# Finding text boundaries and finding topic boundaries: two different tasks ?

Alexandre Labadié, Violaine Prince

# Finding text boundaries and finding topic boundaries: two different tasks ?

Alexandre Labadié and Violaine Prince

LIRMM
161 rue Ada
34392 Montpellier Cedex 5, France
{labadie,prince}@lirmm.fr
http://www.lirmm.fr

**Abstract.** The goal of this paper is to demonstrate that usual evaluation methods for text segmentation are not adapted for every task linked to text segmentation. To do so we differentiated the task of finding text boundaries in a corpus of concatenated texts from the task of finding transitions between topics inside the same text. We worked on a corpus of twenty two French political discourses trying to find boundaries between them when they are concatenated, and to find topic boundaries inside them when they are not. We compared the results of our distance based method to the well known c99 algorithm.

**Key words:** Topic detection, topic change, evaluation methods, text segmentation

## Introduction

The huge amount of text available on the Internet and other media, allows users to access more and more information. The drawback of this abundance is that information is less and less relevant and workable. Many research fields, such as information retrieval (IR), try to solve this problem by formating data and/or selecting information the more accurately possible. Text segmentation significantly helps improving methods used in these domains since it is considered as one of the fundamental actions in IR [14],[18] .
There are many distinct tasks labeled as 'text segmentation'. For instance, identifying and extracting text from multimedia support where it is mixed with pictures or videos is called as such [13]. The task of grouping words into morphemes or bigger linguistic units is sometimes also referred as text segmentation (e.g. in written Asiatic languages where words boundaries are not easy to assess[26], [27]). In this paper, we concentrate on '**topic based text segmentation**'. This type of process tries to find the topical structure [9] of a text and thus provide a possible thematic decomposition of a given document [21]. Most texts do not talk about only one topic. The bigger the documents, the more topics they include. The goal of topic based text segmentation is to find where a topic begins

and where it ends, within a text. For practical purposes, we will use the name 'text segmentation' to refer to topic based text segmentation.

Basically, the goal of text segmentation is to divide a text into multiple segments which are thematically coherent and distinct. Each of these text segments should ideally bear one topic, but topics could be complex units from a rhetorical point of view, needing explanations, examples or argumentations. This brings out the question of defining the concept of a topic. Browsing literature shows that there are several definitions of a topic and a large body of works in (topical) text segmentation. Generally speaking, a topic is: *the subject matter of a conversation or discussion.* In linguistics, it is defined as: *the part of the proposition that is being talked about (predicated).* Thus one may admit that the topic of a text segment is *what talking is about.* So, the goal of an automatized text segmentation could be simplified into dividing a text in segments, each sentence of which "talks about" the same subject.

To evaluate automatic methods of text segmentation, most papers (among which [6] and [7] are representative examples) use a common protocol: They concatenate multiple texts, and consider each of them as an instance of a thematically coherent text segment. They assume that retrieving text boundaries in a concatenation and segmenting topically a text are equivalent tasks. Although they might appear as syntactically similar, semantically, actions are very different. A concatenation of texts is not designed by an author as a discourse instance, in the way that collecting and grouping several papers on a subject does not make a dissertation about that subject. In this paper, we will question the commonly admitted hypothesis that finding boundaries of concatenated texts and finding boundaries of topic segments are the same task, by presenting two complementary approaches: First, common text segmentation methods which similarly process text boundaries and in text topic boundaries, and second, our approach, which separates both tasks (all described in section 2). Then we will compare one segmentation method of the first type(Choi's c99 algorithm) and our method on the same set of data a French political discourse corpus in section 3. Results will definitely separate the methods capabilities: Common segmentation methods get good results in finding text boundaries, but their performances drop when handling in text topic boundaries, whereas our method shows rather fair results in text boundaries whereas it scores satisfyingly in topic boundaries detection. This section discusses the benefits of considering text boundaries detection and topic change as two different task that should be evaluated differently. We will conclude on possible other approaches of evaluating text segmentation methods.

## 1 Existing methods and the task (tasks ?)

As said in introduction, literature is abundant on the subject, and mostly methods divide into two main categories: Supervised ones, more or less data dependent, and unsupervised methods, trying to avoid the liabilities of learning. In this paper we concentrate on unsupervised methods, since they can be evaluated on corpora as broadly distinct as possible, which is a better case for evaluation.

### 1.1 Main approaches for unsupervised text segmentation

Within this subfield, there are also several methods of text segmentation, but they can be classified into three main approaches.

**Similarity text segmentation methods** These methods consider each text sentence as an atomic element in their analysis and represent it as a vector which, most of the time, is built with the frequency of each term (TF) of the text after the text has been stemmed and purged of useless words with the help of a stop list. To give more weight to 'important' words, the inverse document frequency (IDF) is also quite often present.

The goal of such methods is to measure the gap between sentences, relying on the angle between vectors . A mathematical measure such as the cosine (which is the most used) as a similarity (more exactly, a dissimilarity) measure leads to build similarity matrices, which are employed to search for boundaries in the text.

One of the most efficient similarity based method is probably Choi's C99 algorithm [6]. C99 uses the similarity matrix to build local ranking of proximity between sentences. The more similar to their neighbors the sentences are, the higher their ranks. The lowest rank in the new built ranking matrix shows the boundary between the two main parts of the text. These two parts are then considered as two independent texts, and the algorithm is applied on each part. The algorithm stop when the lowest rank detected is the last sentence of the analyzed part of the text.

**Graphical text segmentation methods** By using a graphical representation of TF, it is easier to see how terms are dispatched all over the text. [10] uses this kind of representation in IR. The principle is quite simple, each word is represented by one or more dots on a a bi-dimensional graphic. The number and positions of dots depend on where and how many times the word appears in the text. For example, a word appearing in sentence $i$ and sentence $j$ will be represented by four dots : $(i, i)$, $(i, j)$, $(j, i)$ and $(j, j)$. Parts of the text where a strong term is repeated appear on the graphic as dot clouds.

This visual approach of TF representation has been used by [23] to develop his DotPlotting algorithm, which identifies text segments by finding the boundaries of the most dense dot clouds. Reynard computes the density of an area of the graphic by dividing the number of dots by the surface of the area. Then the algorithm finds the text segment boundaries by maximizing the density of the dot clouds and/or minimizing the size of "empty" areas in the graphic.

Graphical methods inspired also the one developed by [11], which considers the text segmentation issue as a picture segmentation issue. The authors used an anisotropic diffusion algorithm on a graphic representation of the text distance matrix. By doing so, their algorithm strengthens the divergence between dense areas and boundaries.

**Lexical chains text segmentation methods** Lexical chains text segmentation links multiple occurrences of the same term in a text to form a chain. When the distance between two occurrences of a term is too important, the chain is considered broken. This distance is generally the number of sentences between two consecutive occurrences of one word.

*Segmenter* [12], is software based on this approach with a little specificity: The number of sentences breaking the word chain depends on the syntactic class of the word, thus enhancing discrimination.

Another lexical chain based algorithm, is the *TextTiling* algorithm developed by [8]. A **consistency score** is given to each text block depending the following block. This score is computed on the basis of a first ”lexical” score given to each pair of consecutive sentences. The 'lexical' score is obtained by computing some parameters between two consecutive pairs. These parameters are typically the number of common words between the two pairs, the number of new words and the number of still active lexical chains in the considered sentences. So, the score of each text segment is a normalized scalar product of each pairs score. If a text segment has a very different score from the next and previous text segments, there is a change of topic in this text segment.

### 1.2 Limits in current text segmentation

All these approaches have in common the almost exclusive use of lexical cohesion [20], which means that they only look for similar and/or different words to find text segments or boundaries. If a few use syntactic information, it is limited to the word part-of-speech tag ( noun, verb, adjective, etc.). In natural language, the word/constituent function also bears information. If a noun is the subject of a verb, it could mean something totally different from what it means if it were its object. This lack of syntactic information is one of the limits of such word-based methods.

Another limitation of lexical cohesion based methods which as been pointed by [25], is the intensive use of synonyms as a stylistic effect. In many languages, and particularly in French, the language on which we experiment, repeating several times the same word in a paragraph or even a short text is considered unsightly. This massive use of synonyms makes these approaches quite inefficient as they are based on the exact repetition of words. It is possible to use some semantic resources like WordNet to counterbalance this, but languages requiring such a use of synonyms have also great polysemy issues. So, doing so only changes the problem into another.

More specifically, Bestgen and Piérard [2] have observed that, if these methods are quite efficient at finding text boundaries in a corpus of concatenated texts, they get poor results at finding in text topic segments [2] . These results can be explained by the differences between a whole text and just a segment of it. A text, is a complete entity. With a beginning (generally described as the introduction), a main body (development) and an end (conclusion). So, a text is self-sufficient in terms of information and structure. It does not need any contextual information to be understood. On the other side, a text segment is just a part of a bigger entity. If the text is the 'main topic' then segments are 'sub-topics' and the relation between main and sub is a semantic relation for sure. As an incomplete entity, the segment needs other segments to bear any meaning. Lexical cohesion based methods need a lot of information to be efficient, and most of the time a single topic segment does not bear enough of it. Moreover, as an incomplete entity, it has to refer to other parts of the text to be linked with it, in the way a subtopic is also related to its parent, child or brother subtopic in a topical tree [19].

### 1.3 Transeg: A distance based method

We have developed a distance based text segmentation specifically designated to find topic variations inside the text called Transeg.

**Textual representation** The first step of our approach is to convert each text sentence into a semantic vector obtained using the French language parser SYGFRAN [3]. These vectors are Roget like semantic vectors [24], but using the Larousse thesaurus [16] as a reference. Sentence vectors are recursively computed by linearly combining sentence constituents, which are themself computed by linearly combining word vectors. The weights of each word vectors is the result of a constituents and dependencies syntactic analysis[1]. So, these vectors bear both the semantic and the syntactic information of the sentence.

**Text segmentation** Using this sentence representation, we try to find transition zones inside the text. The notion of transition zone come from the idea that topic change boundaries inside a text are not isolated sentences, but small groups of sentences. To find them, we slide a window along the text, considering each half of the window as a potential segment (fig. 1). Each potential text segment is then represented by one vector, which is a weighted barycenter of its sentence vectors. We added a stylistic information by giving a better weight to first sentences, relying on the fact that introductions bear the important information [15],[17]. Then we compute a distance (we call it thematic distance) between the two barycenter, and consider it as the window central sentence transition score.
Transition zones are successive sentences with a transition score greater than a threshold. This threshold is the result of a detailed observation of DEFT'06 political corpus. We computed distances on many discourses and their topic segments (the sum of their sentences were around 100000) and obtained an average distance of 0.45 and a $\sigma$ of 0.08. Boundary sentences are selected in the transition zones. A more detailed description of this approach can be found in [22].
In our first implementations of this method we used the angular distance to

[1] The formula is given in [4] and has no relation with Kendall's (1948) measure of concordance

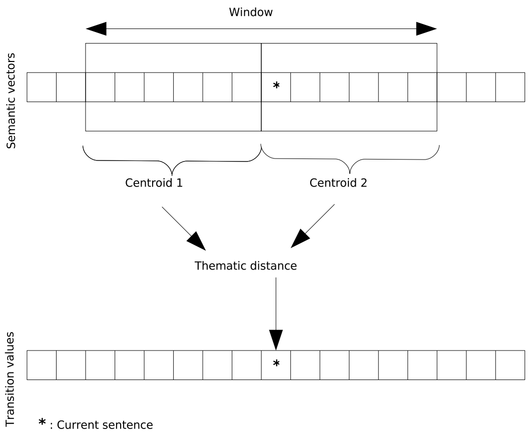

**Fig. 1.** Giving a transition score to each sentences

compute transition score. In this paper we used an extended version of the concordance distance first proposed by [5].

**Concordance distance** Semantic vectors resulting from the analysis have 873 components and most of them are not even activated. With so much null values in the vector the angular distance is not enough discriminant. The goal of the concordance distance is to be more discriminant by not only considering the vectors components values, but their ranks to.
Considering two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$, we sorted their values from the most activated to the less activated and chose to keep only the first values of the new vectors ($\frac{1}{3}$ of the original vector). $\boldsymbol{A_{sr}}$ and $\boldsymbol{B_{sr}}$ are respectively the sorted and reduced versions of $\boldsymbol{A}$ and $\boldsymbol{B}$. Obviously $\boldsymbol{A_{sr}}$ and $\boldsymbol{B_{sr}}$ could have no common strong component (so the distance will be 1), but if they have some we can compute two differences :

**The rank difference:** if $i$ is the rank of $C_t$ a component of $\boldsymbol{A_{sr}}$ and $\rho(i)$ the rank of the same component in $\boldsymbol{B_{sr}}$, we have :

$$E_{i,\rho(i)} = \frac{(i - \rho(i))^2}{Nb^2 + (1 + \frac{i}{2})} \tag{1}$$

Where $Nb$ is the number of values kept.

**The intensity difference:** We also have to compare the intensity of common strong components. If $a_i$ is the intensity of $i$ rank component from $\boldsymbol{A_{sr}}$ and $b_{\rho(i)}$ the intensity of the same component in $\boldsymbol{B_{sr}}$ (its rank is $\rho(i)$), we have:

$$I_{i,\rho(i)} = \frac{\left\| a_i - b_{\rho(i)} \right\|}{Nb^2 + (\frac{1+i}{2})} \tag{2}$$

These two differences allow us to compute an intermediate value $P$:

$$P(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}}) = (\frac{\sum_{i=0}^{Nb-1} \frac{1}{1+E_{i,\rho(i)}*I_{i,\rho(i)}}}{Nb})^2 \tag{3}$$

As $P$ concentrate on components intensities and ranks, we introduce the overall components direction by mixing $P$ with the angular distance. If $\delta(\boldsymbol{A}, \boldsymbol{B})$ is the angular distance between $\boldsymbol{A}$ and $\boldsymbol{B}$, then we have:

$$\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}}) = \frac{P(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}}) * \delta(\boldsymbol{A}, \boldsymbol{B})}{\beta * P(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}}) + (1 - \beta) * \delta(\boldsymbol{A}, \boldsymbol{B})} \tag{4}$$

Where $\beta$ is a coefficient used to give more weight (or less) to $P$. It is easy to prove that neither $P$ nor $\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}})$ are symmetric.
But $\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}})$ was designed in a context of text classification, to compare text vectors to class vectors. As only the likelihood of a text to the class center had to be measured, $\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}})$ did not need to be symmetric. But in our context of text segmentation we needed a symmetric value. even if $\boldsymbol{A}$ come before $\boldsymbol{B}$ in a text, $\boldsymbol{A}$ is not more important than $\boldsymbol{B}$. So the final concordance distance $D(\boldsymbol{A}, \boldsymbol{B})$ we use, is:

$$D(\boldsymbol{A}, \boldsymbol{B}) = \frac{\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}}) + \Delta(\boldsymbol{B_{sr}}, \boldsymbol{A_{sr}})}{2} \tag{5}$$

## 2 Experiment and result on French political discourses

To test the assumption that text and topic boundaries detection are different tasks, we have set up an experiment comparing C99 and our method. Both are unsupervised, therefore not data sensitive (they do not learn, don't adapt to data specificities, therefore a given corpus could be used several times with no effect on results). The first has been tested on concatenated texts by its author, the second has been tested on both concatenated texts and un-concatenated texts in the DEFT'06 [1] competition (an equivalent of TREC Novelty task for French). So in order to compare methods, we tried them on a set of concatenated texts and we measured their scores according to our two criteria : text boundaries detection, in text topic boundaries detection. The following subsections describe data, experiments and results.

### 2.1 Data: a corpus of French political discourse

We chose a corpus of concatenated French political discourses, extracted from the training corpus proposed in the workshop DEFT'06 [1], which proposed several other corpora, but we chose to work on political discourses, for two main reasons:
- As they were identified by experts, internal boundaries looked less artificial than just beginnings of concatenated texts.

- As an argumentative text, the topical structure of a political discourse should be more visible than other more mundane texts. The mentioned workshop was about finding topic boundaries in three different corpora in politics (the one we chose) law and science but we discarded the other domains because of several biases that could be introduced by artificial devices (words such as 'article' in European law texts or paragraph line break that was questionably considered as a topic frontier in the science corpora by the organizers).

There was a lot of noise inside the political corpus. Some discourses were exclusively in capital letters, which is quite annoying when processing a language like French, that discriminates words according to accents on vowels. And some of the "discourses" were, in fact, interviews. So, we manually selected, separated and cleaned discourses from this corpus and created two different corpora:

- Each discourse separately with its internal topic boundaries.
- All discourses concatenated. We only kept the first sentence of each discourse as a boundary (internal topic boundaries were ignored).

From an original corpus of more than $30,0000$ sentences of a questionable quality we extracted 22 discourses totalizing $1,895$ sentences and $54,551$ (Table 1). No information on the discourses were at our disposal, except the beginning of topic segments (which could have been beginnings of texts or real topic boundaries), so this manual cleaning of the corpus took lot of time and significantly reduced the amount data. But it was a necessity to have a workable data set.

The original corpus, full of noise (entire sentences in capital letter, empty sentences, punctuation repetition, etc.), brings some discredit on the DEFT'06 workshop results. But, noise is a common problem in natural language processing and as it should be done with, it should not invalidate the DEFT'06 experiment. In our case, as we tried to differentiate two tasks commonly considered as one, we needed the cleanest data set possible.

### 2.2 Experiments

We set up a first run of both Transeg and the LSA augmented c99 Choi algorithm on the concatenated discourses, and a second one on each discourse separately. We chose to use the latest version of c99 because it is commonly recognized as one of the best text segmentation methods (if not the best at all). To be sure that there is not any implementation error, we used the 1.3 binary release that can be downloaded on Choi's personal Linguaware Internet page (`http://www.lingware.co.uk/homepage/freddy.choi/software/software.htm`).

To evaluate the results of both methods, we used the DEFT'06 workshop tolerant recall and precision ([1]). These recall and precision count as relevant potential boundary sentences which are in a window around the boundary sentence identified by experts. This evaluation give a better idea of algorithms efficiency on the task of finding inner texts topic boundaries and does not have a significant influence on the task of finding texts boundaries. The team of DEFT'06 saw in [1] that the use of either strict or tolerant measure had no effect on the ranking of the submissions they had to evaluate.

We computed the $FScore$ with these tolerant recall and precision, using the well known formula:

$$FScore = \frac{(\beta^2 + 1) * recall * precision}{\beta^2 * precision + recall} \tag{6}$$

With $\beta = 1$.

We have to note that both method consider first sentences of texts as a boundaries and that every first sentence of each text is considered as a boundary when computing recall, precision and $FScore$ (so both methods have always at least one good answer).

### 2.3 Results

All results were multiplied by 100 for legibility purpose First of all, we see that

| | Words | Sentences | Transeg | | | c99 | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Recall | FScore | Precision | Recall | FScore |
| All texts concatenated | 54,551 | 1,895 | 3.68% | 31.82% | 3.3% | 13.33% | 9.09% | **5.41** |

**Table 1.** Results of run 1

| | Words | Sentences | Transeg | | | c99 | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Recall | FScore | Precision | Recall | FScore |
| Text 1 | 617 | 22 | 50% | 33.33% | **40%** | 33.33% | 33.33% | 33.33% |
| Text 2 | 3,042 | 100 | 33.33% | 37.5% | **35.3%** | 50% | 12.5% | 20% |
| Text 3 | 2,767 | 92 | 42.86% | 85.71% | **57.14%** | 20% | 14.29% | 16.67% |
| Text 4 | 1,028 | 40 | 33.33% | 33.33% | **33.33%** | 20% | 33.33% | 25% |
| Text 5 | 4,532 | 157 | 12.5% | 18.18% | **14.82%** | 16.67% | 9.09% | 11.76% |
| Text 6 | 5,348 | 212 | 8.7% | 18.18% | 11.76% | 20% | 18.18% | **19.04%** |
| Text 7 | 1,841 | 47 | 100% | 42.86% | **60%** | 100% | 14.29% | 25% |
| Text 8 | 1,927 | 74 | 60% | 33.33% | **42.86%** | 100% | 11.11% | 20% |
| Text 9 | 1,789 | 53 | 75% | 100% | **85.72%** | 25% | 16.67% | 20% |
| Text 10 | 1,389 | 31 | 33.33% | 20% | 12.5% | 100% | 20% | **16.67%** |
| Text 11 | 2,309 | 81 | 30% | 50% | **37.5%** | 33.33% | 16.67% | 22.22% |
| Text 12 | 7,193 | 211 | 15.38% | 16.25% | **8.88%** | 33.33% | 3.13% | 5.72% |
| Text 13 | 6,097 | 305 | 20.59% | 33.33% | **25.46%** | 17.65% | 14.29% | 15.78% |
| Text 14 | 1,417 | 57 | 40% | 33.33% | **36.36%** | 100% | 16.67% | 28.58% |
| Text 15 | 3,195 | 79 | 40% | 8% | 13.34% | 66.67% | 8% | **14.28%** |
| Text 16 | 1,995 | 60 | 66.67% | 28.57% | 40% | 57.14% | 57.14% | **57.14%** |
| Text 17 | 558 | 16 | 33.33% | 33.33% | 33.33% | 50% | 66.67% | **57.14%** |
| Text 18 | 696 | 25 | 100% | 37.5% | **54.54%** | 40% | 25% | 30.76% |
| Text 19 | 678 | 26 | 33.33% | 33.33% | 33.33% | 50% | 66.67% | **57.14%** |
| Text 20 | 1,388 | 57 | 50% | 66.67% | **57.14%** | 100% | 16.67% | 28.58% |
| Text 21 | 3,127 | 110 | 62.5% | 25% | **35.72%** | 40% | 10% | 16% |
| Text 22 | 1,618 | 40 | 60% | 75% | **66.66%** | 100% | 25% | 40% |

**Table 2.** Results of run 2

results are not spectacular (be it in table 1 or table 2). $FScore$ is a very strict measure, even when softened by using tolerant recall and precision. The best $FScore$, obtained by Transeg in run 2 text 9, is of 85.72% for a precision of 75% and a recall of 100% and the worst is of 5.72%. This give us a good view of the quality of current text segmentation methods and of the progresses we can make in this domain.

Considering run 1, c99 has a better $FScore$ and precision than Transeg. This confirm our initial postulate that c99 is better than us at finding texts boundaries. But, we also see that both methods have overall bad results. When watching in detail the results of both methods we see that:

- C99 bring back only 15 potential boundaries also it should have bring back at least 22 (one for each text). And only 2 of them are in the tolerance window.
- Transeg bring back 190 potential boundaries (which is far to much), for only 7 in the tolerance window.

These results are significant of the differences between the two approaches. Transeg has been conceived to be very sensitive to variations. So on the many sentences composing the corpus, it detected many variations. Transeg is clearly too sensitive for such tasks. C99 detected far less variations and seem far less sensitive, why? C99 is designed to detect brutal changes in the lexical field. As our corpus is exclusively composed of political discourses, texts are quite uniform. This could explain its overall bad results (even if better than us) on run 1.

Considering run 2, Transeg has a better $FScore$ on 16 on the 22 composing the corpus. On these 16 texts our recall is always better or equal to c99 and our $FScore$ are from 20% (text 1) to 329% (text 9) better than c99 ones. Transeg has also the best $FScore$ of both runs with 85.72% on text 9. C99 has a better $FScore$ on 6 texts, but it is at best twice Transeg $FScore$ on the same text. Anyway, we should notice that c99 has comparatively good precision on most of the texts. Thus, when examining texts where c99 is better we see that they are in two categories: - Texts with few boundaries. C99 seems to be very effective on short texts with just one inner topic boundary. With few boundaries identified, and first sentences always identified as boundaries, mathematically c99 has a very good precision on such short texts (text 10 for example). - Enumerations. Text 6 for example, which is quite big, is a record of the government spokesman where he enumerates dealt subject during the weekly minister reunion. So it is basically an enumeration of different subjects with different vocabularies and no real transition between the different segments.

## 3 Conclusion

In this paper we presented strong evidences that finding text boundaries in a corpus of concatenated texts and finding topic segments inside a specific text are two different task that need (at least) two different approaches. As we already said in the introduction and in the first section, lexical cohesion based methods are more efficient at identifying entire texts in a corpus of concatenated texts than at finding topic boundaries inside texts. . On the opposite methods integrating syntactic, semantic and/or stylistic information seem to be more sensitive to small variation inside a text and are more appropriated when it come to find topic segments inside a text. So, developing methods specifically for one or another of these tasks could be a better approach than the current one consisting in considering the two tasks as one.

Judging these results we should also consider evaluation methods specifically designated for each task. If it is easy to create data set to evaluate methods that find texts boundaries inside a huge amount of concatenated texts. It is far more difficult to find data sets where inner topic segments are identified. Such corpus need at least one linguistic or domain expert to identify each potential topic boundaries, which is very time consuming even for a small amount of text. And one expert is probably not enough. Due to the subjectivity of such task, it is better to ask two groups of expert to generate the corpus. The first to propose boundaries and the second to validate. This would be far more time consuming and cost consuming than only one expert of course.

We were lucky to have the DEFT'06 corpus to test our method. But, topic boundaries, in this corpus, were identified by people managing the government Internet site. They are supposed to be political experts and to have the skills to find change of topics inside a political discourse. But are their boundaries all exact ? And a better question, are their choices the only right ones ? As we already said, topic based text segmentation is a subjective task as well as other natural language processing tasks like automatic summary for example. Maybe are we doing wrong by trying to evaluate these tasks on generated (automatically or by experts) data sets. We are envisaging other ways of evaluating these methods, by, for example, asking experts to evaluate the result of the automatic method and not to generate corpus.

Finally, we should notice the complementarity of both tasks and both approaches. If it is hard consider a fusion of both approaches, the development of an automatic process choosing between methods that concentrate on finding texts and methods that concentrate on finding inner topic segments could be of great help in a domain such as IR.

## References

1. J. Azé, T. Heitz, A. Mela, A. Mezaour, P. Peinl, and M. Roche. Présentation de deft'06 (defi fouille de textes). *Proceedings of DEFT'06*, 1:3–12, 2006.
2. Y. Bestgen and S. Piérard. Comment évaluer les algorithmes de segmentation automatiques ? essai de construction d'un matriel de référence. *Proceedings of TALN'06*, 2006.
3. J. Chauché. Un outil multidimensionnel de l'analyse du discours. *Proceedings of Coling'84*, 1:11–15, 1984.
4. J. Chauché and V. Prince. Classifying texts through natural language parsing and semantic filtering. *In Proceedings of LTC'03*, 2007.
5. J. Chauché, V. Prince, S. Jaillet, and M. Teisseire. Classification automatique de textes  partir de leur analyse syntaxico-sémantique. *Proceedings of TALN'03*, pages 55–65, 2003.

6. F. Y. Y. Choi. Advances in domain independent linear text segmentation. *Proceedings of NAACL-00*, pages 26–33, 2000.
7. F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. *Proceedings of EMNLP*, pages 109–117, 2001.
8. M. A. Hearst. Text-tilling : segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, pages 59–66, 1997.
9. M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. *Proceedings of the ACM SIGIR-93 International Conference On Research and Development in Information Retrieval*, pages 59–68, 1993.
10. J. Helfman. Similarity patterns in language. *Visual Languages*, pages 173–175, 1994.
11. X. Ji and H. Zha. Domain-independant segmentation using anisotropic diffusion and dynamic programming. *Proceedings of ACM/SIGIR Conference of Research and Developpement in Information Retrieval*, 2003.
12. M. Kan, J. L. Klavans, and K. R. McKeown. Linear segmentation and segment significance. *Proceedings of WVLC-6*, pages 197–205, 1998.
13. D. Karatzas. *Text Segmentation in Web Images Using Color Perception and Topological Features*. ECS Publications, UK, 2003.
14. M. Kaszkiel and J. Zobel. Passage retrieval revisited. *In Proceedings of theTwentieth International Conference on Research and Development in Information Access (ACMSIGIR)*, pages 178–185, 1997.
15. A. Labadié and Chauché. Segmentation thématique par calcul de distance sémantique. *Proceedings of DEFT'06*, 1:45–59, 2006.
16. Larousse. *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, Paris, 1992.
17. A. Lelu, C. M., and S. Aubain. Coopération multiniveau d'approches non-supervises et supervises pour la detection des ruptures thématiques dans les discours présidentiels franais. *In Proceedings of DEFT'06*, 2006.
18. F. Llopis, A. Ferrandez, J. L. Vicedo, and G. A. Text segmentation for efficient information retrieval. *Proceedings of CICLing*, 2276:373–380, 2002.
19. K. McCoy and J. Cheng. Focus of attention: Constraining what can be said next. *in C. Paris, W. Swartout, and W. Mann, ' Natural Language Generation in Artificial Intelligence and Computational Linguistics'*, 1991.
20. J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:20–48, 1991.
21. J. M. Ponte and W. B. Croft. Text segmentation by topic. *European Conference on Digital Libraries*, pages 113–125, 1997.
22. V. Prince and A. Labadié. Text segmentation based on document understanding for information retrieval. *In Proceedings of NLDB'07*, pages 295–304, 2007.
23. J. C. Reynar. *Topic Segmentation: Algorithms and Applications*. Phd thesis, University of Pennsylvania, 1998.
24. P. Roget. *Thesaurus of English Words and Phrases*. Longman, London, 1852.
25. L. Sitbon and P. Bellot. Evaluation de méthodes de segmentation thématique linéaire non supervisés après adaptation au franais. *Proceedings of TALN'04*, 2004.
26. Z. Wu and G. Tseng. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of the American Society for Information Science*, 44:532–542, 1993.
27. C. C. Yang and K. W. Li. A heuristic method based on a statistical approach for chinese text segmentation. *Journal of the American Society for Information Science and Technology*, 56:1438–1447, 2005.