Topic driven access to scientific handbooks

Caracciolo, C.

CHAPTER 6

Connecting Map and Link Targets

In the previous chapter we have seen how two topic segmentation algorithms and two structural segmentation methods perform against a manually created gold standard segmentation on a domain specific corpus, viz. the *Handbook of Logic and Language*. Now, we explore the use of the segments obtained using these algorithms and methods as targets of link targets. Specifically, our aim is to find link targets for topics in the LoLaLi map, so that the topics can serve as starting points for exploring the *Handbook*. We operationalize the task of finding link targets as follows: we select a set of topics from the map and use them as queries to run against the collection of segments. For evaluation purposes we need to define a gold standard as well as appropriate measures to evaluate the results.

The remainder of this chapter is organized as follows. In Section **??** we discuss the creation of an annotated corpus to evaluate the results of the retrieval runs. In Section **??** we introduce three evaluation measures that we use to evaluate single retrieved segments. In Section **??** we generalize these measures to evaluate sets of segments. In Section **??** we describe the experimental setting, present the retrieval methods used, and discuss the results obtained. Finally, in Section **??**, we discuss the work carried out in the course of this chapter, and formulate our conclusions.

6.1 Annotation of Relevance Assessments

We used topics from the map as sources for which link targets needed to be identified. Topics were selected based on the content of the two chapters used for the annotation described in Section 5.3.3, i.e., Chapter 3 [van Eijck and Kamp, 1997] and Chapter 10 [Muskens et al., 1997] from the *Handbook of Logic and Language*. Titles of topics

were taken as queries.

Guidelines. The annotation was performed by a single annotator who annotated relevant paragraphs for each query. The notion of relevance utilized here is binary, where a paragraph is relevant to a certain topic if it "talks about" it.[1] The annotator was given the following guidelines:

1. the minimal unit of relevance is a paragraph;
2. if a paragraph contains a cross-reference (to another section of the document), ignore the cross-reference.

The rationale for the first instruction is to annotate the smallest unit used in the segmentation phase. The rationale for the second instruction is that we hypothesized that cross-references within the text will be rendered as hyperlinks and therefore immediately available to the end user. We used the same two chapters as for the experiments described in Section 5.3.3 in order to gain multiple views of the same data. The annotator was given the chapters with no indication of the annotation of segments, and he knew nothing about the previous segmentation.

For the annotation we used 37 queries and 191 paragraphs were found relevant (43 in Chapter 3, 148 in Chapter 10): on average, each query has 5.1 relevant paragraphs.

6.2 How to Evaluate a Link Target

The manual annotation for relevance is given in terms of blocks of paragraphs, and, as we have just seen, segments can be one or more paragraphs long. Then, a retrieved segment might include only paragraphs that are marked as relevant, or none, or some relevant paragraphs. Figure **??** gives a schematic representation of these possibilities: the horizontal line represents the entire document and the small vertical lines mark paragraphs. Above the line, relevant paragraphs are those marked by r, and below the line there are the retrieved segments. Then, segment A is totally non-relevant, segment B totally relevant, and segments C, D, E include both relevant and non-relevant paragraphs. Note that this figure does not present the results of an actual retrieval run, as the segments obtained by the application of the segmentation algorithms we used are all non-overlapping.

In our view, a retrieved segment is relevant if it contains at least one relevant paragraph (i.e., segments B, C, D and E in Figure **??** are all relevant), but it would also be useful to keep track of the *amount* of relevant matter in a segment returned by a system for generating link targets. Therefore, we introduce a measure for precision of a segment, that we call C-precision (C_p), to account for the number of relevant paragraphs

[1] The analytical review of relevance by Saracevic [1975] is a key paper about how the notion of relevance is understood. For more recent surveys about the notion of relevance in IR, see [Borlund, 2003] and [Mizzaro, 1997]. For the notion of aboutness, we refer to the work by Hutchins [1977].

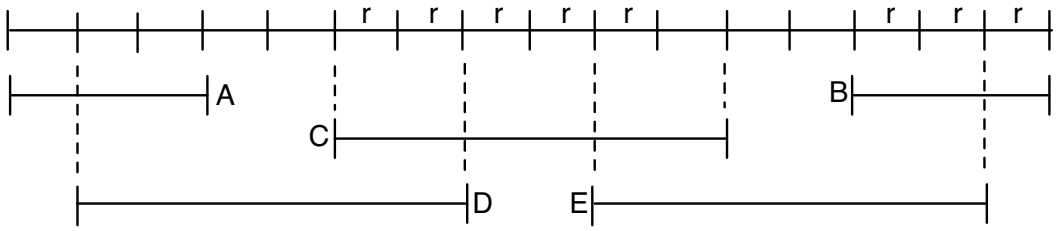

Figure 6.1: Possible configuration of retrieved segment with respect to the annotation for relevance.

out of the total number of paragraphs in the segment returned by a system; see below for more details.

Although informative, this measure fails to distinguish retrieved segments in terms of *where* the relevant paragraphs are, neither inside the segment, nor with respect to the other (contiguous) relevant paragraphs in the document (outside the segment). In other words this measure does not capture the notion of *entry point* to the document [de Vries et al., 2004]. For example, consider Figure **??** and the difference between segment *C* and segment *D*. Segment *C* starts with a relevant paragraph, does not have relevant paragraphs preceding it, and includes all relevant contiguous paragraphs in the area; segment *D*, on the other hand, includes four non-relevant paragraphs before the first relevant one, and leaves out a number of contiguous relevant paragraphs after that.

In order to be able to take these factors into consideration, we introduce two error measures. We say that a segment has an *onset error* if it either begins with one or more non-relevant paragraphs (*early onset error* (EoE)), or if it does begin with a relevant paragraph, but some contiguous (preceding) relevant paragraph was not included in the segment (*late onset error* (LoE)). It has *no onset error* otherwise (NoE), if it begins with a relevant paragraph and has no contiguous relevant paragraphs preceding it. The late and early onset errors are related to the notion of an entry point to the text (cf. Section 2.1). Note that we do not take into consideration the "exit point" of a segment, i.e., its ending point: since the exit point of a segment and the entry point of the following segment are contiguous, considering both would be redundant. In the following we discuss in detail each of these measures.

C-precision. Let S be a retrieved segment. The C-precision of S, $C_p(S)$, is defined as the proportion of relevant paragraphs included in S:

$$C_p(S) = \frac{|Relevant\ in\ S|}{|S|}, \tag{6.1}$$

where $|S|$ is the total number of paragraphs in S.

The segments depicted in Figure **??** have the following C-precision scores: $C_p(A) = 0$; $C_p(B) = 1$; $C_p(C) = 5/6$; $C_p(D) = 1/3$; $C_p(E) = 3/6$.

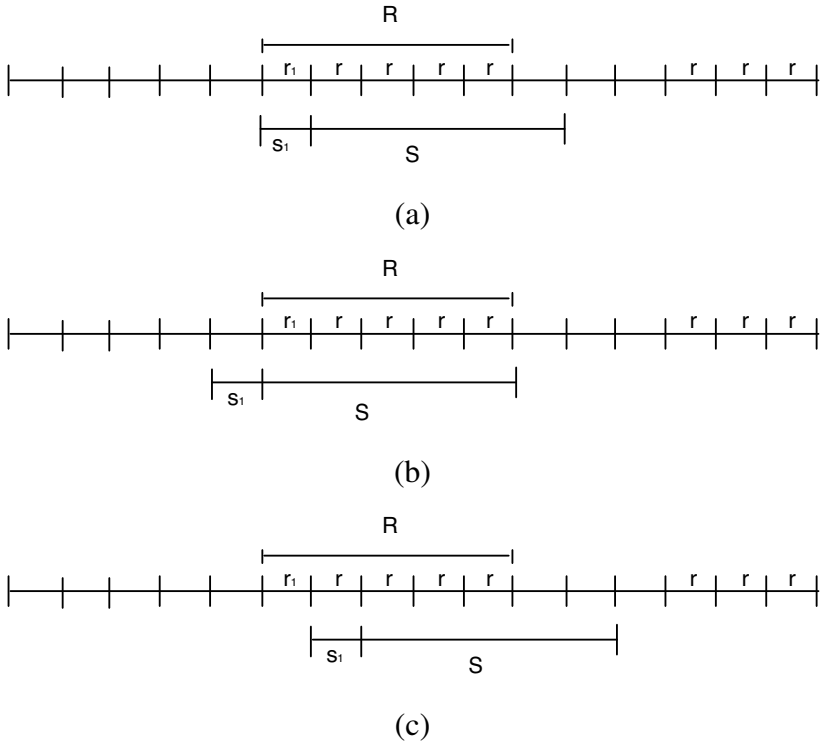

Figure 6.2: (a) Segment with NoE. (b) Segment with EoE. (c) Segment with LoE.

Early onset Error. If a segment begins with one or more non-relevant paragraphs, like segment D in Figure **??**, we say that the segment has an early entry point (Early onset Error, or EoE), EoE measures the *proportion of superfluous paragraphs* included at the beginning of the segment.

Paragraphs in a text are linearly ordered, so that for any pair of paragraphs a and b ($a \neq b$) it is always possible to say whether $a < b$ or $b > a$ in the order (i.e., a comes before b or the other way around). Therefore, the distance between two points in the text can be expressed in terms of the number of paragraphs separating them. Having made this observation, let us call s_1 the first paragraph of a retrieved segment S. Let r_1 be the first paragraph of a sequence of contiguous relevant paragraphs R intersecting S (Figure **??**). If $s_1 = r_1$, the segment has optimal entry point (Figure **??** (a)), therefore we expect that both early and late onset errors are zero (see segments B and C in Figure **??**). When $s_1 < r_1$, segment S has an early entry point (see Figure **??** (b), or segment D in Figure **??**), measured by the early onset error.

The *Early onset Error (EoE)* measures the proportion of non-relevant material included in the segment preceding the first relevant paragraph (i.e., $r_1 - s_1 > 0$). More formally:

$$EoE = \min\left\{1, \min_{R:s_1 \leq r_1} \frac{r_1 - s_1}{|S|}\right\}, \tag{6.2}$$

where $|S|$ stands for the size of S expressed in terms of number of paragraphs if contains, and $r_1 - s_1$ is the number of non-relevant paragraphs in S preceding the relevant ones. Note that $\frac{r_1 - s_1}{|S|}$ is equal to 0 when there is no error, i.e., the beginning of the retrieved segment coincides with the beginning of the relevant block. And $\frac{r_1 - s_1}{|S|} \geq 1$ when the error is maximum, i.e., when the onset is so early that the entire segment includes only non-relevant paragraphs. The minimum in the above equation restricts the range of EoE to exactly $[0,1]$. The advantage of restricting the range of EoE is that segments containing only non-relevant paragraphs have maximum error, independently of their distance from the closest relevant paragraph.

Late onset Error. If a retrieved segment begins with one or more relevant paragraphs, but at least one relevant paragraph precedes its beginning, like segment E in Figure **??**, we say that the segment has a late entry point (i.e., it makes a Late onset Error, or LoE). In other words, the LoE measures the *proportion of missed relevant paragraphs* at the beginning of the segment. Analogously, if $s_1 - r_1 > 0$ the LoE measures the proportion of relevant paragraphs missed at the beginning of the segment (Figure **??** (c)). More formally:

$$LoE = \min \left\{ 1, \min_{R: r_1 \leq s_1} \frac{s_1 - r_1}{|R|} \right\}, \tag{6.3}$$

where $|R|$ stands for the number of relevant paragraphs at the beginning of segment S. Then, $s_1 - r_1$ is the number of relevant paragraphs preceding s_1, and $\frac{s_1 - r_1}{|R|}$ is equal to 0 when there is no error, i.e., the beginning of the retrieved segment coincides with the beginning of the relevant block. When the error is maximum, i.e., when the onset is so late that the entire segment includes only non-relevant paragraphs $\frac{s_1 - r_1}{S} \geq 1$. Again, the minimum restricts the range of LoE to exactly $[0,1]$.

A segment with perfect entry point (NoE), i.e., coinciding with the beginning of the relevant block R, will have $LoE = EoE = 0$ (segment B and C in Figure **??**). A totally non-relevant segment, i.e., with no relevant paragraphs in it, will have $LoE = EoE = 1$ (see segment A in Figure **??**). No segment can have $LoE = 0$ and $EoE = 1$; similarly, no segment can have $LoE = 1$ and $EoE = 0$.

Discussion. C-precision is meant to give a measure of the quantity of relevant content in a candidate link. It takes into account the size of the segment in an indirect manner, so that a segment consisting of only one relevant paragraph has $C_p = 1$, a segment containing no relevant paragraphs has $C_p = 0$, and a segment including all relevant paragraphs in the document will likely have low precision, how low exactly depends on the size of the document (and on the number of relevant paragraphs in it).

The other two measures, EoE and LoE, have the advantage of providing a relative measure, ranging from 0 to 1. EoE has a bias for longer documents, since it divides the number of non-relevant paragraphs by the total number of paragraphs in the segment. Arguably, shorter documents should not be penalized, although what should or should

not be penalized depends on the user modeling we adopt. For example, some users prefer to have longer documents retrieved and navigate into them, others prefer shorter documents to inspect and discard faster.

C-precision and late onset error count relevant paragraphs in a slightly different manner. When computing C-precision, all relevant paragraphs in the segment are counted, while the late onset error only considers the sequence of relevant paragraphs at the beginning of the segment. Let us consider again the example depicted in Figure **??**. Segment *E* in that figure contains three relevant paragraphs out of six (its C-precision is 0.5), "distributed" into two blocks, with some non-relevant text in the middle: here we have a segment that contains a mixture of relevant and non-relevant paragraphs. If we are only interested in the amount of relevant content in the segment, then all relevant paragraphs in it should be counted, independently of their position in the text. On the other hand, if we are interested in the readability of the segment, only its first paragraph should be considered and, in that is relevant, the contiguous relevant paragraphs after that.

The three measures just introduced will be used to assess the quality of single retrieved segments. In the next section we aggregate them in order to be able to evaluate the behaviour of entire collections of segments, obtained in the previous chapter.

6.3 Evaluating a Collection of Segments

In order to be able to assign a score to a set of segments (as opposed to a single segment) with respect to a query or even a set of queries, we consider the following ways of aggregating per-segment results:

1. average number of segments with NoE (per topic);
2. average C-precision;
3. total number of non-relevant segments retrieved;
4. average number of non-relevant paragraphs at the beginning of a relevant segment;
5. average early onset error (EoE);
6. average number of relevant paragraphs missed at the beginning of a relevant segment;
7. average late onset error (LoE).

All measures are applied at cut-off three, i.e., only considering the first three retrieved segments for each query. In IR applications based on larger datasets, performance measures based on shallow ranked lists (such as ranked lists that have length three) may be unstable when used with small set of queries [Buckley and Voorhees, 2000,

2004]; however, in our setting and with our user scenario, a cut-off at rank three is a reasonable choice, as our task is highly precision-oriented (cf. Chapter 5) and the dataset (document collection) relatively small.

By looking at the proportion of segments with NoE per query, we look at how many segments per query provide an "ideal" entry point. The value of C-precision averaged for all queries gives a measure of the "amount" of relevant matter contained in all the retrieved segments. The third measure we take into consideration is the total number of non relevant segments retrieved. The last four measures deal with the quality of an entry point to the document. The fourth and fifth measures are coupled together: the average number of non-relevant paragraphs at the beginning of a relevant segment gives a measure of the *effort* required by the reader in order to reach the first relevant paragraph available, while the average EoE gives a normalized measure of that effort. The sixth and seventh measures are analogous to the previous two measures, because the relevant text starts before the beginning of the document, and one has to move back in order to find a good entry point. These measures depend on the size of the document and on the number of contiguous relevant paragraphs available.

6.4 Experimental Setting and Results

Now that we have defined our evaluation measures, we turn to our experiments. The research question that we formulated in Chapter 1 (Research Question 4) is the following: What is the most suitable candidate link to be connected to the map? In other words, we ask what the relative performance of link target finding on different segmentations is. Do "better" segmentations give rise to "better" link target finding results?

In order to answer our question we considered the each sets of segments as a collection of documents within which to retrieve relevant segments with respect to the selected queries. By running these experiments we aim at seeing what the difference is, if any, between the collections of segments obtained with different segmentation methods. The information retrieval engine that we used in the experiments we describe in this chapter was developed at the University of Amsterdam [van Hage, 2004]. Documents are preprocessed using TreeTagger [Schmid, 1994], then indexes and inverted files are stored in a database for reasons of efficiency; the weighting schemas used are tf.idf and OKAPI [Robertson and Walker, 1994]. Both are briefly described in Appendix **??**: the former schema is widely know as an appropriate weighting schema in case of short documents [Allan et al., 2003], and so is the latter, because it combines idf with a notion of the "normal" size of a document in the collection, expressed as the average of the size of all documents. In this way the average size of documents in the collection is used as a reference point against which to compare the length of a document. OKAPI is still one of the best performing retrieval schemas, despite its age.

For our link target finding experiments we use the outputs of the segmentation methods described in Section 5.3.3 and summarized in Table 5.6. For evaluation purposes, we used the ground truth described in Section **??** and the measures described in

Segm. method	Avg. NoE/topic	Avg. C_p	Tot. non-rel. segm.	Avg. non-rel. par. begin	Avg. EoE	Avg. rel. par. missed	Avg. LoE
Paragraphs	**1.08**	**0.36**	**69**	**0.00 (0)**	**0.64**	2.11 (17)	0.71
Sections	0.83	0.07	78	7.77 (13)	0.72	3.00 (1)	0.67
TT default	1.06	0.35	70	**0.00 (0)**	0.65	2.12 (16)	0.71
TT s5-w20	1.00	0.33	72	**0.00 (0)**	0.67	2.05 (15)	0.73
TT s5-w30	1.03	0.34	71	**0.00 (0)**	0.66	2.12 (16)	0.72
TT s20-w30	1.06	0.32	71	1.00 (3)	0.67	2.05 (12)	0.70
TT s20-w40	1.06	0.31	71	3.50 (2)	0.67	**1.78 (12)**	0.69
C99 default	0.92	0.09	78	6.44 (16)	0.75	4.88 (5)	**0.66**
C99 r9	0.83	0.06	81	10.78 (19)	0.81	6.00 (1)	0.67
C99 r57	0.97	0.10	77	6.00 (17)	0.76	4.88 (5)	0.67

Table 6.1: Summary values for all collections of segments, across all queries, using tf.idf. Highest scores per measure are in **boldface**.

Sections **??** and **??**.

Table **??** and **??** report the scores for the measures introduced in the previous section, respectively for tf.idf and OKAPI. These tables summarize results across all 37 queries used on both chapters.[2] The highest scores are in **boldface**.

Table **??** shows that, using the tf.idf weighting scheme, the segmentation into paragraphs achieves the best scores for most of the measures. The reason for this good result is that if a segment only contains one paragraph, and that one is relevant, the entire segment has maximum C-precision and perfect entry point. On the other hand, if the paragraph is non-relevant, the entire segment is non-relevant, and the proportion of early onset does not apply.

Using tf.idf, the segmentation by section gets rather low scores for all measures. In fact, the longer a segment is, the more likely it is that it also contains non-relevant paragraphs, which directly affects the C-precision of the segment. If the segment is long, it is also more likely that the non relevant paragraphs in it are placed at the beginning of the segment. For example, for segments based on sections, the proportion of late onset error is quite high, but there is only one case where it happens, clearly in one case when the annotator judged relevant a sequence of paragraphs spanning two sections. The number of non-relevant segment retrieved when segments are as long as entire sections suggests that tf.idf tends to discriminate short documents better than long ones.

The segmentation obtained by applying TextTiling show figures similar to the seg-

[2]We discarded one of the queries, as it had no relevant text in the documents.

Segm. method	Avg. NoE/topic	Avg. C_p	Tot. non-rel. segm.	Avg. Non-rel. par. begin	Avg. EoE	Avg. Rel. par. missed	Avg. LoE
Paragraphs	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	**2.42 (18)**	0.72
Sections	0.47	0.04	90	6.50 (10)	0.83	3.00 (1)	0.79
TT default	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s5-w20	**1.06**	**0.35**	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s5-w30	**1.06**	0.34	**70**	**0.00 (0)**	**0.65**	2.36 (16)	0.71
TT s20-w30	0.97	0.29	74	1.00 (3)	0.69	2.48 (11)	0.72
TT s20-w40	1.00	0.29	73	3.50 (2)	0.69	**2.42 (13)**	0.71
C99 default	0.78	0.07	85	6.15 (13)	0.77	5.50 (4)	**0.70**
C99 r9	0.67	0.05	87	8.93 (14)	0.81	6.00 (1)	0.73
C99 r57	0.72	0.07	88	4.33 (12)	0.81	5.50 (4)	0.75

Table 6.2: Summary values for all collections of segments, across all queries, using OKAPI. Highest scores per measure are in **boldface**.

mentation by paragraphs, which squares with the observations just made. In contrast, the segmentation obtained by applying C99 gets similar figures to the segmentation by sections.

Table **??** presents the results obtained running the retrieval algorithm using the OKAPI weighting schema and the same evaluation measures as in Table **??**. In general, the results exhibit a pattern similar to those obtained with the tf.idf weighting schema. Values for single paragraphs are quite stable with respect to the previous run, although some values have decreased slightly. Based on the results shown in Tables **??** and **??** we cannot conclude that there is a substantial difference in performance between tf.idf and OKAPI for our application. Instead, the fact that the results we obtain are consistent, although slightly different, confirms the fact that both weighting schemas are appropriate for retrieving short documents and that tf.idf is hard to beat.

Again with OKAPI, the segmentations obtained by applying TextTiling and the structural segmentation by paragraphs have similar results, if not exactly the same. Also the highest number of non-relevant segments retrieved is obtained when the algorithm retrieves entire sections. The segmentation based on paragraphs yields the highest scores in terms of proportion of NoE, while the division into sections is the one that scores worst. All C99 versions perform only slightly better than segmentation by sections. Again, paragraphs have the highest C-precision, together with a few of the segmentation based on TextTiling. C-precision for sections is extremely low, and all versions of C99 show very similar results.

Concerning the early onset error, we recall that it implies that the segment begins

with at least one non-relevant paragraph. It is equal to 1 when the segment is totally non-relevant, equal to 0 when there are no non-relevant paragraphs at the beginning of the segment. The single paragraph segmentation and TextTiling have the lowest average EoE, a fact that is explained by the high precision: since a single paragraph can only be either totally relevant or totally non-relevant, it follows that in case of many relevant segments, there will be many zeros on average. This is also witnessed by the fact that C-precision and early onset error sum to one for this system. C99 with default settings scores the highest EoE, due to the large size of the segments. Concerning the LoE, this time the lowest error rate is scored by C99 with default parameters, immediately followed by the baseline based on paragraphs.

Summarizing, when all measures are taken into account, the best collections of segments are those obtained by applying TextTiling (the closer to the default values, the better) and the structural segmentation by paragraphs. TextTiling produces short segments but not always consisting of single paragraphs (cf. Table 5.6).

6.5 Conclusions

We have considered the link generation process as a result of two distinct phases: first, texts are segmented into topically coherent segments, and then segments are retrieved on the basis of their similarity to the queries. We addressed the fourth question asked in Chapter 1 (Research Question 4): What is the most suitable candidate link to be connected to the map? In order to answer this question, we used information retrieval techniques that we applied to the collections of segments that we obtained in the experiments presented in the previous chapter.

For the evaluation of our work, we introduced a measure for the content of relevant matter in a segment (C-precision), and two measures of error in the entry point (early onset error, late onset error). We approximated the notion of readable entry point by means of a quantitative measure of the distance between the beginning of the segment and the closest relevant text. The error measures we proposed measure the quality of an entry point in terms of error with respect to the relevant paragraphs included or missed by the segments.

Segments based on entire sections score consistently worse than all other methods. On the other hand, the segments based on single paragraphs and the segments obtained with TextTiling score consistently best, with little or no difference. The success of these two types of segments is partly explained by a slight bias of the retrieval algorithms toward short text. Short segments, if relevant at all, also tend to have high C-precision, because an individual paragraph is either totally relevant or not relevant at all. Moreover, when a segment is only one paragraph long, it can only have a late onset error, never an early onset error, and the early onset error becomes a complement of C_p. This explains why the paragraphs have a small EoE value. Interestingly, the other two algorithms that score the next lowest average EoE (in both retrieval settings) are two variations of TextTiling: the size of the segment constraint the entry onset error. The

average values of LoE have a smaller range than EoE, and C99 with default parameters is the algorithm with lowest value. Again, the size of the segment influences the onset error: in this case the more inclusive a segment is, the lower the LoE.

The conclusion of the work presented in this chapter is that neither the structural segmentation by sections, nor the segmentation by C99 (based on divisive clustering) can be used as a basis for topic driven access. Instead, segmentations by paragraphs and TextTiling show encouraging results as they tend to maximize the amount of relevant content and minimize the early onset error. In particular, given the fact that paragraphs are very "cheap" to obtain, as they are given together with the text, it is worth experimenting with single paragraphs used as a unit for segmentation, and apply semantic treatments à la TextTiling *after* the retrieval phase. We hypothesize that in this way also the late onset error could be minimized. Also, the cost of the processing would be minimal and it would be possible to select the segments to show to the reader at query-time.