Intended boundaries detection in topic change tracking for text segmentation

Alexandre Labadié[1] and Violaine Prince[1]

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

Abstract. This paper propose a topical text segmentation method based on intended boundaries detection and compare it to a well known default boundaries detection method, c99. We ran the two methods on a corpus of twenty two French political discourses and results showed us that intended boundaries detection is better than default boundaries detection on well structured text.

Introduction

Topical text segmentation is becoming an important issue in information retrieval (IR) applications. It addresses the function of dividing texts into segments corresponding to different topics.A direct application would be retrieving appropriate segments to a query [9], [18], instead of complete texts, in which the user would not easily find the few sentences concerning his/her specific need. Another is topical tagging of segments, to create titles or subtitles, useful in applications where huge amounts of linear texts are provided without sections. A third is using text topic segmentation (also called subtopic segmentation in some papers) in automatic summarization [8].
This issue has been tackled by several researchers in both IR and natural language processing (NLP). It could be summarized into two major actions: Either detecting topical boundaries, i.e. finding where the topic changes [7], or detecting topics as such, i.e, retrieving the sentences that have been recognized as 'speaking about' a given topic (an issue called topic detection and tracking, TDT). This paper focuses on **topic change**, that is, how topical boundaries could be detected and thus lead to text segmentation.
Most methods in topic change detection are based on recognizing boundaries *by default*: They assume that a topic border is to be drawn in the no man's land between two topical areas, 'where a large shift in the vocabulary occurs'[17]. The way these areas are defined is generally by a similarity or a density measure. The area goes as long as similarity (respectively density) are sufficiently respected. Two representatives of these methods are presented in section 1. Their main liability relies on the proper NLP unit on which they built their areas: They restrict their data to words, thus loosing the rhetorical and syntactic information conveyed by texts. The issue tackled in this paper is thus the following: Assuming that this information is not unuseful, could one produce a method that takes it into account, and evaluate it on data? In other words, could we track a topic boundary, not as a default choice, but as a deliberate one, thanks to the structural information (rhetorical, syntactic) embedded in natural language output? This has lead us to define an **intended boundary** detection action, described and evaluated in section 2.

1 Text segmentation by default boundaries detection: C99 and DotPlotting

We chose to present two methods, among several others, detecting boundaries by default. We chose them because c99 is considered as very efficient, and Dotplotting because it is typical of the default boundary detection philosophy.

1.1 C99

Developed by Choi [5], c99 is text segmentation algorithm strongly based on the lexical cohesion principle [15]. It is, at this time, one the best and most popular algorithms in the text segmentation domain [2]. C99 uses similarity matrix of the text sentences. First projected in a word vectorial space representation, sentences are then compared using the cosine similarity measure (by the way, the most used measure). Similarity values are used to build the similarity matrix. More recently, Choi improved c99 by using the Latent Semantic Analysis (LSA) achievements to reduce the size of the word vectorial space [6]. The author then builds a second matrix known as the *rank matrix.* The latter is computed by giving to each case of the similarity a rank equal to the number of cases around the examined one (in a layer) which have a lesser similarity score. This rank is normalized by the number of cases that were really inside the layer to avoid side effects. C99 then finds topic boundaries by recursively seeking the optimum density of matrices along the rank matrix diagonal. The algorithm stops when the optimal boundaries returned are the end of the current matrix or, if the user gave this parameter to the algorithm, when the maximum number of text segments is reached.

1.2 DotPlotting

Another well known text segmentation algorithm is an adaptation of DotPlotting to text segmentation proposed by [19]. This algorithm is based on a graphical representation of the text, where each word is one or more dots on 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. Then, the algorithm try to regroup dots on the graphic in clouds with an optimal density. These dots clouds are the topical segments.

1.3 Limits of such approaches

Default boundaries detection methods only regroup sentences into 'density bags' (or similar concepts) neglecting the text structure (be it topic structure, syntactic or semantic structure). This lack of structural information can lead to some mistakes, like missing the transition between two different but close topics, for example.
Although preferring to detect boundaries by default, other methods, based on the concept of **lexical chains**, try to introduce an 'intended boundary detection'. Lexical chains text segmentation methods link 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 to be broken. This distance is generally the number of sentences between two consecutive occurrences of one word. $Segmenter$ from [8] and $TextTiling$ from [7] are two good examples of such methods.
Even when searching for lexical breaks, these methods do not really focus on understanding the nature of a transition between two topics. They only assume that a change in the lexical field is a change of topic. If a change in the lexical field, most of the time, leads to a change of topic, there can be change of topic without a significant change in the lexical field. For instance, if we take the subsections of this section, there is no significant change in vocabulary whereas one addresses a given method, the second another method, and the third focuses on their liabilities. In fact, it is more a lexical first occurrence (like the word 'cloud' or 'dot' in the second subsection) than a break in the lexical chain (around the words 'text', 'segment' or 'algorithm') that could be a clue for a possible subtopic beginning [16].

1.4 Relevant non lexical approaches

Previously presented approaches concentrate on the lexical aspect of the text to find text segments. Strongly based on the lexical cohesion principle [15], these methods ignore (or don't use so much) other information like : syntax, style or rhetoric. Some approaches based on the Rhetorical Structure Theory [13] like the one presented by [14], try to use discourse markers to find the structure of a text. But, these methods are supervised and quite domain dependent and our work focuses on unsupervised domain independent methods.

We developed a text segmentation method, based on a vectorial representation of the text and on distances between these vectors, concentrating on intentionally searching for boundaries between topic segments, by defining these intended boundaries.

2 Intented boundaries detection by thematic distance computing

$Transeg$, the method we developed, is based on a vectorial representation of the text and on a precise definition of what we assume a transition between two text segments should be.

2.1 Vectorial representation of the text

The first step of our approach is to convert each text sentence into a semantic vector obtained using the French language parser SYGFRAN[3] (Any other parser for any other language, providing a constituents and dependencies analysis would be compatible with our approach). These vectors are Roget like semantic vectors ([20]), but using the Larousse thesaurus ([11]) as a reference. Sentence vectors are recursively computed by linearly combining sentence constituents, which are themselves computed by linearly combining word vectors. The weights of each word vectors are computed with a formula relying on a constituents and dependencies syntactic analysis (The formula is given in [4]). So, these vectors bear both the semantic and the syntactic information of the sentence.

2.2 Transition zones and boundaries:

In well written structured texts, the transition between a topic and the next one is not abrupt. An author should conclude one topic before introducing another. We called this specific part of text between two segments the **transition zone**. Ideally, the transition should be composed of two sentences:
- The last sentence of the previous segment.
- The first sentence of the beginning new segment.
$Transeg$ tries to identify these two sentences in order to track topic boundaries.

Transition score and beginning of a new segment: The **transition score** of a sentence represents its likelihood of being *the first sentence of a segment*. To compute this score, we supposed that every sentence of the text is the first sentence of a ten sentences long segment. We compared this 'potential segment' with another potential segment composed by the ten preceding sentences. This size of ten sentences for a segment was chosen by observing results on the corpus of French political discourses we work on, segmented by human experts. We saw that the average size of a segment was around ten sentences (10.16) with a σ of (3.26). So decided to use this empirical value as the standard segment size. However, this value has no impact on boundaries detection. Any other might fit as well.
To compute the score of each sentence of the text, we slide a twenty sentences long window along the text, considering each half of the window as a potential segment. 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 ([10],[12]). Then we calculate a distance (called thematic distance) between the two barycenters, and consider it as the window *central sentence transition score* (figure 1). In our first experiments, we used

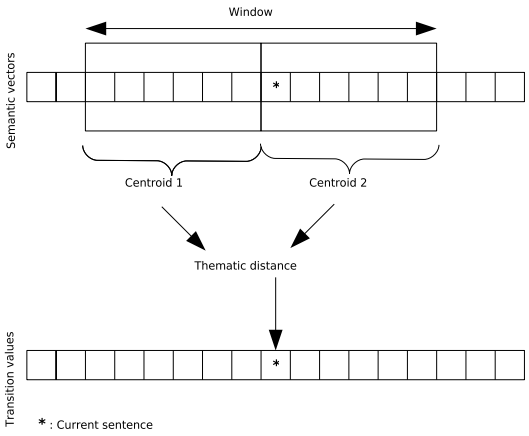

Fig. 1. The transition score of a sentence represent its likelihood of being the first sentence of a segment

the angular distance between two vectors as a thematic distance, but found it not discriminant enough (although better than a cosine, because the latter does not equally cover the two halves of a 90 degree angle). We now use the augmented concordance distance, which has been designed to be as discriminant as possible.

Augmented concordance distance: Semantic vectors resulting from the analysis have 873 components and most of them are not even activated. With so many null values in the vector, angular distance is not really representative of a shift in direction. The goal of the concordance distance is to be more discriminant by not only considering the vectors components values, but their ranks too.
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 β is a coefficient used to give more weight (or less) to P. $\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}})$ is the concordance distance, presented in [4].It is easy to prove that neither P nor $\Delta(\boldsymbol{A_{sr}}, \boldsymbol{B_{sr}})$ are symmetric. But in our context of text segmentation we needed a symmetric value. So we augmented the concordance distance:

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

Transition zones Once each sentence has a transition score, we identify parts of the text where boundaries are likely to appear. These transition zones are successive sentences with a transition score greater than a determined threshold (figure 2). As we defined the ideal transition zone as a two sentences long text segment, isolated sentences are ignored. The

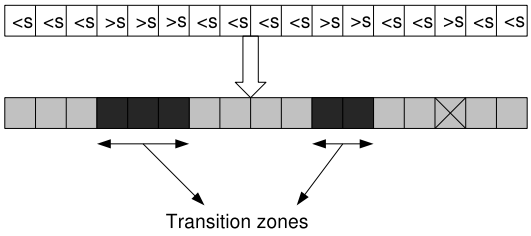

Fig. 2. Identifying transition zones

threshold we chose is 0.45. As for the standard size of a segment, this value has been deduced from our corpus. In order to know whether it is corpus dependent or not, we segmented we browsed two other corpora segmented by human experts, and belonging to the fields of computer science an law (these corpora were available for the DEFT06 competition on text segmentation participants. [1]). The threshold seemed to remain constant on these data. This is not a proof that it is completely corpus independent, and needs to be further investigated. However, at a first attempt, it resisted variation, and we assumed it to be representative of a 'natural trend' of topical discrimination, among other criteria, of course. We computed augmented concordance distances between all identified text segments and as a result we have an average distance of 0.45 with a σ of 0.08.

Ending sentences and breaking score: To identify boundaries inside transition zones we needed another information. We defined the transition score of a sentence as its likelihood of being the first sentence of a segment. The **breaking score** is a sentence likelihood of being *the last sentence of a segment*.
We supposed that the last sentence of a topic should conclude the topic and more or less introduce the next topic. So the thematic distance of this sentence to its segment should be quite equal to the thematic distance of this sentence to the next segment. The breaking score B_i of the i sentence is:

$$B_i = 1 - |D_p - D_n| \tag{6}$$

Where D_p is the thematic distance of the sentence to the previous segment and D_n the thematic distance of the sentence to the next segment. The closer D_p and D_n are to each other, the closer to 1 B_i is.
The last step of our method consists in multiplying the transition score of of each sentence of a transition zone with the breaking score of the previous sentence. The higher score has high probabilities of being the first sentence of a new segment.

2.3 Experiments

We compared our method to the popular c99 algorithm [5]by running them on twenty two French political discourses of our corpus. These discourses have been extracted from the original corpus because they were far cleaner and more usable than the average of the corpus. To be safe from any implementation errors, we used the LSA augmented c99 algorithm implementation proposed by Choi himself (http://www.lingware.co.uk/ homepage/freddy.choi/software/software.htm). The results of our experiment are presented in table 1.

They show that in 16 texts over 22, Transeg has a better Fscore value than c99. Knowing that the corpus is composed of political discourses where syntax and rhetoric are important elements, this gives credit to the assumption that neglecting them would possibly reduce the topical segmentation efficiency. In the 6 texts where c99 performed better, we noticed that they were either short, one topic texts, and then Transeg over-segmented them, or they contained enumerations of different actions, and therefore, sensitivity to lexical shifting is more efficient than to structural information. However, the proportion indicates that most texts are complex structures, and convey precious indications going beyond the sole use of words.

	Words	Sentences	Transeg			c99		
			Precision	Recall	FScore	Precision	Recall	FScore
Text 1	617	22	50	33.33	**20**	33.33	33.33	16.67
Text 2	3042	100	33.33	37.5	**17.65**	50	12.5	10
Text 3	2767	92	42.86	85.71	**28.57**	20	14.29	8.33
Text 4	1028	40	33.33	33.33	**16.67**	20	33.33	12.5
Text 5	4532	157	12.5	18.18	**7.41**	16.67	9.09	5.88
Text 6	5348	212	8.7	18.18	5.88	20	18.18	**9.52**
Text 7	1841	47	100	42.86	**30**	100	14.29	12.5
Text 8	1927	74	60	33.33	**21.43**	100	11.11	10
Text 9	1789	53	75	100	**42.86**	25	16.67	10
Text 10	1389	31	33.33	20	12.5	100	20	**16.67**
Text 11	2309	81	30	50	**18.75**	33.33	16.67	11.11
Text 12	7193	211	15.38	6.25	**4.44**	33.33	3.13	2.86
Text 13	6097	305	20.59	33.33	**12.73**	17.65	14.29	7.89
Text 14	1417	57	40	33.33	**18.18**	100	16.67	14.29
Text 15	3195	79	40	8	6.67	66.67	8	**7.14**
Text 16	1995	60	66.67	28.57	20	57.14	57.14	**28.57**
Text 17	558	16	33.33	33.33	16.67	50	66.67	**28.57**
Text 18	696	25	100	37.5	**27.27**	40	25	15.38
Text 19	678	26	33.33	33.33	16.67	50	66.67	**28.57**
Text 20	1388	57	50	66.67	**28.57**	100	16.67	14.29
Text 21	3127	110	62.5	25	**17.86**	40	10	8
Text 22	1618	40	60	75	**33.33**	100	25	20

Table 1. Comparison between c99 and Transeg

3 Conclusion

In this paper we have considered that topic change detection methods, for text segmentation, generally rely on lexical information, and tend to discard other types of information existing in texts, e.g., rhetorical, stylistic and syntactic information, generally subsumed under the label of structural information. They also favor default topical boundaries detection, whereas focused detection on intended boundaries suggest other possible tracks for asserting topic change. Assuming that structural information has a role to play in detecting intended boundaries, we built a segmenter, called Transeg, based on spotting transition zones between topics in texts. This paper has focused on transition zones definition and the appropriate actions to detect them, by assigning a transition score and a breaking score to each sentence of the text. The transition score indicates its ability to play the role of the first sentence of a segment, and the breaking score, its likelihood of being the last one. With values over a given threshold, transition and breaking score become representative of an intended topical boundary. To determine the efficiency of Transeg, we evaluated it by running it on the same corpus as c99, a popular default boundaries detection algorithm. Results have shown that structural information has an impact on segmentation efficiency.

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. F. Y. Y. Choi. Advances in domain independent linear text segmentation. *Proceedings of NAACL-00*, pages 26–33, 2000.
6. F. Y. Y. Choi, P. Wiemer-Hastings, and J. Moore. Latent semantic analysis for text segmentation. *Proceedings of EMNLP*, pages 109–117, 2001.
7. M. A. Hearst. Text-tilling : segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, pages 59–66, 1997.
8. M. Kan, J. L. Klavans, and K. R. McKeown. Linear segmentation and segment significance. *Proceedings of WVLC-6*, pages 197–205, 1998.
9. M. Kaszkiel and J. Zobel. Passage retrieval revisited. *Proceedings of theTwentieth International Conference on Research and Development in Information Access (ACMSIGIR)*, pages 178–185, 1997.
10. A. Labadié and Chauché. Segmentation thématique par calcul de distance sémantique. *Proceedings of DEFT'06*, 1:45–59, 2006.
11. Larousse. *Thésaurus Larousse - des idées aux mots, des mots aux idées*. Larousse, Paris, 1992.
12. A. Lelu, C. M., and S. Aubain. Coopération multiniveau d'approches non-supervises et supervises pour la détection des ruptures thématiques dans les discours présidentiels français. *In Proceedings of DEFT'06*, 2006.
13. W. Mann and S. A. Thompson. Rhetorical structure theory : A theory of text organization. Technical Report ISI/RS-87-190, University of Southern California, Information Sciences Institute, 1987.

14. D. Marcu. The rhetorical parsing of natural language texts. *In Proceedings of the Meeting of the Association for Computational Linguistic*, pages 96–103, 1997.
15. 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.
16. R. J. Passonneau and D. Litman. Lintention-based segmentation: Humanreliability and correlation with linguistic cues. *Proceedings of the 31st Annual Meeting of theAssociation for Computational Linguistics,*, pages 148–155, 1993.
17. L. Pevzner and M. Hearst. A critique and improvement of anevaluation metric for text segmentation. *Computational Linguistics*, pages 113–125, 2002.
18. V. Prince and A. Labadié. Text segmentation based on document understanding for information retrieval. *In Proceedings of NLDB'07*, pages 295–304, 2007.
19. J. C. Reynar. *Topic Segmentation: Algorithms and Applications*. Phd thesis, University of Pennsylvania, 1998.
20. P. Roget. *Thesaurus of English Words and Phrases*. Longman, London, 1852.