On Evaluation Methodologies for Text Segmentation Algorithms

Sylvain Lamprier, Tassadit Amghar, Bernard Levrat and Frederic Saubion
LERIA, University of Angers
2, Bd Lavoisier 49000 ANGERS, FRANCE
{*lamprier, amghar, levrat, saubion*}*@info.univ-angers.fr*

Abstract

The WindowDiff evaluation measure [12] is becoming the standard criterion for evaluating text segmentation methods. Nevertheless, this metric is really not fair with regard to the characteristics of the methods and the results that it provides on different kinds of corpus are difficult to compare. Therefore, we first attempt to improve this measure according to the risks taken by each method on different kinds of text. On the other hand, the production of a segmentation of reference being a rather difficult task, this paper describes a new evaluation metric that relies on the stability of the segmentations face to text transformations. Our experimental results appear to indicate that both proposed metrics provide really better indicators of the text segmentation accuracy than existing measures.

1 Introduction

The purpose of automatic text segmentation is to identify the most important thematic breaks in a document in order to cut it into homogeneous units, disconnected from other adjacent parts [14]. More precisely, the segmentation process partitions a text by determining boundaries between contiguous segments related to different topics, defining so semantically coherent parts of text that are sufficiently large to expose some aspect of a given subject. Thematic segmentation of texts can also be seen as a grouping process of basic units (words, sentences, paragraphs...) in order to highlight local semantical coherences [9]. Text segmentation turns out to be very useful for several applications since it aims to provide the structure of a document in terms of the different topics it covers [11]. Many segmentation methods have been proposed and most of them, analyzing the distribution of the words in the text, rely on statistical approaches such as Text Tiling [7], C99 [5], ClassStruggle [10], DotPlotting [13] or Segmenter [8].

Thematic segmentation of text is a rather subjective task. Then, choosing a set of criteria to evaluate and quantify the results may reveal difficult. Main existing measures, presented in Section 2, evaluate the quality of a text segmentation by comparing it with a segmentation of reference [2, 12]. Nevertheless, in Section 3, we realize that this kind of evaluation causes some troubles concerning the building of the reference and the interpretation of the results. Therefore, in Section 4, we attempt to limit the impact of these problems by establishing a normalization function and, in Section 5, we explore a new paradigm of evaluation, less costly and more fair, which does not rely on a reference but assesses the methods w.r.t. the stability of the detected boundaries face to text transformations. Measures are finally experimented in Section 6.

2 Existing evaluation measures

Given a text represented as a sequence $\mathcal{U}$ of n units of text $u_1 \dots u_n$, a segmentation is a pair $\mathcal{S} = (\mathcal{U}, \mathcal{B})$ where $\mathcal{B} \subseteq \mathbb{N}$ is a set of m $(m \leq n-1)$ distinct boundaries $b_1 ... b_m$ such that $\forall i \in [1, m], 1 \leq b_i < n$. A segmentation $(\mathcal{U}, \mathcal{B})$ defines a set of $m+1$ parts of text $u_1 \dots u_{b_1} \mid \dots \mid u_{b_{m-1}+1} \dots u_{b_m} \mid u_{b_m+1} \dots u_n$. In the remaining of the paper, we need a function $\#bound : \mathcal{S} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ which returns the number of boundaries of $\mathcal{B}$ existing between two units u_i and u_j of $\mathcal{U}$ in the segmentation $\mathcal{S} = (\mathcal{U}, \mathcal{B})$. We have then $\#bound(\mathcal{S}, i, j) = card(\{b \in \mathcal{B} / i \leq b < j\})$ where $card$ is the usual set cardinality function. In the following, we note $\mathcal{R} = (\mathcal{U}, \mathcal{B}_\mathcal{R})$ the segmentation of reference and $\mathcal{H} = (\mathcal{U}, \mathcal{B}_\mathcal{H})$ the hypothesis one.

Among the actual evaluation measures of the text segmentation accuracy, cite first the classical measures of Precision, the ratio of true boundaries in $\mathcal{H}$, and Recall, the ratio of referent boundaries retrieved:

$$Recall(\mathcal{H}, \mathcal{R}) = \frac{card(\mathcal{B}_\mathcal{R} \cap \mathcal{B}_\mathcal{H})}{card(\mathcal{B}_\mathcal{R})} \tag{1}$$

$$Precision(\mathcal{H}, \mathcal{R}) = \frac{card(\mathcal{B}_\mathcal{R} \cap \mathcal{B}_\mathcal{H})}{card(\mathcal{B}_\mathcal{H})} \tag{2}$$

These two criteria, commonly used in the field of information retrieval, encounter some problems in the context of text segmentation. First, these criteria are not sensitive face to the *near-misses* [12], i.e., a boundary created at a wrong place leads to the same penalty irrespective of its distance from a true boundary. It is clear that the scores should be better in case of minor errors. Moreover, there is an inherent tradeoff between precision and recall: the increase of one criterion tends to cause the fall of the other. For example, a method segmenting a text more frequently than others tend to obtain a better recall but a worse precision. Some researchers use a weighted combination of both criteria, such as the F1-measure [1]. Others plot a precision-recall curve showing the scores for precision at different levels of recall. However the results are difficult to interpret [3].

In order to solve these problems, a probabilistic evaluation metric has been introduced in [2]. The Pk metric evaluates the probability of error of the segmentation by considering the probability for two sentences to be locally in the same segment in each segmentation (i.e., $\mathcal{H}$ and $\mathcal{R}$). More precisely, for each pair of sentences u_i and u_j separated by a distance k ($|i-j|=k$), the algorithm determines whether both sentences are in the same or in different segments in the reference segmentation, and increases a counter if the segmentation to be evaluated disagrees. The resulting score is normalized by the number of evaluations. In [12], authors have judged this Pk metric really better than the recall and precision but have nevertheless identified several problems:

- false negatives (missing boundaries) are more penalized than false positives (supplementary boundaries),
- some errors are hidden by others,
- the near-miss errors are too much penalized,
- the measure being very sensitive to variations in segment size, results are difficult to interpret.

They thus have proposed a new measure of evaluation for text segmentation methods inspired by the Pk metric. This measure, called WindowDiff, considers the number of boundaries found between two sentences separated from a distance k:

$$\mathit{WindowDiff}\,(\mathcal{H},\mathcal{R}) = \tfrac{1}{n-k}\times \sum_{i=1}^{n-k}(|\#bound(\mathcal{R},i,i+k) - \#bound(\mathcal{H},i,i+k)|) \tag{3}$$

In [12], authors have shown that this measure solves the problems of the Pk metric mentioned above. More precisely, it gains a great stability face to the size variations of the segments and reveals as severe for missing boundaries as it is for false alarms (supplementary boundaries).

3 Motivations

In this section we highlight some remaining problems related to the evaluation of segmentation methods.

3.1 WindowDiff Analysis

A first criticism on the WindowDiff measure (formula 3) concerns the penalties induced by boundaries located at the beginning and at the end of the text. Whereas errors at the extremities of texts are as important as others, w.r.t. formula 3, errors located between sentences 1 and k and between sentences $n-k+1$ and n are less penalized than others since the window takes them into account a lower number of times. This problem can be solved by adding k fictive sentences at the beginning and at the end of the text and computing the score on the shifted segmentations. Another way, employed in the following, is to rewrite the formula:

$$\mathit{WindowDiff}\,(\mathcal{H},\mathcal{R}) = \tfrac{1}{(n+k-2)}\times \left(\begin{array}{ll} \sum\limits_{i=2}^{k}(|\#bound(\mathcal{R},1,i)-\#bound(\mathcal{H},1,i)|) & + \\ \sum\limits_{i=1}^{n-k}(|\#bound(\mathcal{R},i,i+k)-\#bound(\mathcal{H},i,i+k)|) & + \\ \sum\limits_{i=n-k+1}^{n-1}(|\#bound(\mathcal{R},i,n)-\#bound(\mathcal{H},i,n)|) & \end{array}\right) \tag{4}$$

Our second remark is related to the fact that the number of boundaries of the reference and the number of boundaries created by the segmentation methods are important factors in the evaluation process. In order to evaluate the risk (probability) of having a high score of WindowDiff according to these two numbers of boundaries, segmentations of reference and segmentations to test have been generated by a random process. In order to limit random variations, 1000 instances of $\mathcal{R}$ and $\mathcal{H}$ have been generated for each couple of possible number of boundaries between 0 and 100 (assuming we worked on a text of 101 sentences). Figure 1 on the left represents the result of the WindowDiff measurements on these segmentations randomly created. Each curve gives the difference between $\mathcal{R}$ and $\mathcal{H}$ by WindowDiff (with an arbitrary window size of 6) for each number of boundaries of reference ($card(\mathcal{B}_{\mathcal{R}})$) w.r.t. the number of boundaries of the segmentation to test ($card(\mathcal{B}_{\mathcal{H}})$). The fact that the curves are not horizontal lines shows that the expectation of the score obtained by WindowDiff is not fair for the matter of the sensibility of methods. Consequently, a segmentation method, which tends to create more boundaries than the reference, will be penalized while extra boundaries do not necessarily correspond to real errors. It depends on the sensibility of the method.

Moreover, Figure 1 on the right shows that Windowdiff is not fair neither w.r.t. the frequency of boundaries of the reference. The curve represents the WindowDiff results w.r.t. to the number of boundaries of the reference, $\mathcal{R}$ and $\mathcal{H}$ having the same number of boundaries ($card(\mathcal{B}_{\mathcal{R}}) = card(\mathcal{B}_{\mathcal{H}})$). The mathematical expectation of the score of WindowDiff increases with the number of boundaries until it is equal to the half number of sentences.

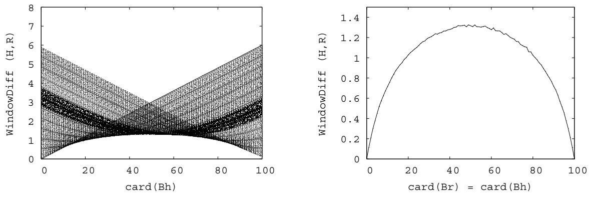

Figure 1. Measurements of WindowDiff with a size of window of 6.

The size of the window used in this measure determines the tolerance of near-misses. Let k be the size of the window and d the number of sentences gap between a boundary detected and its place in the reference. If d belongs to $[1, k/2[$, the score is lower than in the case of no boundary detected. If d equals $k/2$, the score equals the score obtained in the case of no boundary detected. Otherwise, if d belongs to $]k/2, k]$, the score is higher than in the case of no boundary detected but lower than in the case of two errors (one miss and one false alarm). In [12], authors advise users of WindowDiff to set the size of the window to half of the average true segment size. We thus have realized the same tests with a different k for each comparison, which depends on the number of boundaries of the reference and thus on the average size of segments:

$$k = round(\frac{1}{2} \times \frac{n}{card(\mathcal{B}_{\mathcal{R}}) + 1}) \qquad (5)$$

with $round(x)$ returning the integer number closest to x and n being the number of sentences of the segmented text. Figures 2 represent same curves than those of Figures 1 but using this new size of window. They show that changing the size of the window according to the size of segments of the reference, limits the expectation differences. However, the gaps between each changes lead to observations of perturbation waves (see on right figure). Differences of expectations are still present between two different numbers of boundaries to retrieve and thus evaluations on texts of a corpus having a different number of boundaries to retrieve have not the same weight. An evaluation on a corpus is then valid only if all texts have the same number of boundaries to retrieve. Note also that the number of sentences of the text has some effect on the expected results, i.e., results obtained on two texts containing the same number of boundaries to retrieve but not the same number of sentences are thus not comparable. Unfortunately, it is almost impossible to generate a corpus of texts containing the same number of sentences and the same number of boundaries to retrieve. Therefore, results of WindowDiff on two different texts are thus difficult to compare.

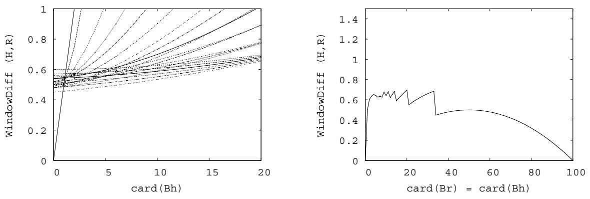

Figure 2. Measurements of WindowDiff with a varying size of window.

Since its curves are not horizontal, left figure shows that differences of sensibility still lead to different score expectations. Consequently, WindowDiff measurement can thus not be the sole criterion to be taken into account. The number of detected boundaries has also to be considered since methods that segment texts with a high frequency rate take more risks to commit errors than other ones. This problem finally comes down to the inherent tradeoff between recall and precision mentionned above.

3.2 Problems induced by the use of a reference

The existing evaluations measures of the text segmentation methods (Precision, Recall, Pk-measure, WindowDiff) compare segmentations computed with segmentations of reference. One may point out some problems related to:

- **Characteristics of the methods:** The computed segmentation depends on the sensibility of the chosen method w.r.t. to the thematic changes of the text and on the size of the basic units to gather. Moreover, a segmentation method may adopt a specific point of view (for example a segmentation oriented by a user's query [4]). Methods that do not have suitable characteristics may be really penalized w.r.t. to the segmentation process used for the reference. For example, a method, which is very sensitive to thematic changes, may create really more segments than the reference. The comparison between a segmentation and the reference is thus not fair w.r.t. the features of the methods.

- **Production of a reference segmentation:** The production of a reference segmentation may turn out to be really difficult. First, it can be hand-made by experts. Unfortunately, this way is really costly and not fully reliable (points of view are often divergent [7]). Similarly to segmentation methods, human subjects may adopt different segmentation strategies. Another way to obtain a segmentation of reference is to use preformatted texts. The paragraphs already exist and the segmentation methods have to recognize them. However, due to the existing transitions, such a segmentation is usually too difficult to realize. Moreover, the difficulty level of such an evaluation is difficult to assess. Finally, the most usually used method in the literature, which appears to be the best suited to obtain a segmentation of reference, is to create automatically texts by gathering different articles from a corpus and by considering the frontiers between the articles as the subtopic boundaries to retrieve [5]. Tests based on concatenated articles may be less conclusive than tests performed on more homogeneous texts but, given the state of the art, this kind of boundaries appears to be difficult enough to recognize.

The problems encountered by WindowDiff led us to adapt it in order to have a more equitable criterion of comparison with a reference segmentation w.r.t. to the different characteristics of the methods and that provides comparable results over different corpuses. Furthermore, in order to address the problems related to the comparison of a segmentation with a segmentation of reference, we have introduced a new measure that evaluates the methods w.r.t. their stability.

4 Normalization of WindowDiff

The observations realized in Section 3.1 led us to look for a normalization function of WindowDiff, in order to be more equitable face to the number of boundaries variations. If we could compute the mathematical expectation function $E(\mathcal{H},\mathcal{R})$ of the WindowDiff measure according to the number of boundaries of each segmentation, we would be able to normalize the results according to the risks taken:

$$NWin(\mathcal{H},\mathcal{R}) = \frac{\mathit{WindowDiff}(\mathcal{H},\mathcal{R})}{E(\mathcal{H},\mathcal{R})} \tag{6}$$

The number of possible combinations of boundaries is $C_{n-1}^{card(\mathcal{B}_\mathcal{R})}$ in the segmentation of reference and to $C_{n-1}^{card(\mathcal{B}_\mathcal{H})}$ in the computed segmentation to evaluate. With k being the size of the window and n the number of sentences, the probability to have i boundaries in a window is equal to $\frac{C_k^i \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{R})-i}}{C_{n-1}^{card(\mathcal{B}_\mathcal{R})}}$ for the reference and to $\frac{C_k^i \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{H})-i}}{C_{n-1}^{card(\mathcal{B}_\mathcal{H})}}$ for the hypothesis. The probability $P(i \neq j)$ to have a different number of boundaries in both segmentations in a window is thus:

$$P(i \neq j) = \\ \sum_{i=0}^{m_r} \sum_{j=0, j\neq i}^{m_h} \frac{C_k^i \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{R})-i}}{C_{n-1}^{card(\mathcal{B}_\mathcal{R})}} \times \frac{C_k^j \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{H})-j}}{C_{n-1}^{card(\mathcal{B}_\mathcal{H})}} \tag{7}$$

with m_r being the minimum between k and $card(\mathcal{B}_\mathcal{R})$ and m_h the minimum between k and $card(\mathcal{B}_\mathcal{H})$. The score of WindowDiff represents the average of scores obtained on each possible window in the text. The probability $P(i \neq j)$ being the same for each one, the mathematical expectation of the score of WindowDiff is:

$$E(\mathcal{H},\mathcal{R}) = \\ \sum_{i=0}^{m_r} \sum_{j=0}^{m_h} \frac{C_k^i \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{R})-i}}{C_{n-1}^{card(\mathcal{B}_\mathcal{R})}} \times \frac{C_k^j \times C_{n-1-k}^{card(\mathcal{B}_\mathcal{H})-j}}{C_{n-1}^{card(\mathcal{B}_\mathcal{H})}} \times |i-j| \tag{8}$$

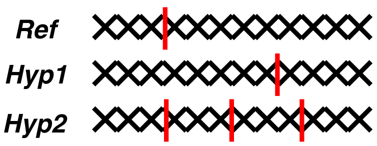

Figure 3. Examples of segmentation.

Figure 3 shows examples of segmentation. In this, the "X" symbols are sentences and "|" are boundaries. The line Ref represents the application of the reference segmentation $\mathcal{R}$ and lines $Hyp1$ and $Hyp2$ the application of two computed segmentations to evaluate, $\mathcal{H}_1$ and $\mathcal{H}_2$, on a sequence of units $\mathcal{U}$. The segmentation $\mathcal{H}_2$ appears to be better than $\mathcal{H}_1$ since it retrieves the boundary existing in the reference. However, the method is more sensitive to thematical changes and thus provide more segments. The WindowDiff score is $2/3$, for both segmentations. The normalization enables us to highlight the dominance of $\mathcal{H}_2$, $NWin(\mathcal{H}_2,\mathcal{R}) = \frac{2/3}{1.16} = 0.57$ whereas $NWin(\mathcal{H}_1,\mathcal{R}) = \frac{2/3}{0.5} = 1.33$. Of course, a segmentation that would not have the additional boundaries of $\mathcal{H}_2$ would obtain a better score but the normalization enables to limit problems induced by differences of sensitivity.

The normalized optimal score that a segmentation can obtain (i.e., the score obtained with all boundaries at their exact place, for segmentations having less boundaries than the reference, and with every boundaries of the reference retrieved, for segmentations having more) w.r.t both numbers of boundaries is:

$$O(\mathcal{H},\mathcal{R}) = \frac{(k \times |card(\mathcal{B}_\mathcal{R}) - card(\mathcal{B}_\mathcal{H})|)}{(n+k-2) \times E(\mathcal{H},\mathcal{R})} \tag{9}$$

If the missing or supplementary boundaries of a segmentation were justified, we could substract this optimal score from the normalized WindowDiff to obtain the real score of the segmentation. However, a too great or too low number of boundaries does not necessarily result from a different behavior face to the thematic changes. One does not know anything about these missing or supplementary boundaries. Nevertheless, we establish a more tolerant measure, called TNWin, that subtracts the optimum O multiplied by a coefficient of tolerance t from the normalized WindowDiff:

$$TNWin(\mathcal{H},\mathcal{R}) = NWin(\mathcal{H},\mathcal{R}) - t \times O(\mathcal{H},\mathcal{R}) \quad (10)$$

The problem of this measure is the confidence we grant to the supplementary or missing boundaries. They can be justified or not. The weight t of the formula corresponds to the probability of justifiability of these boundaries. We set this probability to 0.5 (equiprobability). The experiments realized in Section 6.1 show that this value appears to allow the best segmentations to obtain the best scores.

5 Stability measure

The comparison of a segmentation with a reference raises some problems, especially concerning the inequality w.r.t. the differences of sensitivity of the methods. Our aim is to define an evaluation process that does not need a segmentation of reference. In [14], thematic segments are defined as parts of text with strong intrinsic relationships, disconnected from other adjacent parts. Segments are thus parts of text that have a high internal cohesion and a low similarity with their neighbors. These two criteria do not take into account the order of the sentences in the segments. This observation led us to make some assumptions:

- **Hypothesis 1:** If the initial segments found by a method may be justified somehow (and if that fact is not the result of a stroke of luck), changes in the order of sentences inside the segments (permutations of units u_i and u_j verifying $\#bound(\mathcal{H}, i, j) = 0$) should not disturb the detection of boundaries (i.e., new detected boundaries are identical to initial ones).
- **Hypothesis 2:** If the initial segments found by a method are not correct, changes in the order of sentences inside the segments modify the computations of the segmentation method and the detected boundaries are not the same.
- **Hypothesis 3:** The less accurate a segmentation method is, the fewer chance it has to retrieve the same boundaries, after sentences permutations inside each segments, than those initially detected.

These assumptions, validated beyong in Section 6.2, lead us to establish a new evaluation measure, the Stability Test, which is described by algorithm 1. The stop criterion used in this algorithm is a number of restarts of the process. The higher this number is, the steadier the stability measure is. Nevertheless, a relatively low number is sufficient to highlight segmentation variations. In experiments of Section 6.2, we used a number of 100 restarts.

Algorithm 1 Stability Test

```
Segmentation of the text by the method to assess;
while the stop criterion is not encountered yet do
    - Random sentence permutations inside each segment;
    - Segmentation of the new text by the method to assess;
    - Computation of the recall and the precision of the
    new segmentation w.r.t. the initial boundaries;
end while
- Computation of the averages of recall and precision;
- Computation of the F1-measure w.r.t. both averages.
```

The criteria used for the comparisons between initial boundaries and the ones obtained after sentences permutations are recall and precision. Indeed, WindowDiff cannot been used here since, whereas the accuracy of a method can be evaluated by the number of stable boundaries, it does not depend on the boundaries shift sizes and thus near-misses do not have to be considered. Moreover, the tradeoff existing between recall and precision is not a problem here, since they do not inform on the features of the method but take into account the stability of the boundaries. An F1 measure [1] can then combine both scores with equal importance into a single parameter in order to assess more easily the stability of the boundaries:

$$F_1(Recall, Precision) = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (11)$$

This kind of evaluation does not imply to have a segmentation of reference. It is realized by comparisons between segmentations obtained by the same method. This induces an higher tolerance face to the methods characteristics.

6 Experiments

6.1 Normalized WindowDiff

Experiments have been realized over three existing segmentation methods:

- **TT**: The TextTiling Method[1] [7] is based on the distribution of the terms in the text by giving scores of co-

[1] An implementation in C is available on: www.sims.berkeley.edu /˜hearst/.

hesion between textual units according to several criteria (number of common words, number of new words, number of active lexical chains).

- **C99**: The C99 method[2] [5] introduces a "local ranking" of the similarities between textual units by determining for each pair the number of surrounding pairs having a lower similarity in the k closest pairs ($k = 11$ is the default value of C99). Then, the whole textual units are laid out on a 2D plan (in a similar way to the DotPlotting method [13]) according to their position in the text. The algorithm aims at finding the densest areas according to the rank of the units in the ranking carried out. The areas that maximize this density form the subtopic segments of the text.
- **C**$_\beta$: The ClassStruggle method [10] is a statistical linear text segmentation method that uses an initial clustering of the sentences of the text based on their similarity (computed thanks to repetitions of meaningful terms) in order to have a global view on the semantic relations existing between them. In this approach, the obtained clusters evolve by taking into account their proximity in the text. Considering the clusters as topics of the text, ClassStruggle performs a linear traversal of the document in order to determine the most appropriate class assignment for each sentence, depending on their context of occurrence. The process goes along as long as modifications occur in the clusters. Finally, boundaries are created between sentences belonging to two different classes. This method enables the tuning of a parameter β that influences the features of the segmentation obtained. Indeed, the lower this coefficient is, the more frequently segmented the text is. In order to assess the influence of the parameter β, we have realized experiments on multiple versions of the method using different values of β.

The experiments have been carried out on articles from the corpus of text of the international conference TREC-1 [6], which includes full articles from the Associated Press published between 1988 and 1989. Those articles have been gathered to form different sets of 150 documents. Separations between articles constitute the segmentation of reference. Each document owns four boundaries to retrieve and contains an average of 59 sentences (the standard deviation of this number is $\sigma = 17,66$). The average (μ) and standard deviation (σ) results of WindowDiff (WDiff.), precision (Prec.), recall (Rec.) and number of detected boundaries (Nb.) are given in Table 1.

Method	WDiff.		Prec.		Rec.		Nb.	
	μ	σ	μ	σ	μ	σ	μ	σ
TT	0,38	0,25	0,43	0,24	0,50	0,23	5,03	1,93
C99	0,34	0,20	0,45	0,21	0,63	0,25	5,21	1,34
$C_{0,6}$	0,46	0,29	0,43	0,24	0,64	0,26	7,00	2,96
$C_{0,65}$	0,39	0,23	0,48	0,26	0,65	0,26	6,16	2,34
$C_{0,7}$	0,30	0,19	0,55	0,27	0,65	0,27	5,18	1,87
$C_{0,75}$	0,27	0,17	0,57	0,28	0,58	0,27	4,39	1,68
$C_{0,8}$	0,26	0,13	0,56	0,30	0,47	0,27	3,45	1,47
$C_{0,85}$	0,30	0,13	0,46	0,37	0,28	0,26	2,47	1,35

Table 1. Results of the classical measures

In [10], it has been shown that the best values for the β parameter of ClassStruggle are located between 0.7 and 0.75. These observations appear valid here according to the recall and precision criteria. Indeed, versions of ClassStruggle using an higher value do not obtain a better precision. It shows an obvious loss of segmentation quality since the precision should increase with the fall of the number of segments. Same observation is realized for values lower than 0.7, the recall does not increase. Versions of ClassStruggle using $\beta = 0.7$ or $\beta = 0.75$ then appear to be actually better than others. However, we note that the best score of WindowDiff is obtained by the version using the value of 0.8 and that the version using $\beta = 0.85$ obtains the same score than the one using $\beta = 0.7$. It is nevertheless clear that these versions are worse, the consideration of near-misses cannot entirely explain that. The observation mentionned in Section 3.1 appears then to be verified here, WindowDiff is not fair w.r.t. the number of created boundaries, a method segmenting texts more frequently takes more risks to obtain an high score of WindowDiff. Table 2 gives the normalized WindowDiff (NWin) and the tolerant normalized WindowDiff (TNWin), using a coefficient of tolerance $t = 0.5$, of each method on the same corpus.

Method	NWin		TNWin	
	μ	σ	μ	σ
TT	0,50	0,22	0,39	0,17
C99	0,46	0,21	0,33	0,17
$C_{0,6}$	0,51	0,22	0,33	0,18
$C_{0,65}$	0,47	0,22	0,31	0,18
$C_{0,7}$	0,41	0,22	0,29	0,20
$C_{0,75}$	0,40	0,24	0,30	0,21
$C_{0,8}$	0,43	0,22	0,32	0,18
$C_{0,85}$	0,55	0,24	0,42	0,20

Table 2. Results of normalized measures

The normalization appears to really limit the problems of WindowDiff. Indeed, the ranking that it is possible to establish according to the results obtained on NWin is closer to the reality (i.e., the ranking w.r.t. recall and precision) than the one realized w.r.t. WindowDiff. Nevertheless, as mentionned in Section 4, supplementary or missing boundaries penalize versions creating a different number of boundaries than the segmentation of reference. For example, it is clear that $C_{0.7}$ is better than $C_{0.75}$ but it obtains a worse score of

[2] An implementation in java is available on: www.freddychoi.me.uk.

NWin, its supplementary boundaries penalizing it.

The tolerant normalized WindowDiff (TNWin) has been created to solve this problem. It enables to reduce scores of NWin, assuming that supplementary or missing boundaries can be justified in the logic of the method. The ranking established w.r.t. this new measure appears to be the same than the one we could establish w.r.t. precision and recall. The best version (i.e., $\mathrm{C}_{0.7}$) obtains the best score and the worst one (i.e., $\mathrm{C}_{0.85}$) obtains the worst score.

Additional tests have been conducted on multiple corpuses having other sizes and other numbers of boundaries to retrieve. Same observations have been realized, the normalization of WindowDiff enables to more coherently rank the methods w.r.t. their accuracy. Moreover, results being really close to those of Table 2, the normalization seems to well standardize them w.r.t. the features of the texts.

6.2 Stability Test

The purpose of this Section is to assess the validity of the three hypothesis stated above in order to use the stability test for the evaluation of the segmentation methods accuracy.

6.2.1 Hypothesis 1

In order to check the first assumption, we have conducted a study on two groups G_1 and G_2 of ten human subjects each. The experiments were carried out on twenty documents D_1 and D_2 produced for the experiments on the normalization of WindowDiff (Section 6.1). Each document containing five articles, we first asked to both groups of subjects to retrieve separations between articles, G_1 on documents D_1 and G_2 on documents D_2. Nobody made any errors. These results are widely better than ones obtained in the Hearst's study mentionned in Section 3.2. This is due to the fact that boundaries between articles are easier to retrieve. Human subjects can then simulate a perfect segmentation method.

The purpose of the study is to verify the assumption stating that, if the boundaries are justified and are not the result of a stroke a luck, changes in the order of sentences inside each segment do not induce changes in places of detected boundaries. In order not to bias the study, each group exhanges its set of documents, G_1 thus works now on S_2 and G_2 on S_1. Sentences of each previously retrieved segment are randomly permutated, differently for each subject. Each subject receive then ten transformed documents to segment. The results are the same than previously, nobody made any error, the assumption seems thus to be verified.

6.2.2 Hypothesis 2

In order to verify the second assumption, we proceeded same manner as for hypothesis 1. Two groups of ten human subjects G_1 and G_2 had to retrieve boundaries in two sets of ten documents D_1 and D_2. The purpose is to verify the fact that, if the method does not work efficiently, permutations of the sentences in the initial segments induce changes in the detection of boundaries. As mentionned above, human subjects can be assimilate as perfect segmentation methods on the corpus used. This is due by the fact that every subjects speak well english. With a german corpus, subjects speaking not german at all, we could simulate unperfect methods. The documents used here are thus formed by articles taken from the german journal ”Die Welt”. Each document contains five articles.

Same manner as for hypothesis 1, we first asked to both groups of subjects to retrieve separations between articles, G_1 on documents D_1 and G_2 on documents D_2. Only 10% of the detected boudaries were correct. These subjects can thus simulate really unperfect methods, every subjects made a lot of errors. Each group exchanges then its set of documents, G_1 thus works now on S_2 and G_2 on S_1. Sentences of each previously retrieved segment are randomly permutated, differently for each subject. Only 5% of the new boundaries are the same than initial ones. The hypothesis 2 seems to be valid, the relaunching of the segmentation process on a text, after permutations of the sentences in initial segments, leads to changes in detected boundaries if the method does not work efficiently on the text in concern.

6.2.3 Hypothesis 3

The efficiency of human subjects being difficult to assess, we proceeded here same manner as for the experiments on the normalization measures of WindowDiff. Initial segments used are the ones found in Section 6.1 with each method. We aim to retrieve the same ranking of versions w.r.t. the stability measures. Results of precision (STP) and recall (STR) w.r.t. the initial boundaries, in spite of permutations of sentences inside of each initial segment, are given in Table 3. The F1-measure (F_1) is given in order to simplify the interpretation of the results.

Method	STP		STR		F_1	
	μ	σ	μ	σ	μ	σ
TT	0,65	0,14	0,64	0,17	0,64	0,15
C99	0,69	0,18	0,70	0,19	0,69	0,18
$\mathrm{C}_{0,6}$	0,68	0,16	0,70	0,16	0,68	0,15
$\mathrm{C}_{0,65}$	0,71	0,15	0,71	0,16	0,71	0,15
$\mathrm{C}_{0,7}$	0,74	0,16	0,73	0,18	0,73	0,16
$\mathrm{C}_{0,75}$	0,74	0,17	0,71	0,20	0,72	0,18
$\mathrm{C}_{0,8}$	0,72	0,17	0,66	0,22	0,68	0,20
$\mathrm{C}_{0,85}$	0,66	0,22	0,58	0,26	0,60	0,24

Table 3. Results of stability measures

The results appear to be roughly the same than the ones obtained by normalization measures, i.e. the ranking of methods which can be established w.r.t. these stability measures is the same that we could realize according to the recall and precision measures of Table 1. The third hypothesis appears to be valid, the more the segmentation method accurate is, the less the permutations of the sentences in each initial segment perturb the computations, the more stable the segmentation is.

7 Conclusion

This paper deals with the problem of the evaluation of text segmentation methods, and especially with the difficulty of producing a reference and being fair according to the sensibility of the methods to assess. It has shown that results of the WindowDiff criterion are difficult to interpret since they are too dependent on the number of detected boundaries. This criterion favors a lot methods that create few boundaries. Moreover, results coming from different kinds of corpus are not easily comparable.

These observations led us first to normalize the WindowDiff criterion. This normalization is realized by dividing the obtained score of WindowDiff by its mathematical expectation. It enables to adjust the score w.r.t. the risks taken by a method according to the number of boundaries it creates, providing then a really better equality of chances between the different methods. The normalization enables also to adapt the scores according to the number of boundaries to retrieve and the number of sentences of the text, allowing us to compare scores of evaluation on different kinds of text. Nevertheless, the methods creating too many or too few boundaries are still a little bit penalized. A more tolerant measure allows to improve results of such methods by assuming some divergences as justified.

This last measure being not fully reliable and the production of a segmentation of reference being difficult, we propose here an evaluation measure that relies on the stability of the segmentation face to permutations of sentences inside of each initial segment of the text. The more stable a segmentation is, the higher its probability to be accurate is. This measure, being based on multiple restarts of the segmentation method, is not suited to evaluate non-determinist methods. It can not be applied neither to methods relying on linguistic markers since the principle is to test the robustness of the segmentation face to permutations of sentences inside segments. Nevertheless, this new measure presents several advantages. First, this metric enables to evaluate a segmentation method whithout the need of an annotated corpus. Second, the evaluation is done by comparing segmentations realized by the same method. Methods are then evaluated according to their own specific characteristics whithout penalizing any approach. Third, the measure tests the method itself rather than testing its launching on specific examples of texts. This limits the possible strokes of luck of the methods. The size of the corpus can then be really smaller, the launching on a low number of texts is sufficient to assess the accuracy of the method. Indeed, the experiments in Section 6 have been conducted on a corpus of 150 texts but the score of each method would have been roughly the same after 30 examples (contrarily to other evaluation criteria). At last, a rate of confidence can be associated with the provided segmentation.

The experiments realized in Section 6 show that these new metrics are good indicators of the text segmentation methods accuracy.

8 Aknowledgments

This work was supported by "Angers Loire Metropole".

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.

[2] D. Beeferman, A. Berger, and J. Lafferty. Text segmentation using exponential models. In C. Cardie and R. Weischedel, editors, *Proc. of CEMNLP'97*, pages 35–46, 1997. ACL.

[3] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.

[4] P. Bellot. *Méthodes de classification et de segmentation locales non supervisées pour la recherche documentaire*. PhD thesis, Université d'Avignon, Jan. 2000.

[5] F. Y. Choi. Advances in domain independent linear text segmentation. In *Proc. of ACL'00*, pages 26–33, San Francisco, 2000. Morgan Kaufmann Publishers Inc.

[6] D. Harman. Overview of the first trec conference. In *Proc. of SIGIR '93*, pages 36–47, 1993. ACM Press.

[7] M. A. Hearst. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, 1997.

[8] M.-Y. Kan, J. L. Klavans, and K. R. McKeown. Linear segmentation and segment significance. In *Proc. of WVLC'98)*, pages 197–205, 1998. ACL SIGDAT.

[9] H. Kozima. Text segmentation based on similarity between words. In *Meeting of the ACL*, pages 286–288, 1993. ACL.

[10] S. Lamprier, T. Amghar, B. Levrat, and F. Saubion. Classstruggle: a clustering based text segmentation. In *Proc. of SAC'07*, pages 600–604, 2007. ACM Press.

[11] D. McDonald and H. Chen. Using sentence-selection heuristics to rank text segments in txtractor. In *Proc. of JCDL'02*, pages 28–35, 2002. ACM Press.

[12] L. Pevzner and M. A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36, 2002.

[13] J. C. Reynar. *Topic Segmentation : Algorithms and applications*. PhD thesis, University of Pennsylvania, Seattle, 2000.

[14] G. Salton, A. Singhal, C. Buckley, and M. Mitra. Automatic text decomposition using text segments and text themes. In *Proc. of Hypertext'96*, pages 53–65, 1996. ACM Press.