{Count(\text{n-gram})}] \quad (3)$$

The $Count_{match}$(n-gram) function counts the number of exclusive or no-exclusive n-grams co-occurring between the two sentences, and the function $Count$(n-gram) the maximum number of n-grams that exists in the shorter sentence[1].

The Sumo-Metric

Four main premises guided our research: (1) Achieve maximum automation in corpus construction - minimum or even no human intervention, with high reliability, (2) Penalize equal and almost equal sentences - they are not useful for our research needs, but frequent in real-world situations, (3) Consider pairs having a high degree of lexical reordering, and different syntactic structure and (4) Define a computationally fast and well founded metric. The basic idea of the *Sumo-Metric* lays on the notion of exclusive lexical links between a sentence pair, as shown in figure 1.

It is another form to think about 1-gram exclusive overlap. If a link is established between sentence S_a and sentence S_b, for the word **w**, then other occurrences of word **w** in sentence S_a will engage a new link to sentence S_b if there exists at least one more occurrence of **w** in S_b, besides the one which is already connected.

Definition

Let's introduce some notations. The number of links between the two sentences are defined as λ and the number of words in the longest and shortest sentence as x and y, respectively. To calculate the *Sumo-Metric* $S(.,.)$, we first evaluate the function $S(x, y, \lambda)$ as in Equation 4

[1] In our experiments, we will only show the results with no exclusive n-grams as results were worst with exclusive n-grams as we will show in the last section.

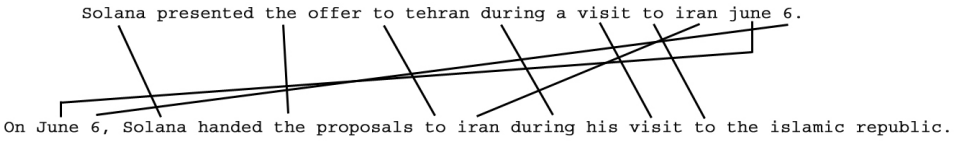

Figure 1: Links between a sentence pair.

$$S(x, y, \lambda) = \alpha \log_2(\frac{x}{\lambda}) + \beta \log_2(\frac{y}{\lambda}) \quad (4)$$

where $\alpha, \beta \in [0, 1]$ and $\alpha + \beta = 1$. After that, we compute the *Sumo-Metric* $S(.,.)$ as in Equation 5.

$$S(S_a, S_b) = \begin{cases} S(x, y, \lambda) & if \;\; S(x, y, \lambda) < 1.0 \\ e^{-k*S(x,y,\lambda)} & otherwise \end{cases} \quad (5)$$

With the α and β parameters, one may weight the value of the two main components involved in the calculation as in any linear interpolation. For example, to give more relevance to the component that depends on $\frac{\lambda}{y}$ (the shortest sentence), let β be superior to 0.5. In our experiments we equally weighted both components, i.e. $\alpha = \beta = 0.5$[2]. The effect of using the $log_2(.)$ function is to gradually penalize pairs that are very similar - remark that for equal pairs the result is exactly zero. The second branch of function 5 guarantees that the metric never returns values greater than 1.0. Theoretical work shows that this is the case when $x^\alpha y^\beta > 2\lambda$. For $\alpha = \beta = 0.5$, this occurs when $\sqrt{xy} > 2\lambda$ or $xy > 4\lambda^2$. As an example, let us consider the following two situations:

$\langle x, y, \lambda \rangle = \langle 15, 6, 5 \rangle \;\Rightarrow\; S(x, y, \lambda) = 0.924$
$\langle x, y, \lambda \rangle = \langle 30, 6, 5 \rangle \;\Rightarrow\; S(x, y, \lambda) = 1.424$

The first example is clearly a relevant situation. However, the second example is over-evaluated in terms of similarity. As a consequence, $e^{-k*S(x,y,\lambda)}$ is a penalizing factor, where the constant k is a tuning parameter[3] that may scale this factor more or less. In particular, we can see its effect as follows.

$\langle x, y, \lambda \rangle = \langle 30, 6, 5 \rangle \;\Rightarrow\; e^{-k*S(x,y,\lambda)} = 0.014$

In fact, when sentences tend to be very asymmetric, in number of words, the computation of $S(x, y, \lambda)$ gives values greater than 1.0, despite the number of links that exist. So, the higher $S(x, y, \lambda)$ is beyond 1.0, the more unlikely the pair will be classified as positive with respect to $S(.,.)$.

Complexity

The *Sumo-Metric* is computed in $\Theta(x * y)$ time, in the worst case - when the sentences are completely different, i.e. there is no link among them. In that case, we compute $x * y$ comparisons i.e. each word in the longest sentence is compared with each word from the shortest one. In the best situation the computation will take only $\Theta(y)$ time. This is the case when the shortest sentence is a prefix of the longest one. In terms of comparison, all metrics show time complexity $\Theta(x * y)$ except the *exclusive LCP n-gram overlap* metric that evidences time complexity $\Theta((x + y)log(x + y))$.

[2]Best results were obtained in this case for the used corpora set.
[3]$k = 3$ was used in our experiments.

The Corpora Set

Two standard corpora were used for comparative tests between metrics: The Microsoft Research Paraphrase Corpus (W.B Dolan & Brockett), labeled $\{MSRPC\}$ and a corpus supplied by Daniel Marcu, labeled $\{KMC\}$, that has been used for research in the field of Sentence Compression (Knight & Marcu 2002; M. Le Nguyen & Ho 2004). By adapting these corpora we created three new corpora to serve as a benchmark for our specific purpose. We also automatically created a corpus of web news stories using Google News to test the metrics in real-world conditions. One major limitation with the $\{KMC\}$ corpus is that it only contains positive pairs. Therefore it should not be taken as such to perform any evaluation. Indeed, we need an equal number of negative pairs of sentences to produce a fair evaluation for any paraphrase detection metric. Although the $\{MSRPC\}$ corpus already contains negative pairs, they are only 1901 against 3900 positive examples. To perform an equitable evaluation, we first expanded both corpora by adding negative sentence pairs selected from web news stories so that they have the same number of positive and negative examples and also created a new corpus based on the combination of the $\{MSRPC\}$ and the $\{KMC\}$.

The $\{MSRPC \cup X^-_{1999}\}$ this new derived corpus contains the original $\{MSRPC\}$ collection of 5801 pairs (3900 positives and 1901 negatives) plus 1999 extra negative sentences (symbolized by X^-_{1999}), selected from web news stories.

The $\{KMC \cup X^-_{1087}\}$ from the $\{KMC\}$, we derived a new corpus that contains its 1087 positive pairs plus a set of negative pairs, in equal number, selected from web news stories and labeled it in the same manner as the previous corpus.

The $\{MSRPC^+ \cup KMC \cup X^-_{4987}\}$ from the $\{MSRPC\}$ and the $\{KMC\}$ we built a bigger corpus gathering the positive $\{MSRPC\}$ part i.e. 3900 positive examples, and the 1087 positive pairs of sentences from the $\{KMC\}$ corpus,

giving a total of 4987 positive pairs. To balance these positive pairs we added an equal number of negative pairs, selected in a same manner as described previously. In this corpus, we intentionally ignored the $\{MSRPC\}$ negative pairs as many pairs that are labeled negative, following the guidelines expressed in (W.B Dolan & Brockett), are in fact useful paraphrases.

The $\{WNS\}$ in order to perform an exhaustive evaluation of paraphrase metrics, we automatically built a real-world corpus of web news stories that likely contains paraphrases. It was compiled on October 2006 from Google News for three distinct news stories and contains 166 stories.

Results

In a first step, we present a comparative study between already existing metrics and new adapted ones over the first three corpora mentioned in the previous section. In a second step, once the best metric has been found, we propose a comparative evaluation of three clustering algorithms to determine clusters of paraphrases.

How to Classify a Paraphrase?

Before presenting the results, it is is necessary to talk about a classical problem in classification - *thresholds*. Thresholds are parameters that unease the process of evaluation. Ideally, the best parameter should be determined for each metric. However, this is not always the case and wrong evaluations are sometimes proposed in the literature. In our evaluation, we do not pre-define any threshold for any metric. Instead, for each metric, we automatically compute the best threshold. This computation is a classical problem of function maximization or optimization. In particular, we use the bisection strategy (Polak 1971) as it computes fast, and well approximates the global maximum of our functions. As a result, we are optimizing the value of the threshold for each metric in the same way and do not introduce any subjectivity in the choice of the parameters. In Table 1, we present the obtained thresholds for the five compared metrics using a *10-fold cross validation* scheme. In the remainder of this paper, we will rename $\{MSRPC \cup X^{-}_{1999}\}$ as **A**, $\{KMC \cup X^{-}_{1087}\}$ as **B** and $\{MSRPC^{+} \cup KMC \cup X^{-}_{4987}\}$ as **C** in order to ease the reading.

Table 1: Thresholds mean and standard deviation

thresholds	A	B	C
edit	17.222 ± 0.111	20.167 ± 1.375	17.313 ± 0.000
sim_o	0.203 ± 0.007	0.261 ± 0.003	0.254 ± 0.000
sim_{exo}	0.501 ± 0.000	0.725 ± 0.013	0.501 ± 0.000
bleu	0.502 ± 0.003	0.501 ± 0.000	0.501 ± 0.000
sumo	0.077 ± 0.004	0.005 ± 0.001	0.007 ± 0.000

The results show that the bisection strategy performs well for our task as the standard deviation for each measure and corpus is almost negligible.

First Experiments

In order to evaluate the results of each metric over each corpus, we computed both the F-Measure (Rijsbergen 1979) and the Accuracy (Mitchell 1997). In particular, the results were calculated by averaging the 10 F-Measure and Accuracy values obtained from the *10-fold cross validation* test executed over the data. For every fold, the best threshold was found on the $\frac{9}{10}$ training data and then used on the $\frac{1}{10}$ test block to measure the correspondent F-Measure and Accuracy. The overall results are presented in Table 2.

Table 2: $F-Measure$ and $Accuracy$ results.

%	A-F_β	A-$Acc.$	B-F_β	B-$Acc.$	C-F_β	C-$Acc.$
edit	74.41	67.67	70.65	68.02	80.98	79.02
sim_o	78.06	73.15	94.66	94.47	91.92	91.79
sim_{exo}	77.27	72.37	90.87	90.23	87.19	86.00
bleu	70.77	66.17	82.39	78.89	76.79	74.13
sumo	**80.92**	**78.19**	**98.45**	**98.43**	**98.53**	**98.53**

The results evidenced in Table 2 show that the *Sumo-Metric* outperforms all state-of-the-art metrics over all corpora. For instance, on the biggest corpus (C), the *Sumo-Metric* correctly classified, on average, 98.53% of all the 9974 sentence pairs, either positives or negatives. It shows systematically better F-Measure and Accuracy measures over all other metrics showing an improvement of (1) at least 2.86% in terms of F-Measure and 3.96% in terms of Accuracy and (2) at most 6.61% in terms of F-Measure and 6.74% in terms of Accuracy compared to the second best metric which is also systematically the sim_o similarity measure. Another interesting result is the fact that three metrics behave the same way over all corpora. While the *Sumo-Metric* is always the best measure, the simple word n-gram overlap (sim_o) and the exclusive LCP n-gram overlap (sim_{exo}) metrics always get second and third places, respectively. So, the hypothesis proposed by (G. Dias & Lopes) does not seem to stand for paraphrase detection. Indeed, counting many times the same links gives more weight to paraphrase candidates than just counting only once the relevant "meaningful" links. On the other hand, the BLEU metric and the Edit Distance obtain the worst results over all corpora. However, their behavior is quite different as it is also evidenced in the next subsection. The BLEU measure only outperforms the Edit Distance for the **B** corpus. Here, it is important to point at two facts that lead to this situation: (1) the **A** corpus was computed based on the Edit Distance and contains a majority of positive examples that are near string matches, and (2) the **C** corpus is unbalanced as it contains more positive examples from the **A** corpus than from the **B** corpus. As a consequence, the Edit Distance gives better results than the BLEU metric for **A** and **C** corpora as they contain more near string matches as positive examples. Unlikely, the **B** corpus contains humanly created paraphrases that generally show higher lexical and syntactical diversity. In this case, the BLEU measure shows better behavior than the Edit Distance.

The Influence of Random Negative Pairs

One may criticize that the superior performance obtained by the *Sumo-Metric* depends exclusively on the set of equal or quasi-equal pairs which are present in the corpora. However, this is not the case. Indeed, to acknowledge this situation, we performed another experiment with a corpus similar to the **C** corpus (the biggest one) but without any quasi-equal or equal pair. Let's call it the **C'** corpus. The performance obtained over the **C'** is illustrated in Table 3 and clearly shows that the *Sumo-Metric* outperforms all other state-of-the-art metrics in all evaluation situations, even when equal or quasi-equal pairs are not present in the corpora.

Table 3: Corpus without quasi-equal or equal pairs

Accuracy %	edit	sim_o	sim_{exo}	bleu	sumo
C'	84.31	96.36	90.19	77.98	99.58

In this case, we only show the Accuracy measure as the F-measure evidences similar results. This give us at least 99% statistical confidence[4] (1% significance) that $Accuracy_{sumo} > Accuracy_{simx}$, where $simx \in \{edit, sim_o, sim_{exo}, bleu\}$ (any other tested metric).

Second Experiments

While previous similarity measures are tailored to extract pairs of sentences, clustering algorithms should describe groups of sentences with similar structures. There are two main reasons to apply clustering for paraphrase detection. On one hand, as (Barzilay & Lee 2003) evidence, clusters of paraphrases can lead to better learning of text-to-text rewriting rules compared to just pairs of paraphrases. On the other hand, clustering algorithms may lead to better performance than stand-alone similarity measures as they may take advantage of the different structures of sentences in the cluster to detect a new similar sentence.

While (Barzilay & Lee 2003) only mention the usage of the complete-link hierarchical clustering algorithm and do not show any results, we present the results for three algorithms that do not need the pre-definition of the expected number of clusters: the complete-link hierarchical clustering algorithm (Day & Edelsbrunner 1984), the single-link hierarchical clustering algorithm (Day & Edelsbrunner 1984) and the QT algorithm (L.J. Heyer & Yooseph)[5]. We implemented a QT algorithm and for hierarchical clustering used the LingPipe package[6], a suite of Java libraries for the linguistic analysis of human language. So, each algorithm was tested over the same similarity matrix based on the *Sumo-Metric* over the $\{WNS\}$ corpus i.e. over a real-world situation. As the $\{WNS\}$ corpus was automatically created, a manual evaluation was needed to assess the results. So, we first statistically defined a subset of n elements of all the clusters that were found by each algorithm for a confidence level of 90% with $\pm$ 0.075 precision error following simple random sampling (Bhattacharrya & Johnson 1977) as explained in Equation 6 [7].

$$n = p^*(1-p)^* \left[\frac{z_{\alpha/2}}{d}\right]^2 \quad (6)$$

The evaluation was individually made by two researchers and results were then cross-validated to decrease subjectivity as much as possible. Both researchers were given the following guidelines to define correct clusters of paraphrases: (1) two sentences are paraphrases if their semantic contents are similar or if the content of one sentence can be entailed by the content of the other one and (2) a cluster of paraphrases is a correct cluster if all combinations of two sentences are paraphrases[8]. The results are presented in Table 4 where S-HAC and C-HAC respectively stand for Simple and Complete-link Hierarchical clustering algorithms, QT for the QT algorithm and sumo for the stand-alone *Sumo-Metric*.

Table 4: Precision of clustering algorithms

Precision %	sumo	S-HAC	C-HAC	QT
$\{WNS\}$	61.79	57.72%	56.91%	64.03

The *Sumo-Metric* plays the role of the baseline that clustering algorithms should overpass. However, the results show that only the QT algorithm provides better results. Indeed, both the simple and the complete-link hierarchical clustering algorithms show worst results than the baseline[9]. As (Barzilay & Lee 2003) mention, clustering may lead to better results than stand-alone similarity measures. However, unlike (Barzilay & Lee 2003), the Hierarchical clustering algorithms do not seem to be the right choice for paraphrase clustering.

Recall of Clustering Results

Although, we do not know the correct number of clusters of paraphrases in the $\{WNS\}$ corpus, we propose to evaluate the recall of each clustering algorithm by their capacity to rebuild an adapted subset of the reference corpus $\{MSRPC\}$, labeled $\{MSRPC_{1000} \cup LIT_{2000}\}$ [10]. The results are presented in Table 5 where $\widehat{r}$ is the recall estimator and the "Correct" column contains the number of original correct paraphrase pairs reconstructed as a cluster.

These results show that the three clustering algorithms perform equally and achieve good recall.

[4] By making a proportion statistical test for the accuracies: $H_0 : p_1 = p_2$ against $H_1 : p_1 > p_2$.

[5] We are making some experiments with the average-link hierarchical clustering algorithm (Day & Edelsbrunner 1984) and the EM algorithm (A. Dempster & Rubin), but at the moment of the submission results were not ready yet.

[6] http://www.alias-i.com/lingpipe/

[7] In these experiments, $p^* = 0.64$, $d = 0.075$ and $z_{\alpha/2} = 1.65$ is the usual upper $\alpha/2$ of the standard normal distribution.

[8] We point at that quasi-exact and exact matches of sentences are not considered correct paraphrases.

[9] For all clustering algorithms, we chose the same normalized similarity factor of 0.8 (or distance 0.2) for the definition of the clusters.

[10] This corpus contains 1000 positive pairs from the $\{MSRPC\}$ and 2000 sentences picked from classical literature books, both in random fashion.

Table 5: Recall of clustering algorithms

Clust. Algor.	QT	S-HAC	C-HAC
Correct	836	826	841
$\widehat{r}$	83.60%	82.69%	84.10%

Conclusion and Future Work

In this paper, we proposed a new metric, the *Sumo-Metric*, for finding paraphrases. But, we also performed a comparative study between already existing metrics and new adapted ones and proposed a new benchmark of paraphrase test corpora. In particular, we tested the performance of 5 metrics over 4 corpora. One main and general conclusion is that the *Sumo-Metric* performed better than any other measure over all corpora either in terms of F-Measure and Accuracy. Moreover, the Word Simple N-gram Overlap and the Exclusive LCP N-gram Overlap are systematically second and third in the ranking over all corpora, thus negating (G. Dias & Lopes)'s assumption for the task of paraphrase detection. Finally, the *Levenshtein Distance* (Levenshtein 1966) performs poorly over corpora with high lexical and syntactical diversity unlike the BLEU measure. However, when paraphrases are almost string matches, the Edit Distance outperforms the BLEU measure. Nevertheless, in all cases, we must point at that the Edit Distance and the BLEU measure are always classified fourth or fifth in the ranking. In a second part of the paper, we showed that clustering of paraphrases can lead to improved results when compared to stand-alone similarity measures and provide with clusters of paraphrases that can lead to better learning of text-to-text rewriting rules compared to just pairs of paraphrases. However, this situation was only evidenced the QT clustering algorithm. Indeed, both the Simple and the Complete-link hierarchical clustering algorithms show worst results than the baseline, the simple paraphrase pair detection with the *Sumo-Metric*. As future work, we plan to insert the notion of tf.idf (Salton & Buckley 1988) in our metric, as we believe that word links between sentences should have distinct weights. Another improvement may be the integration of the notion of content character n-grams as in (G. Dias & Lopes). This would lead to use automatically acquired "meaningful" character sequences instead of words thus avoiding language-dependent stemming and allowing better counts between sentences.

Acknowledgment

We would like to thank Daniel Marcu for providing us with his corpus of paraphrases.

References

A. Dempster, N. L., and Rubin, D. Maximum likelihood from incomplete data via the em algorithm.

2003. *Sentence Alignment for Monolingual Comparable Corpora.*, Sapporo, Japan.

Barzilay, R., and Lee, L. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *In Proceedings of HLT-NAACL.*

Bhattacharrya, G., and Johnson, R. 1977. *Statistical Concepts and Methods*.

Day, W. H., and Edelsbrunner, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification.* 1:1–24.

G. Dias, S. G., and Lopes, J. Extraction automatique d'associations textuelles à partir de corpora non traités.

Jing, H., and McKeown, K. 2000. Cut and paste based text summarization. *In Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics* 178–185.

K. Papineni, S. Roukos, T. W. W.-J. Z. 2001. Bleu: a method for automatic evaluation of machine translation. *IBM Research Report RC22176*.

Knight, K., and Marcu, D. 2002. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91–107.

Levenshtein, V. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physice-Doklady* 10:707–710.

L.J. Heyer, S. K., and Yooseph, S. Exploring expression data: Identification and analysis of coexpressed genes.

M. Le Nguyen, S. Horiguchi, A. S., and Ho, B. T. 2004. Example-based sentence reduction using the hidden markov model. *ACM Transactions on Asian Language Information Processing (TALIP)* 3(2):146–158.

Marsi, E., and Krahmer, E. 2005. Explorations in sentence fusion. *In Proceedings of the 10th European Workshop on Natural Language Generation*.

Mitchell, T. 1997. *Machine Learning*.

Polak, E. 1971. Computational methods in optimization. *New York Academic Press*.

Rijsbergen, C. J. V. 1979.

Salton, G., and Buckley, C. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management* 24(5):513–523.

2002. *Automatic Paraphrase Acquisition from News Articles*, Sao Diego, USA.

Structuring, M. L. P. T. 2003. Experiments with sentence ordering. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).*

V. Hatzivassiloglou, J. K., and Eskin, E. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning.

W.B Dolan, C. Q., and Brockett, C. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.

Yamamoto, M., and Church, K. 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics* 27(1):1–30.