Biology Based Alignments of Paraphrases for Sentence Compression

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

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

Guillaume Cleuziou
LIFO - University of Orléans
Orléans, France
guillaume.cleuziou@univ-orleans.fr

Abstract

[1] In this paper, we present a study for extracting and aligning paraphrases in the context of Sentence Compression. First, we justify the application of a new measure for the automatic extraction of paraphrase corpora. Second, we discuss the work done by (Barzilay & Lee, 2003) who use clustering of paraphrases to induce rewriting rules. We will see, through classical visualization methodologies (Kruskal & Wish, 1977) and exhaustive experiments, that clustering may not be the best approach for automatic pattern identification. Finally, we will provide some results of different biology based methodologies for pairwise paraphrase alignment.

1 Introduction

Sentence Compression can be seen as the removal of redundant words or phrases from an input sentence by creating a new sentence in which the gist of the original meaning of the sentence remains unchanged. Sentence Compression takes an important place for Natural Language Processing (NLP) tasks where specific constraints must be satisfied, such as length in summarization (Barzilay & Lee, 2002; Knight & Marcu, 2002; Shinyama et al., 2002; Barzilay & Lee, 2003; Le Nguyen & Ho, 2004; Unno et al., 2006), style in text simplification (Marsi & Krahmer, 2005) or sentence simplification for subtitling (Daelemans et al., 2004).

[1]Project partially funded by Portuguese FCT (Reference: POSC/PLP/57438/2004)

Generally, Sentence Compression involves performing the following three steps: (1) Extraction of paraphrases from comparable corpora, (2) Alignment of paraphrases and (3) Induction of rewriting rules. Obviously, each of these steps can be performed in many different ways going from totally unsupervised to totally supervised.

In this paper, we will focus on the first two steps. In particular, we will first justify the application of a new measure for the automatic extraction of paraphrase corpora. Second, we will discuss the work done by (Barzilay & Lee, 2003) who use clustering of paraphrases to induce rewriting rules. We will see, through classical visualization methodologies (Kruskal & Wish, 1977) and exhaustive experiments, that clustering may not be the best approach for automatic pattern identification. Finally, we will provide some results of different biology based methodologies for pairwise paraphrase alignment.

2 Related Work

Two different approaches have been proposed for Sentence Compression: purely statistical methodologies (Barzilay & Lee, 2003; Le Nguyen & Ho, 2004) and hybrid linguistic/statistic methodologies (Knight & Marcu, 2002; Shinyama et al., 2002; Daelemans et al., 2004; Marsi & Krahmer, 2005; Unno et al., 2006).

As our work is based on the first paradigm, we will focus on the works proposed by (Barzilay & Lee, 2003) and (Le Nguyen & Ho, 2004).

(Barzilay & Lee, 2003) present a knowledge-lean algorithm that uses multiple-sequence alignment to

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 177–184,
Prague, June 2007.

learn generate sentence-level paraphrases essentially from unannotated corpus data alone. In contrast to (Barzilay & Lee, 2002), they need neither parallel data nor explicit information about sentence semantics. Rather, they use two comparable corpora. Their approach has three main steps. First, working on each of the comparable corpora separately, they compute lattices compact graph-based representations to find commonalities within groups of structurally similar sentences. Next, they identify pairs of lattices from the two different corpora that are paraphrases of each other. Finally, given an input sentence to be paraphrased, they match it to a lattice and use a paraphrase from the matched lattices mate to generate an output sentence.

(Le Nguyen & Ho, 2004) propose a new sentence-reduction algorithm that do not use syntactic parsing for the input sentence. The algorithm is an extension of the template-translation algorithm (one of example-based machine-translation methods) via innovative employment of the Hidden Markov model, which uses the set of template rules learned from examples.

In particular, (Le Nguyen & Ho, 2004) do not propose any methodology to automatically extract paraphrases. Instead, they collect a corpus by performing the decomposition program using news and their summaries. After correcting them manually, they obtain more than 1,500 pairs of long and reduced sentences. Comparatively, (Barzilay & Lee, 2003) propose to use the N-gram Overlap metric to capture similarities between sentences and automatically create paraphrase corpora. However, this choice is arbitrary and mainly leads to the extraction of quasi-exact or exact matching pairs. For that purpose, we introduce a new metric, the *Sumo-Metric*.

Unlike (Le Nguyen & Ho, 2004), one interesting idea proposed by (Barzilay & Lee, 2003) is to cluster similar pairs of paraphrases to apply multiple-sequence alignment. However, once again, this choice is not justified and we will see by classical visualization methodologies (Kruskal & Wish, 1977) and exhaustive experiments by applying different clustering algorithms, that clustering may not be the best approach for automatic pattern identification. As a consequence, we will study global and local biology based sequence alignments compared to multi-sequence alignment that may lead to better results for the induction of rewriting rules.

3 Paraphrase Corpus Construction

Paraphrase corpora are golden resources for learning monolingual text-to-text rewritten patterns. However, such corpora are expensive to construct manually and will always be an imperfect and biased representation of the language paraphrase phenomena. Therefore, reliable automatic methodologies able to extract paraphrases from text and subsequently corpus construction are crucial, enabling better pattern identification. 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 but are written in different styles. Web news stories are an obvious example. Thus, 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.

3.1 Paraphrase Identification

A few unsupervised metrics have been applied to automatic paraphrase identification and extraction (Barzilay & Lee, 2003; Dolan & Brockett, 2004). However, these unsupervised methodologies show a major drawback by extracting quasi-exact[2] or even exact match pairs of sentences as they rely on classical string similarity measures such as the *Edit Distance* in the case of (Dolan & Brockett, 2004) and *word N-gram overlap* for (Barzilay & Lee, 2003). Such pairs are clearly useless.

More recently, (Anonymous, 2007) proposed a new metric, the *Sumo-Metric* specially designed for asymmetrical entailed pairs identification, and proved better performance over previous established metrics, even in the specific case when tested with the *Microsoft Paraphrase Research Corpus* (Dolan & Brockett, 2004). For a given sentence pair, having each sentence x and y words, and with λ exclusive links between the sentences, the *Sumo-Metric* is defined in Equation 1 and 2.

[2]Almost equal strings, for example: *Bush said America is addicted to oil.* and *Mr. Bush said America is addicted to oil.*

$$S(S_a, S_b) = \begin{cases} S(x,y,\lambda) & if \;\; S(x,y,\lambda) < 1.0 \\ 0 & if \;\; \lambda = 0 \\ e^{-k*S(x,y,\lambda)} & otherwise \end{cases} \tag{1}$$

where

$$S(x,y,\lambda) = \alpha \log_2(\frac{x}{\lambda}) + \beta \log_2(\frac{y}{\lambda}) \tag{2}$$

with $\alpha, \beta \in [0,1]$ and $\alpha + \beta = 1$.

(Anonymous, 2007) show that the *Sumo-Metric* outperforms all state-of-the-art metrics over all tested corpora. In particular, 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 word N-gram overlap similarity measure used by (Barzilay & Lee, 2003).

3.2 Clustering

Literature shows that there are two main reasons to apply clustering for paraphrase extraction. 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.

However, as (Barzilay & Lee, 2003) do not propose any evaluation of which clustering algorithm should be used, we experiment a set of clustering algorithms and present the comparative results. Contrarily to what expected, we will see that clustering is not a worthy effort.

Instead of extracting only sentence pairs from corpora[3], one may consider the extraction of paraphrase sentence clusters. There are many well-known clustering algorithms, which may be applied to a corpus sentence set $S = \{s_1, ..., s_n\}$. Clustering implies the definition of a similarity or (distance) matrix $A_{n \times n}$, where each each element a_{ij} is the similarity (distance) between sentences s_i and s_j.

[3]A pair may be seen as a cluster with only two elements.

3.2.1 Experimental Results

We experimented four clustering algorithms on a corpus of *web news stories* and then three human judges manually cross-classified a random sample of the generated clusters. They were asked to classify a cluster as a ”wrong cluster” if it contained at least two sentences without any entailment relation between them. Results are shown in the next table 1.

Table 1: Precision of clustering algorithms

BASE	S-HAC	C-HAC	QT	EM
0.618	0.577	0.569	0.640	0.489

The ”BASE” column is the baseline, where the *Sumo-Metric* was applied rather than clustering. Columns ”S-HAC” and ”C-HAC” express the results for *Single-link* and *Complete-link Hierarchical Agglomerative Clustering* (Jain et al., 1999). The ”QT” column shows the *Quality Threshold* algorithm (Heyer et al., 1999) and the last column ”EM” is the *Expectation Maximization* clustering algorithm (Hogg et al., 2005).

One main conclusion, from table 1 is that clustering tends to achieve worst results than simple paraphrase pair extraction. Only the QT achieves better results, but if we take the average of the four clustering algorithms it is equal to 0.568, smaller than the 0.618 baseline. Moreover, these results with the QT algorithm were applied with a very restrictive value for cluster attribution as it is shown in table 2 with an average of almost two sentences per cluster.

Table 2: Figures about clustering algorithms

Algorithm	# Sentences/# Clusters
S-HAC	6,23
C-HAC	2,17
QT	2,32
EM	4,16

In fact, table 2 shows that most of the clusters have less than 6 sentences which leads to question the results presented by (Barzilay & Lee, 2003) who only keep the clusters that contain more than 10 sentences. In fact, the first conclusion is that the number of experimented clusters is very low, and more important, all clusters with more than 10 sentences showed to be of very bad quality.

The next subsection will reinforce the sight that clustering is a worthless effort for automatic paraphrase corpora construction.

3.2.2 Visualization

In this subsection, we propose a visual analysis of the different similarity measures tested previously: the Edit Distance (Levenshtein, 1966), the BLEU metric (Papineni et al., 2001), the word N-gram overlap and the *Sumo-Metric*. The goal of this study is mainly to give the reader a visual interpretation about the organization each measure induces on the data.

To perform this study, we use a Multidimensional Scaling (MDS) process which is a traditional data analysis technique. MDS (Kruskal & Wish, 1977) allows to display the structure of distance-like data into an Euclidean space.

Since the only available information is a similarity in our case, we transform similarity values into distance values as in Equation 3.

$$d_{ij} = (s_{ii} - 2s_{ij} + s_{jj})^{1/2} \qquad (3)$$

This transformation enables to obtain a (pseudo) distance measure satisfying properties like minimality, identity and symmetry. On a theoretical point of view, the measure we obtain is a pseudo-distance only, since triangular inequality is not necessary satisfied. In practice, the projection space we build with the MDS from such a pseudo-distance is sufficient to have an idea about whether data are organized into classes.

We perform the MDS process on 500 sentences[4] randomly selected from the Microsoft Research Paraphrase Corpus. In particular, the projection over the three first eigenvectors (or proper vectors) provides the best visualization where data are clearly organized into several classes (at least two classes). The obtained visualizations (Figure 1) show distinctly that no particular data organization can be drawn from the used similarity measures. Indeed, we observe only one central class with some "satellite" data randomly placed around the class.

The last observation allows us to anticipate on the results we could obtain with a clustering step. First, clustering seems not to be a natural way to manage such data. Then, according to the clustering method used, several types of clusters can be expected: very small clusters which contain "satellite" data (pretty relevant) or large clusters with part of the main central class (pretty irrelevant). These results confirm the observed figures in the previous subsection and reinforce the sight that clustering is a worthless effort for automatic paraphrase corpora construction, contrarily to what (Barzilay & Lee, 2003) suggest.

[4] The limitation to 500 data is due to computation costs since MDS requires the diagonalization of the square similarity or distance matrix.

4 Biology Based Alignments

Sequence alignments have been extensively explored in bioinformatics since the beginning of the *Human Genome Project*. In general, one wants to align two sequences of symbols (genes in Biology) to find structural similarities, differences or transformations between them.

In NLP, alignment is relevant in sub-domains like Text Generation (Barzilay & Lee, 2002). In our work, we employ alignment methods for aligning words between two sentences, which are paraphrases. The words are the *base blocks* of our sequences (sentences).

There are two main classes of pairwise alignments: the global and local classes. In the first one, the algorithms try to fully align both sequences, admitting gap insertions at a certain cost, while in the local methods the goal is to find pairwise sub-alignments. How suitable each algorithm may be applied to a certain problem is discussed in the next two subsections.

4.1 Global Alignment

The well established and widely used Needleman-Wunsch algorithm for pairwise global sequence alignment, uses dynamic programming to find the best possible alignment between two sequences. It is an optimal algorithm. However, it reveals space and time inefficiency as sequence length increases, since an $m * n$ matrix must be maintained and processed during computations. This is the case with DNA sequence alignments, composed by many thousands of nucleotides. Therefore, a huge optimization effort were engaged and new algorithms appeared like *k-tuple*, not guaranteeing to find optimal alignments but able to tackle the complexity problem.

In our alignment tasks, we do not have these complexity obstacles, because in our corpora the mean length of a sentence is equal to 20.9 words, which is considerably smaller than in a DNA sequence. Therefore an implementation of the Needleman-Wunsch algorithm has been used to generate optimal global alignments.

The figure 2 exemplifies a global word alignment on a paraphrase pair.

4.2 Local Alignment

The Smith-Waterman (SW) algorithm is similar to the Needleman Wunsch (NW) one, since dynamic programming is also followed hence denoting the similar complexity issues, to which our alignment task is immune. The main difference is that SW seeks optimal sub-alignments instead of a global alignment and, as described in the literature, it is well tailored for pairs with considerable differences[5], in length and type. In table 3 we exemplify this by showing two character sequences[6] where one may clearly see that *SW* is preferable:

N	Char. Sequences	Alignments
1	ABBAXYTRVRVTTRVTR FWHWWHGWGFXYTVWGF	XYTRV XYT-V
2	ABCDXYDRQR DQZZSTABZCD	AB-CD ABZCD

Table 3: Preferable local alignment cases.

Remark that in the second pair, only the maximal local sub-alignment is shown. However, there exists another sub-alignment: (DRQ, D-Q). This means that local alignment may be tuned to generate not only the maximum sub-alignment but a set of sub-alignments that satisfy some criterium, like having alignment value greater than some minimum threshold. In fact, this is useful in our word alignment problem and were experimented by adapting the Smith Waterman algorithm.

4.3 Dynamic Alignment

According to the previous two subsections, where two alignment strategies were presented, a natural question rises: which alignment algorithm to use for our problem of inter-sentence word alignment? Initially, we thought to use only the global alignment *Needleman Wunsch* algorithm, since a complete inter-sentence word alignment is obtained. However, we noticed that this strategy is unappropriate for certain pairs, specially when there are syntactical alternations, like in the next example:

> ***During his magnificent speech***, *the president remarkably praised IBM research.*
>
> *The president praised IBM research,* ***during his speech***.

If a global alignment is applied for such a pair, then weird alignments will be generated, like the one that is shown in the next representation (we use character sequences for space convenience and try to preserve the word first letter, from the previous example):

```
D H M S T P R Q I S _ _ _
_ _ _ _ T P _ Q I S D H S
```

Here it would be more adequate to apply local alignment and extract all relevant sub-alignments. In this case, two sub-alignments would be generated:

```
|D H M S|     |T P R P I R|
|D H _ S|     |T P _ P I R|
```

Therefore, for inter-paraphrase word alignments, we propose a dynamic algorithm which chooses the best alignment to perform: global or local. To compute this pre-scan, we regard the notion of link-crossing between sequences as illustrated in the figure 3, where the 4 crossings are signalized with the small squares.

It is easily verifiable that the maximum number of crossings, among two sequences with n exclusive links in between is equal to $\theta = \frac{1}{2} * n * (n-1)$. We suggest that if a fraction of these crossings holds, for example $0.4 * \theta$ or $0.5 * \theta$, then a local alignment should be used. Remark that the more this fraction tends to 1.0 the more unlikely it is to use global alignment.

Crossings may be calculated by taking index pairs $\langle x_i, y_i \rangle$ to represent links between sequences, where x_i and y_i are respectively the first and second sequence indexes, for instance in figure 3 the "U" link has pair $\langle 5, 1 \rangle$. It is easily verifiable that two links $\langle x_i, y_i \rangle$ and $\langle x_j, y_j \rangle$ have a crossing point if: $(x_i - x_j) * (y_i - y_j) < 0$.

4.4 Alignment with Similarity Matrix

In bioinformatics, DNA sequence alignment algorithms are usually guided by a scoring function, related to the field of expertise, that defines what is

[5] With sufficient similar sequences there is no difference between NW and SW.

[6] As in DNA subsequences and is same for word sequences.

the mutation probability between nucleotides. These scoring functions are defined by PAM[7] or BLOSUM[8] matrices and encode evolutionary approximations regarding the rates and probabilities of amino acid mutations. Different matrices might produce different alignments.

Subsequently, this motivated the idea of modeling word mutation. It seems intuitive to allow such a word mutation, considering the possible relationships that exit between words: lexical, syntactical or semantic. For example, it seems evident that between *spirit* and *spiritual* there exists a stronger relation (higher mutation probability) than between *spiritual* and *hamburger*.

A natural possibility to choose a word mutation representation function is the *Edit-distance* (Levenshtein, 1966) (edist(.,.)) as a negative reward for word alignment. For a given word pair $\langle w_i, w_j \rangle$, the greater the *Edit-distance* value, the more unlikely the word w_i will be aligned with word w_j. However, after some early experiments with this function, it revealed to lead to some problems by enabling alignments between very different words, like $\langle total, israel \rangle$, $\langle fire, made \rangle$ or $\langle troops, members \rangle$, despite many good alignments also achieved. This happens because the *Edit-distance* returns relatively small values, unable to sufficiently penalize different words, like the ones listed before, to inhibit the alignment. In bioinformatics language, it means that even for such pairs the mutation probability is still high. Another problem of the *Edit-distance* is that it does not distinguish between long and small words, for instance the pairs $\langle in, by \rangle$ and $\langle governor, governed \rangle$ have both the *Edit-distance* equals to 2.

As a consequence, we propose a new function (Equation 4) for word mutation penalization, able to give better answers for the mentioned problems. The idea is to divide the *Edit-distance* value by the length of the normalized[9] maximum common subsequence $maxseq(.,.)$ between both words. For example, the longest common subsequence for the pair $\langle w_1, w_2 \rangle = \langle reinterpretation, interpreted \rangle$ is "*interpret*", with length equal to 9 and $maxseq(w_1, w_2) = \frac{9}{max\{16,11\}} = 0.5625$

$$costAlign(w_i, w_j) = -\frac{edist(w_i, w_j)}{\varepsilon + maxseq(w_i, w_j)} \quad (4)$$

where ε is a small value[10] that acts like a "safety hook" against divisions by zero, when $maxseq(w_i, w_j) = 0$.

word 1	word 2	-edist	costAlign
rule	ruler	-1	-1.235
governor	governed	-2	-2.632
pay	paying	-3	-5.882
reinterpretation	interpreted	-7	-12.227
hamburger	spiritual	-9	-74.312
in	by	-2	-200.000

Table 4: Word mutation functions comparision.

Remark that with the $costAlign(.,.)$ scoring function the problems with pairs like $\langle in, by \rangle$ simply vanish. The smaller the words, the more constrained the mutation will be.

5 Experiments and Results

5.1 Corpus of Paraphrases

To test our alignment method, we used two types of corpora. The first is the "DUC 2002" corpus (DUC2002) and the second is automatically extracted from related *web news stories* (WNS) automatically extracted. For both original corpora, paraphrase extraction has been performed by using the *Sumo-Metric* and two corpora of paraphrases were obtained. Afterwards the alignment algorithm was applied over both corpora.

5.2 Quality of Dynamic Alignment

We tested the proposed alignment methods by giving a sample of 201 aligned paraphrase sentence pairs to a human judge and ask to classify each pair as *correct*, *acorrect*[11], *error* [12], and *merror*[13]. We also asked to classify the local alignment choice[14] as *adequate* or *inadequate*. The results are shown in the next table:

[7]Point Access Mutation.

[8]Blocks Substitution Matrices.

[9]The length of the longest common subsequence divided by the word with maximum length value.

[10]We take $\varepsilon = 0.01$.

[11]Almost correct - minor errors exist

[12]With some errors.

[13]With many errors

[14]Global or local alignment.

	Global				Local
not para	*correct*	*acorrect*	*error*	*merror*	*adequate*
31	108	28	12	8	12/14
15.5%	63.5%	16.5%	7.1%	4.7%	85.7%

Table 5: Precision of alignments.

For global alignments[15] we have 11.8% pairs with relevant errors and 85.7% (12 from 14) of all local alignment decisions were classified as *adequate*. The *not para* column shows the number of false paraphrases identified, revealing a precision value of 84.5% for the *Sumo-Metric*.

6 Conclusion and Future Work

A set of important steps toward automatic construction of aligned paraphrase corpora are presented and inherent relevant issues discussed, like clustering and alignment. Experiments, by using 4 algorithms and through visualization techniques, revealed that clustering is a worthless effort for paraphrase corpora construction, contrary to the literature claims (Barzilay & Lee, 2003). Therefore simple paraphrase pair extraction is suggested and by using a recent and more reliable metric (*Sumo-Metric*) (Anonymous, 2007) designed for asymmetrical entailed pairs. We also propose a dynamic choosing of the alignment algorithm and a word scoring function for the alignment algorithms.

In the future we intend to clean the automatic constructed corpus by introducing syntactical constraints to filter the wrong alignments. Our next step will be to employ *Machine Learning* techniques for rewriting rule induction, by using this automatically constructed aligned paraphrase corpus.

References

Barzilay R. and Lee L. 2002. *Bootstrapping Lexical Choice via Multiple-Sequence Alignment.* Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP), 164-171.

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

[15]Percentage are calculated by dividing by 170 (201 − 31) the number of true paraphrases that exists.

Dolan W.B. and Brockett C. 2004. *Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources.* Proceedings of 20th International Conference on Computational Linguistics (COLING 2004).

Anonymous 2007. *Learning Paraphrases from WNS Corpora.* Proceedings of 20th International FLAIRS Conference. AAAI Press. Key West, Florida.

Daelemans W., Hothker A., and Tjong E. 2004. *Automatic Sentence Simplification for Subtitling in Dutch and English.* In Proceedings of LREC 2004, Lisbon, Portugal.

Heyer L.J., Kruglyak S. and Yooseph S. 1999. Exploring Expression Data: Identification and Analysis of Coexpressed Genes. *Genome Research*, 9:1106-1115.

Hogg R., McKean J., and Craig A. 2005 *Introduction to Mathematical Statistics.* Upper Saddle River, NJ: Pearson Prentice Hall, 359-364.

Jain A., Murty M. and Flynn P. Data clustering: a review. *ACM Computing Surveys*, 31:264-323

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

Kruskal J. B. and Wish M. 1977. *Multidimensional Scaling*. Sage Publications. Beverly Hills. CA.

Le Nguyen M., Horiguchi S., 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.

Levenshtein V. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physice-Doklady, 10:707-710.

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

Papineni K., Roukos S., Ward T., Zhu W.-J. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. IBM Research Report RC22176.

Shinyama Y., Sekine S., and Sudo K. 2002. *Automatic Paraphrase Acquisition from News Articles.* Sao Diego, USA.

Unno Y., Ninomiya T., Miyao Y. and Tsujii J. 2006. *Trimming CFG Parse Trees for Sentence Compression Using Machine Learning Approaches.* In the Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions.

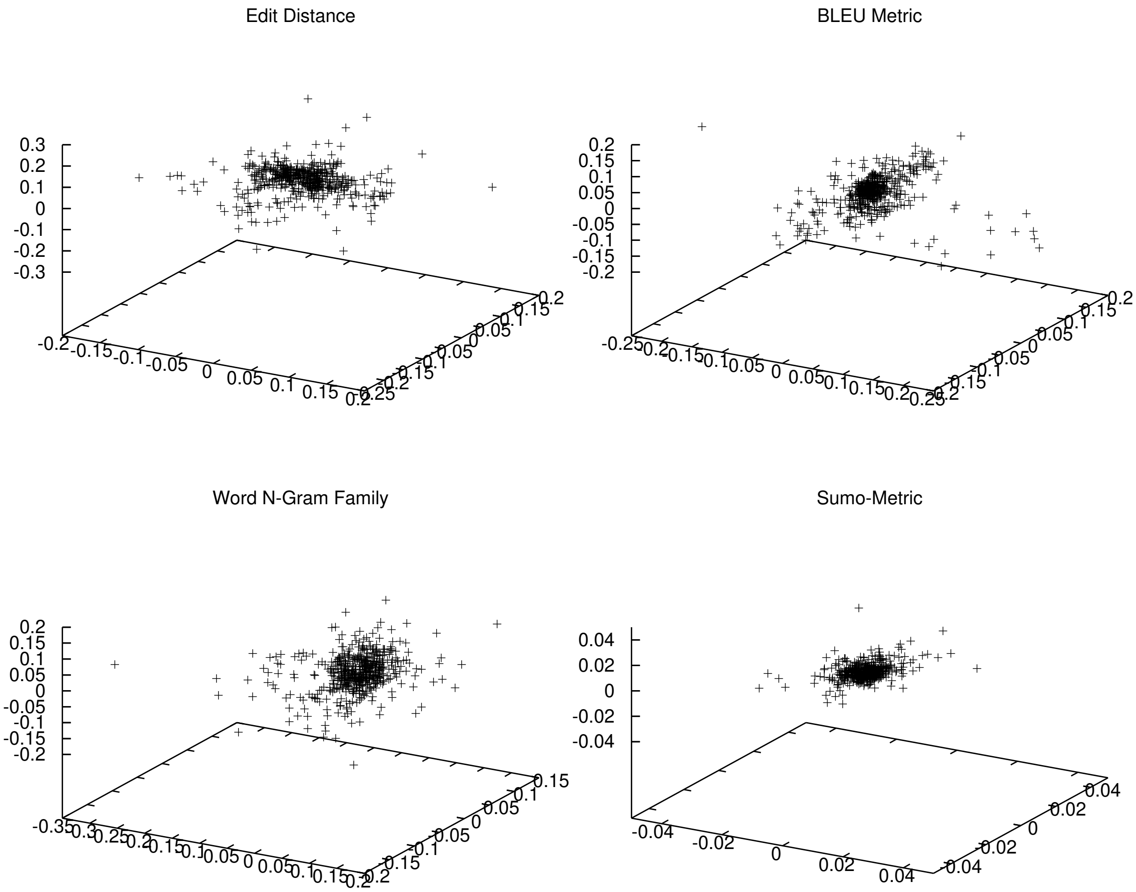

Figure 1: MDS on 500 sentences with the Edit Distance (top left), the BLEU Metric (top right), the Word N-Gram Family (bottom left) and the Sumo-Metric (bottom right).

```
To the horror of their television fans , Miss Ball and Arnaz were divorced in 1960.
__ ___ ______ __ _____ __________ ____ _ ____ Ball and Arnaz ____ divorced in 1960.
```

Figure 2: Global aligned words in a paraphrase pair.

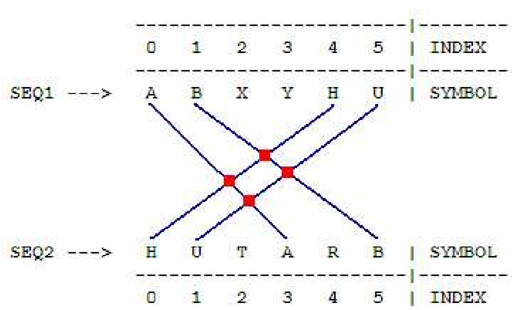

Figure 3: Crossings between a sequence pair.