Using collocations for topic segmentation and link detection

Olivier FERRET
CEA – LIST/LIC2M
Route du Panorama – BP 6
92265 Fontenay-aux-Roses Cedex
olivier.ferret@cea.fr

Abstract

We present in this paper a method for achieving in an integrated way two tasks of topic analysis: segmentation and link detection. This method combines word repetition and the lexical cohesion stated by a collocation network to compensate for the respective weaknesses of the two approaches. We report an evaluation of our method for segmentation on two corpora, one in French and one in English, and we propose an evaluation measure that specifically suits that kind of systems.

1 Introduction

Topic analysis, which aims at identifying the topics of a text, delimiting their extend and finding the relations between the resulting segments, has recently raised an important interest. The largest part of it was dedicated to topic segmentation, also called linear text segmentation, and to the TDT (Topic Detection and Tracking) initiative (Fiscus et al., 1999), which addresses all the tasks we have mentioned but from a domain-dependent viewpoint and not necessarily in an integrated way.

Systems that implement this work can be categorized according to what kind of knowledge they use. Most of those that achieve text segmentation only rely on the intrinsic characteristics of texts: word distribution, as in (Hearst, 1997), (Choi, 2000) and (Utiyama and Isahara, 2001), or linguistic cues as in (Passonneau and Litman, 1997). They can be applied without restriction about domains but have low results when a text doesn't characterize its topical structure by surface clues.

Some systems exploit domain-independent knowledge about lexical cohesion: a network of words built from a dictionary in (Kozima, 1993); a large set of collocations collected from a corpus in (Ferret, 1998), (Kaufmann, 1999) and (Choi, 2001). To some extend, this knowledge permits these systems to discard some false topical shifts without losing their independence with regard to domains.

The last main type of systems relies on knowledge about the topics they may encounter in the texts they process. This is typically the kind of approach developed in TDT where this knowledge is automatically built from a set of reference texts. The work of Bigi (Bigi et al., 1998) stands in the same perspective but focuses on much larger topics than TDT. These systems have a limited scope due to their topic representations but they are also more precise for the same reason.

Hybrid systems that combine the approaches we have presented were also developed and illustrated the interest of such a combination: (Jobbins and Evett, 1998) combined word recurrence, collocations and a thesaurus; (Beeferman et al., 1999) relied on both collocations and linguistic cues.

The topic analysis we propose implements such a hybrid approach: it relies on a general language resource, a collocation network, but exploits it together with word recurrence in texts. Moreover, it simultaneously achieves topic segmentation and link detection, *i.e.* determining whether two segments discuss the same topic.

We detail in this paper the implementation of this analysis by the TOPICOLL system, we report evaluations of its capabilities concerning segmentation for two languages, French and English, and finally, we propose an evaluation measure that integrates both segmentation and link detection.

2 Overview of TOPICOLL

In accordance with much work about discourse analysis, TOPICOLL processes texts linearly: it detects topic shifts and finds links between segments without delaying its decision, *i.e.*, by only taking into account the part of text that has been already analyzed. A window that delimits the current focus

of the analysis is moved over each text to be processed. This window contains the lemmatized content words of the text, resulting from its pre-processing. A topic context is associated to this focus window. It is made up of both the words of the window and the words that are selected from a collocation network[1] as strongly linked to the words of the window. The current segment is also given a topic context. This context results from the fusion of the contexts associated to the focus window when this window was in the segment space. A topic shift is then detected when the context of the focus window and the context of the current segment are not similar any more for several successive positions of the focus window. This process also performs link detection by comparing the topic context of each new segment to the context of the already delimited segments.

The use of a collocation network permits TOPICOLL to find relations beyond word recurrence and to associate a richer topical representation to segments, which facilitates tasks such as link detection or topic identification. But work such as (Kozima, 1993), (Ferret, 1998) or (Kaufmann, 1999) showed that using a domain-independent source of knowledge for text segmentation doesn't necessarily lead to get better results than work that is only based on word distribution in texts. One of the reasons of this fact is that these methods don't precisely control the relations they select or don't take into account the sparseness of their knowledge. Hence, while they discard some incorrect topic shifts found by methods based on word recurrence, they also find incorrect shifts when the relevant relations are not present in their knowledge or don't find some correct shifts because of the selection of non relevant relations from a topical viewpoint. By combining word recurrence and relations selected from a collocation network, TOPICOLL aims at exploiting a domain-independent source of knowledge for text segmentation in a more accurate way.

3 Collocation networks

TOPICOLL depends on a resource, a collocation network, that is language-dependent. Two collocation networks were built for it: one for French, from the *Le Monde* newspaper (24 months between 1990 and 1994), and one for English, from the *L.A. Times* newspaper (2 years, part of the TREC corpus). The size of each corpus was around 40 million words.

The building process was the same for the two networks. First, the initial corpus was pre-processed in order to characterize texts by their topically significant words. Thus, we retained only the lemmatized form of plain words, that is, nouns, verbs and adjectives. Collocations were extracted according to the method described in (Church and Hanks, 1990) by moving a window on texts. Parameters were chosen in order to catch topical relations: the window was rather large, 20-word wide, and took into account the boundaries of texts; moreover, collocations were indifferent to word order. We also adopted an evaluation of mutual information as a cohesion measure of each collocation. This measure was normalized according to the maximal mutual information relative to the considered corpus.

After filtering the less significant collocations (collocations with less than 10 occurrences and cohesion lower than 0.1), we got a network with approximately 23,000 words and 5.2 million collocations for French, 30,000 words and 4.8 million collocations for English.

4 Description of TOPICOLL

TOPICOLL is based on the creation, the update and the use of a topical representation of both the segments it delimits and the content of its focus window at each position of a text. These topical representations are called *topic contexts*. Topic shifts are found by detecting that the topic context of the focus window is not similar anymore to the topic context of the current segment. Link detection is performed by comparing the context of a new segment to the context of the previous segments.

4.1 Topic contexts

A topic context characterizes the topical dimension of the entity it is associated to by two vectors of weighted words. One of these vectors, called *text vector*, is made up of words coming from the text that is analyzed. The other one, called *collocation vector*, contains words selected from a collocation network and strongly linked to the words of the processed text. For both vectors, the weight

[1] A collocation network is a set of collocations between words. This set can be viewed as a network whose nodes are words and edges are collocations.

of a word expresses its importance with regard to the other words of the vector.

4.1.1 Topic context of the focus window

The *text vector* of the context associated to the focus window is made up of the content words of the window. Their weight is given by:

$$wght_{txt}(w) = occNb(w) \cdot signif(w) \qquad (1)$$

where *occNb(w)* is the number of occurrences of the word *w* in the window and *signif(w)* is the significance of *w*. The weight given by (1) combines the importance of *w* in the part of text delimited by the window and its general significance. This significance is defined as in (Kozima, 1993) as its normalized information in a reference corpus[2]:

$$signif(w) = \frac{-\log_2(f_w / S_c)}{-\log_2(1/S_c)} \qquad (2)$$

where f_w is the number of occurrences of the word *w* in the corpus and S_c, the size of the corpus.

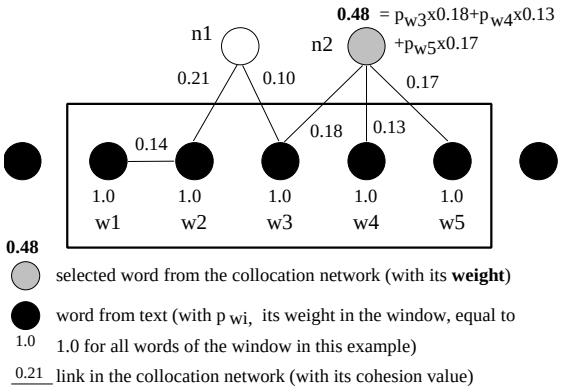

Figure 1 – Selection and weighting of words from the collocation network

The building of the *collocation vector* for the window's context comes from the procedure presented in (Ferret, 1998) for evaluating the lexical cohesion of a text. It consists in selecting words of the collocation network that are topically close to those in the window. We assume that this closeness is related to the number of links that exist between a word of the network and the words of the window. Thus, a word of the network is selected if it is linked to at least *wst* (3 in our experiments) words of the window. A collocation vector may also contain some words of the window as they are generally part of the collocation network and may be selected as its other words.

Each selected word from the network is then assigned a weight. This weight is equal to the sum of the contributions of the window words to which it is linked to. The contribution of a word of the window to the weight of a selected word is equal to its weight in the window, given by (1), modulated by the cohesion measure between these two words in the network (see Figure 1). More precisely, the combination of these two factors is achieved by a geometric mean:

$$wght_{coll}(w) = \sum_i \sqrt{wgth_{txt}(w_i) \cdot coh(w, w_i)} \qquad (3)$$

where $coh(w,w_i)$ is the measure of the cohesion between *w* and w_i in the collocation network.

4.1.2 Topic context of a segment

The topic context of a segment results from the fusion of the contexts associated to the focus window when it was inside the segment. The fusion is achieved as the segment is extended: the context associated to each new position of the segment is combined with the current context of the segment. This combination, which is done separately for text vectors and collocation vectors, consists in merging two lists of weighted words. First, the words of the window context that are not in the segment context are added to it. Then, the weight of each word of the resulting list is computed according to its weight in the window context and its previous weight in the segment context:

$$wght_x(w, Cs, t) = wght_x(w, Cs, t-1) + (signif(w) \cdot wght_x(w, Cw, t)) \qquad (4)$$

with *Cw*, the context of the window, *Cs*, the context of the segment and $wght_x(w,C_{\{s,w\}},t)$, the weight of the word *w* in the vector *x* (*txt* or *coll*) of the context $C_{\{s,w\}}$ for the position *t*. For the words from the window context that are not part of the segment context, $wght_x(w,Cs,t-1)$ is equal to 0.

The revaluation of the weight of a word in a segment context given by (4) is a solution halfway between a fast and a slow evolution of the content of segment contexts. The context of a segment has

[2] In our case, this is the corpus used for building the collocation network.

to be stable because if it follows too narrowly the topical evolution of the window context, topic shifts could not be detected. However, it must also adapt itself to small variations in the way a topic is expressed when progressing in the text in order not to detect false topic shifts.

4.1.3 Similarity between contexts

In order to determine if the content of the focus window is topically coherent or not with the current segment, the topic context of the window is compared to the topic context of the segment. This comparison is performed in two stages: first, a similarity measure is computed between the vectors of the two contexts; then, the resulting values are exploited by a decision procedure that states if the two contexts are similar.

As (Choi, 2000) or (Kaufmann, 1999), we use the cosine measure for evaluating the similarity between a vector of the context window (*Vw*) and the equivalent vector in the segment context (*Vs*):

$$sim(Vs_x, Vw_x) = \frac{\sum_i wg_x(w_i, Cs) \cdot wg_x(w_i, Cw)}{\sqrt{\sum_i wg_x(w_i, Cs)^2 \cdot \sum_i wg_x(w_i, Cw)^2}} \quad (5)$$

where $wg_x(w_i, C_{\{s,w\}})$ is the weight of the word w_i in the vector x (*txt* or *coll*) of the context $C_{\{s,w\}}$.

As we assume that the most significant words of a segment context are the most recurrent ones, the similarity measure takes into account only the words of a segment context whose the recurrence[3] is above a fixed threshold. This one is higher for text vectors than for collocation vectors. This filtering is applied only when the context of a segment is considered as stable (see 4.2).

The decision stage takes root in work about combining results of several systems that achieve the same task. In our case, the evaluation of the similarity between *Cs* and *Cw* at each position is based on a vote that synthesizes the viewpoint of the text vector and the viewpoint of the collocation vector. First, the value of the similarity measure for each vector is compared to a fixed threshold and a positive vote in favor of the similarity of the two contexts is decided if the value exceeds this threshold. Then, the global similarity of the two contexts is rejected only if the votes for the two vectors are negative.

[3] The recurrence of a word in a segment context is given by the ratio between the number of window contexts in which the word was present and the number of window contexts gathered by the segment context.

4.2 Topic segmentation

The algorithm for detecting topic shifts is taken from (Ferret and Grau, 2000) and basically relies on the following principle: at each text position, if the similarity between the topic context of the focus window and the topic context of the current segment is rejected (see 4.1.3), a topic shift is assumed and a new segment is opened. Otherwise, the active segment is extended up to the current position.

This algorithm assumes that the transition between two segments is punctual. As TOPICOLL only operates at word level, its precision is limited. This imprecision makes necessary to set a short delay before deciding that the active segment really ends and similarly, before deciding that a new segment with a stable topic begins. Hence, the algorithm for detecting topic shifts distinguishes four states:

– the *NewTopicDetection* state takes place when a new segment is going to be opened. This opening is then confirmed provided that the content of the focus window context doesn't change for several positions. Moreover, the core of the segment context is defined when TOPICOLL is in this state;

– the *InTopic* state is active when the focus window is inside a segment with a stable topic;

– the *EndTopicDetection* state occurs when the focus window is inside a segment but a difference between the context of the window and the context of the current segment suggests that this segment could end soon. As for the *NewTopicDetection* state, this difference has to be confirmed for several positions before a change of state is decided;

– the *OutOfTopic* state is active between two segments. Generally, TOPICOLL stays in this state no longer than 1 or 2 positions but when neither the words from text nor the words selected from the collocation network are recurrent, *i.e.* no stable topic can be detected according to these features, this number of positions may be equal to the size of a segment.

The transition from one state to another follows the automaton of Figure 2 according to three parameters:

– its current state;
– the similarity between the context of the focus window and the context of the current segment: *Sim* or *no Sim*;
– the number of successive positions of the focus window for which the current state doesn't change: *confirmNb*. It must exceed the $T_{confirm}$ threshold (equal to 3 in our experiments) for leaving the *NewTopicDetection* or the *EndTopicDetection* state.

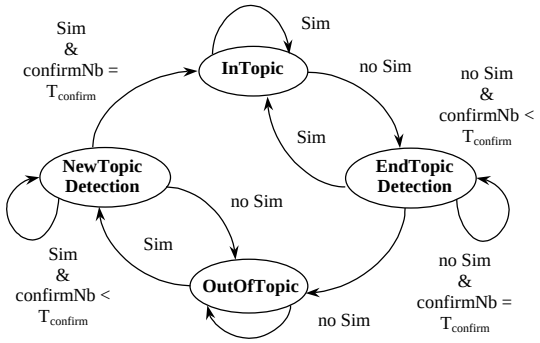

Figure 2 – Automaton for topic shift detection

The processing of a segment starts with the *OutOfTopic* state, after the end of the previous segment or at the beginning of the text. As soon as the context of the focus window is stable enough between two successive positions, TOPICOLL enters into the *NewTopicDetection* state. The *InTopic* state can then be reached only if the window context is found stable for the next *confirmNb*-1 positions. Otherwise, TOPICOLL assumes that it is a false alarm and returns to the *OutOfTopic* state. The detection of the end of a segment is symmetrical to the detection of its beginning. TOPICOLL goes into the *EndTopicDetection* state as soon as the content of the window context begins to change significantly between two successive positions but the transition towards the *OutOfTopic* state is done only if this change is confirmed for the next *confirmNb*-1 next positions.
This algorithm is completed by a specific mechanism related to the *OutOfTopic* state. When TOPICOLL stays in this state for a too long time (this time is defined as 10 positions of the focus window in our experiments), it assumes that the topic of the current part of text is difficult to characterize by using word recurrence or selection from a collocation network and it creates a new segment that covers all the concerned positions.

4.3 Link detection

The algorithm of TOPICOLL for detecting identity links between segments is closely associated to its algorithm for delimiting segments. When TOPICOLL goes from the *NewTopicDetection* state to the *InTopic* state, it first checks whether the current context of the new segment is similar to one of the contexts of the previous segments. In this case, the similarity between contexts only relies on the similarity measure (see (5)) between their collocation vectors. A specific threshold is used for the decision. If the similarity value exceeds this threshold, the new segment is linked to the corresponding segment and takes the context of this one as its own context. In this way, TOPICOLL assumes that the new segment continues to develop a previous topic. When several segments fulfills the condition for link detection, TOPICOLL selects the one with the highest similarity value.

5 Experiments

5.1 Topic segmentation

For evaluating TOPICOLL about segmentation, we applied it to the "classical" task of discovering boundaries between concatenated texts. TOPICOLL was adapted for aligning boundaries with ends of sentences. We used the probabilistic error metric P_k proposed in (Beeferman et al., 1999) for measuring segmentation accuracy[4]. Recall and precision was computed for the *Le Monde* corpus to compare TOPICOLL with older systems[5]. In this case, the match between a boundary from TOPICOLL and a document break was accepted if the boundary was not farther than 9 plain words.

5.1.1 Le Monde corpus

The evaluation corpus for French was made up of 49 texts, 133 words long on average, from the *Le*

[4] P_k evaluates the probability that a randomly chosen pair of words, separated by k words, is wrongly classified, *i.e.* they are found in the same segment by TOPICOLL while they are actually in different ones (miss of a document break) or they are found in different segments by TOPICOLL while they are actually in the same one (false alarm).

[5] Precision is given by N_t / N_b and recall by N_t / D, with D the number of document breaks, N_b the number of boundaries found by TOPICOLL and N_t the number of boundaries that are document breaks.

Monde newspaper. Results in Tables 1 and 2 are average values computed from 10 different sequences of them. The baseline procedure consisted in randomly choosing a fixed number of sentence ends as boundaries. Its results in Tables 1 and 2 are average values from 1,000 draws.

Systems	Recall	Precision	F1-measure
baseline	0.51	0.28	0.36
SEGCOHLEX	0.68	0.37	0.48
SEGAPSITH	0.92	0.52	0.67
TextTiling	0.72	0.81	0.76
$TOPICOLL_1$	0.86	0.74	0.80
$TOPICOLL_2$	0.86	0.78	0.81
$TOPICOLL_3$	0.66	0.60	0.63

Table 1 – Precision/recall for *Le Monde* corpus

$TOPICOLL_1$ is the system described in section 4. $TOPICOLL_2$ is the same system but without its link detection part. The results of these two variants show that the search for links between segments doesn't significantly debase TOPICOLL's capabilities for segmentation. $TOPICOLL_3$ is a version of TOPICOLL that only relies on word recurrence. SEGCOHLEX and SEGAPSITH are the systems described in (Ferret, 1998) and (Ferret and Grau, 2000). TextTiling is our implementation of Hearst's algorithm with its standard parameters.

Systems	Miss	False alarm	Error
baseline	0.46	0.55	0.50
$TOPICOLL_1$	0.17	0.24	0.21
$TOPICOLL_2$	0.17	0.22	0.20

Table 2 – P_k for *Le Monde* corpus

First, Table 1 shows that TOPICOLL is more accurate when its uses both word recurrence and collocations. Furthermore, it shows that TOPICOLL gets better results than a system that only relies on a collocation network such as SEGCOHLEX. It also gets better results than a system such as TextTiling that is based on word recurrence and as TOPICOLL, works with a local context. Thus, Table 1 confirms the fact reported in (Jobbins and Evett, 1998) that using collocations together with word recurrence is an interesting approach for text segmentation. Moreover, TOPICOLL is more accurate than a system such as SEGAPSITH that depends on topic representations. Its accuracy is also slightly higher than the one reported in (Bigi et al., 1998) for a system that uses topic representations in a probabilistic way: 0.75 as precision, 0.80 as recall and 0.77 as f1-measure got on a corpus made of *Le Monde*'s articles too.

5.1.2 C99 corpus

For English, we used the artificial corpus built by Choi (Choi, 2000) for comparing several segmentation systems. This corpus is made up of 700 samples defined as follows: "A sample is a concatenation of ten text segments. A segment is the first *n* sentences of a randomly selected document for the Brown corpus". Each column of Table 3 states for an interval of values for *n*.

Systems	3-11	3-5	6-8	9-11
baseline	0.45	0.38	0.39	0.36
CWM	0.09	0.10	0.07	0.05
U00	0.10	0.09	0.07	0.05
C99	0.12	0.11	0.09	0.09
DotPlot	0.18	0.20	0.15	0.12
Segmenter	0.36	0.23	0.33	0.43
TextTiling	0.46	0.44	0.43	0.48
$TOPICOLL_1$	0.30	0.28	0.27	0.34
$TOPICOLL_2$	0.31	0.28	0.28	0.34

Table 3 – P_k for *C99* corpus

The first seven lines of Table 3 results from Choi's experiments (Choi, 2001). The baseline is a procedure that partitions a document into 10 segments of equal length. CWM is described in (Choi, 2001), U00 in (Utiyama and Isahara, 2001), C99 in (Choi, 2000), DotPlot in (Reynar, 1998) and Segmenter in (Kan et al., 1998).

Table 3 confirms first that the link detection part of TOPICOLL doesn't debase its segmentation capabilities. It also shows that TOPICOLL's results on this corpus are significantly lower than its results on the *Le Monde* corpus. This is partially due to our collocation network for English: its density, *i.e.* the ratio between the size of its vocabulary and its number of collocations, is 30% lower than the density of the network for French, which has certainly a significant effect. Table 3 also shows that TOPICOLL has worse results than systems such as CWM, U00, C99 or DotPlot. This can be explained by the fact that TOPICOLL only works with a local

context whereas these systems rely on the whole text they process. As a consequence, they have a global view on texts but are more costly than TOPICOLL from an algorithmic viewpoint. Moreover, link detection makes TOPICOLL functionally richer than they are.

5.2 Global evaluation

The global evaluation of a system such as TOPICOLL faces a problem: a reference for link detection is relative to a reference for segmentation. Hence, mapping it onto the segments delimited by a system to evaluate is not straightforward. To bypass this problem, we chose an approach close the one adopted in TDT for the link detection task: we evaluated the probability of an error in classifying each couple of positions in a text as being part of the same topic (Cp_{same}) or belonging to different topics (Cp_{diff}). A miss is detected if a couple is found about different topics while they are about the same topic and a false alarm corresponds to the complementary case.

Systems	Miss	False alarm	Error
baseline	0.85	0.06	0.45
TOPICOLL	0.73	0.01	0.37

Table 4 – Error rates for *Le Monde* corpus

As the number of Cp_{diff} couples is generally much larger than the number of Cp_{same} couples, we randomly selected a number of Cp_{diff} couples equal to the number of Cp_{same} couples in order to have a large range of possible values. Table 4 shows the results of TOPICOLL for the considered measure and compares them to a baseline procedure that randomly set a fixed number of boundaries and a fixed number of links between the delimited segments. This measure is a first proposition that should certainly be improved, especially for balancing more soundly misses and false alarms.

6 Conclusion

We have proposed a method for achieving both topic segmentation and link detection by using collocations together with word recurrence in texts. Its evaluation showed the soundness of this approach for working with a local context. We plan to extend it to methods that rely on the whole text they process. We also aim at extending the evaluation part of this work by improving the global measure we have proposed and by comparing our results to human judgments.

References

Beeferman D., Berger A. and Lafferty J. (1999) *Statistical Models for Text Segmentation*, Machine Learning, 34/1, pp. 177–210.

Bigi B., de Mori R., El-Bèze M. and Spriet T. (1998) *Detecting topic shifts using a cache memory*, 5th International Conference on Spoken Language Processing, pp. 2331–2334.

Church K. W. and Hanks P. (1990) *Word Association Norms, Mutual Information, And Lexicography*. Computational Linguistics, 16/1, pp. 22–29.

Choi F., Wiemer-Hastings P. and Moore J. (2001) *Latent Semantic Analysis for Text Segmentation*, NAACL'01, pp. 109–117.

Choi F. (2000) *Advances in domain independent linear text segmentation*, NAACL'00, pp. 26–33.

Ferret O. and Grau B. (2000) *A Topic Segmentation of Texts based on Semantic Domains*, ECAI 2000, pp. 426–430.

Ferret O. (1998) *How to thematically segment texts by using lexical cohesion?*, ACL-COLING'98, pp. 1481–1483.

Fiscus J., Doddington G., Garofolo J. and Martin A. (1999) *NIST's 1998 Topic Detection and Tracking Evaluation*, DARPA Broadcast News Workshop.

Hearst M. (1997) *TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages*, Computational Linguistics, 23/1, pp. 33–64.

Jobbins A. and Evett L. (1998) *Text Segmentation Using Reiteration and Collocation*, ACL-COLING'98, pp. 614–618.

Kan M-Y., Klavans J. and McKeown K. (1998) *Linear segmentation and segment significance*, 6th Workshop on Very Large Corpora, pp. 197–205.

Kaufmann S. (1999) *Cohesion and Collocation: Using Context Vectors in Text Segmentation*, ACL'99, pp. 591–595.

Kozima H. (1993) *Text Segmentation Based on Similarity between Words*, ACL'93, pp. 286–288.

Passonneau R. and Litman D. (1997) *Discourse Segmentation by Human and Automated Means*, Computational Linguistics, 23/1, pp. 103–139.

Reynar R. (1998) *Topic segmentation: Algorithms and applications*, Ph.D. thesis, Computer and Information Science, University of Pennsylvania.

Utiyama M. and Isahara H. (2001) *A Statistical Model for Domain-Independent Text Segmentation*, ACL'2001, pp. 491–498.