Evaluation of Lexical Cohesion Algorithms for Arabic Topic Segmentation

HARRAG Fouzi [1], HAMDI-CHERIF Aboubekeur [2], BENMOHAMMED Mohamed [3]
[1] Computer Science Department, Farhat ABBAS University,
Setif, 19000, Algeria
[2] Computer College, Qassim university,
Buraydah, 51452, Saudi Arabia
[3] Computer Science Department, Mentouri University,
Constantine, 25000, Algeria
[1]hfouzi2001@yahoo.fr
[2]elhamdi62@gmail.com
[3]mibn@yahoo.fr

Abstract: The need of having a topic segmentation system for Arabic text is due essentially to improve the functionalities of Arabic Information Retrieval (AIR). Topic segmentation of texts has been used to improve the accuracy of the subsequent processes such as question answering and information retrieval. In this paper we present the implementation and the evaluation of two algorithms for Arabic text segmentation which are Text-Tilling and C99. We compare the quality of the outputs of the two algorithms and we evaluate the relative performance of Text Tiling algorithm with respect to another cohesion based segmenter: C99 algorithm using the classical Recall/Precision evaluation metrics and the recently introduced Reader Judgment method.

Keywords: Topic Segmentation, Text Tiling algorithm, C99 algorithm, Evaluation, Arabic Language.

Introduction

Topic segmentation can be defined as the task of breaking documents into topically coherent multi-paragraph subparts. In order to provide solutions to access useful information from the ever-growing number of documents on the web, such technologies are crucial as people who search for information are now submerged with unmanageable quantities of text data and most of the time cannot find what they are looking for, as they can only deal with conveniently-sized packages of information. For that purpose, topic segmentation has extensively been used in information retrieval and text summarization.

Text in long document or that obtained from continuous text streams needs to be separated into topically coherent units in order to enable effective querying, analysis and usage. Topic segmentation is a new technique for improving access to information dividing lengthy documents into topically coherent sections. In information retrieval for example, having topically segmented documents can result in the retrieval of short relevant text segments that directly correspond to a user's query instead of long documents examined by user carefully in order to find the object of his interest. Having topically segmented documents also benefits the task of text summarization as a better summary can be obtained from the various segments constituting a document [7]. While extensive research has targeted this technique in English, few have studied it in other languages and almost no one except [7] and [12], has addressed it for Arabic language which focused our interest and push us trying to adopt the two text segmentation algorithms for such language.

This paper is organized as follows: Section 2 presents related work; Section 3 presents an overview of the implemented approaches; results and their discussion are reported in Section 4; finally Section 5 concludes the paper.

1. Previous Work

Approaches that address the problem of topic segmentation can be classified in knowledge-based approaches or word-based approaches. Knowledge-based systems as [11] require an extensive manual knowledge engineering effort to create the knowledge base (semantic network and/or frames) and this is only possible in very limited and well-known domains. To overcome this limitation, and to process a large amount of texts, word-based approaches have been developed. [13] and [20] make use of the word distribution in a text to find a thematic segmentation. These works are well adapted to technical or scientific texts characterized by a specific vocabulary. To process narrative or expository texts such as newspaper articles, [17] and [22] approaches are based on lexical cohesion computed from a lexical network. These methods depend on the presence of the text vocabulary inside their network. So, to avoid any restriction about domains in such kinds of texts, [20] presented a mixed method that augments his system based on word distribution, by using knowledge represented by a lexical co-occurrence network automatically built from a corpus. By making some experiments with these two latter systems, [8] show that adding lexical knowledge isn't sufficient by its own to have an all-purpose method, able to process either technical texts or narratives. They then propose some solutions to choose the most suitable method.

Other Existing approaches of text segmentation can fall into two main groups: lexical cohesion based approaches and feature based approaches. Lexical cohesion

based approaches depend on the tendency of topic units to hang together. Approaches to measure this type of cohesion can be further divided into two categories: similarity based approaches where patterns of syntactic repetitions are used to indicate cohesion, and lexical chaining based approaches where other aspects of lexical cohesion (like relationships between terms) are also analyzed. PLSA is an example of similarity based approach [3]. This system uses the Probabilistic Latent Semantic Analysis model along with the clarity-based similarity metric to detect boundaries. The work measures similarity using the probability distribution of words calculated using the PLSA model instead of using term counts.

The application of Lexical chaining based approaches to text segmentation was first attempted in [15] and [19]. In these works, segmenting a single document to its sub topics was the major goal. Recently, lexical chaining has also been used in news story segmentation [9] [26]. In [9] the lexical chaining based approach is used in conjunction with the similarity based approach. In this work, lexical cohesion between two adjacent blocks is determined by computing the cosine similarity between the two blocks through analyzing the lexical chains that overlap with the two blocks instead of using word counts. Evaluation of this work was based on the Topic Detection and Tracking (TDT) corpora. Work in [26] uses the lexical chaining technique for determining distinct news stories in spoken and written broadcast news streams. The work analyzes the cohesion in text by examining term repetitions and three other basic types of cohesion (synonymy, generalization/specialization and part-whole/whole-part relationships) provided by the WordNet online thesaurus [21]. The second main category in text segmentation is feature based approaches in which features like cue phrases, full proper nouns and named entities are used to detect boundaries between topics. An example of a system that uses that approach is presented in [16]. Feature based approaches can be domain dependent (as in news transcripts), if they depend on very specific domain features. Lexical cohesion can also be added as a feature in the feature based approach as exemplified by work presented in [2][25].

2. Implemented Approaches

In this section two text segmentation systems are described, these systems are Text Tilling [13] and C99 [5]. The two systems are based on lexical cohesion. Text Tiling algorithm uses the cosine similarity metric between term vectors to measure the cohesion strength between adjacent blocks. The C99 algorithm also uses the cosine similarity metric to determine similarities among sentences and then projects these graphically and applies image-processing techniques to determine topic boundaries.

2.1. Preprocessing

The preprocessing stage processes the stream for analysis by removing tags, punctuation and transforming terms into stems. First we have to build the blocks, called tokens, of text. The input text is merely a sequence of characters prior to preprocessing. It is the responsibility of the preprocessor to break the sequence into semantic units in the tokenization step. These units can either be simple words such as the words program and creation, or multi-word phrases such as The United States (as opposed to United and States).

2.2. The Text Tiling Algorithm

The Text Tiling algorithm, for discovering subtopic structure using term repetition, has three main parts [13]:

- Tokenization
- Similarity Determination
- Boundary Identification

Tokenization refers to the division of the input text into individual lexical units. For both versions of the algorithm, the text is subdivided into pseudo sentences of a pre-defined size w (In practice, setting w to 20 tokens) .The morphologically-analyzed token is stored in a table along with a record of the token-sequence number it occurred in, and how frequently it appeared in the token-sequence. A record is also kept for the locations of the paragraph breaks within the text. Closed-class and other very frequent words are eliminated from the analysis.

After tokenization, the next step is the comparison of adjacent pairs of blocks of token-sequences for overall lexical similarity. Another important parameter for the algorithm is the block size: the number of token-sequences that are grouped together into a block to be compared against an adjacent group of token-sequences. This value, labeled k, varies slightly from text to text. In practice, a value of k=6 works well for many texts. Similarity values are computed for every token-sequence gap number; that is, a score is assigned to token-sequence gap i corresponding to how similar the token-sequences from i-k to i are to the token-sequences from i+1 to i+k+1. Note that this moving window approach means that each token-sequence appears in 2k similarity computations. Similarity between blocks is calculated by a cosine measure (see equation. 1): given two text blocks b1 and b2 each with k token-sequences,

$$\text{Score}(i) = \frac{\sum_{t} W_{t,b1} W_{t,b2}}{\sqrt{\sum_{t} W_{t,b1}^{2} \sum_{t} W_{t,b2}^{2}}} \qquad (1)$$

Where t ranges over all the terms that have been registered during the tokenization step, and Wt,bi is the weight assigned to term t in block b1. Thus if the similarity score between two blocks is high, then the blocks have many terms in common. This formula yields a score between 0 and 1, inclusive.

The token-sequence gap numbers are ordered according to how steeply the slopes of the plot are to either side of the token-sequence gap, rather than by their absolute similarity score. For a given token-sequence gap i, the algorithm looks at the scores of the token-sequence gaps to the left of i as long as their values are increasing. When the values to the left peak out, the difference between the score at the peak and the score at i is recorded. The same procedure takes place with the token-sequence gaps to the right of i; their scores are examined as long as they continue to rise.

The relative height of the peak to the right of i is added to the relative height of the peak to the left. (A gap occurring at a peak will have a score of zero since neither of its neighbors is higher than it.) These new scores, called depth scores, corresponding to how sharp a change occurs on both sides of the token-sequence gap, are then sorted. Segment boundaries are assigned to the token-sequence gaps with the largest corresponding scores, adjusted as necessary to correspond to true paragraph breaks. A proviso check is done that prevents assignment of very close adjacent segment boundaries. Currently there must be at least three intervening token-sequences between boundaries. This helps control for the fact that many texts have spurious header information and single-sentence paragraphs.

The algorithm must determine how many segments to assign to a document, since every paragraph is a potential segment boundary. A cutoff based on a particular valley depth is similarly problematic.

2.3. The C99 Algorithm

This segmentation algorithm takes a list of tokenized sentences as input [5]. A tokenizer [10] and a sentence boundary disambiguation algorithm [23] may be used to convert a plain text document into the acceptable input format. Step of Similarity measure start after removing the punctuation and uninformative words from each sentence using a simple regular expression and a stop word list. A stemming algorithm [6] is then applied to the remaining tokens to obtain the word stems. A

dictionary of word stem frequencies is constructed for each sentence. This is represented as a vector of frequency counts.

Let fi,j denote the frequency of word j in sentence i. The similarity between a pair of sentences x, y is computed using the cosine measure as shown in equation. 2. This is applied to all sentence pairs to generate a similarity matrix.

$$\text{Sim}(x, y) = \frac{\sum_j f_{x,j} * f_{y,j}}{\sqrt{\sum_j f_{x,j}^2 * \sum_j f_{y,j}^2}} \qquad (2)$$

Figure 1 shows an example of a similarity matrix. High similarity values are represented by bright pixels. The bottom-left and top-right pixels show the self-similarity for the first and last sentences, respectively. Note that the matrix is symmetric and contains bright square regions along the diagonal. These regions represent cohesive text segments.

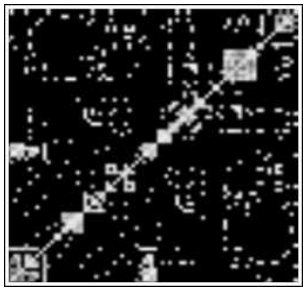

Figure 1. An example of similarity matrix with 11*11 rank matrix

Each value in the similarity matrix is replaced by its rank in the local region. The rank is the number of neighboring elements with a lower similarity value. For segmentation, a 11x11 rank mask is generally used. The output is expressed as a ratio r (see equation. 3), to circumvent normalization problems (consider the cases when the rank mask isn't contained in the image).

$$r = \frac{\#\text{ of elements with a lower value}}{\#\text{ of elements examined}} \qquad (3)$$

The final process determines the location of the topic boundaries. The method is based on Reynar's maximization algorithm [20][25]. A text segment is defined by two sentences i,j (inclusive).

This is represented as a square region along the diagonal of the rank matrix. Let Si,j denotes the sum of the rank values in a segment and the inside area is given by equation. 4.

$$a_{i,j} = (j - i + b) \qquad (4)$$

B = {bl,...,bm} is a list of m (coherent text segments, sk and ak refers to the sum of rank and area of segment k in B. D is the inside density of B (see equation. 5).

$$D = \frac{\sum_{k=1}^{m} s_k}{\sum_{k=1}^{m} a_k} \qquad (5)$$

To initialize the process, the entire document is placed in B as one coherent text segment. Each step of the process splits one of the segments in B. The split point is a potential boundary which maximizes D. The number of segments to generate, m, is determined automatically. For a document with b potential boundaries, b steps of divisive clustering are generated.

3. Results and Discussion

3.1. Evaluation Metrics

There are several ways to evaluate a segmentation algorithm:

- By Comparison with human judgments: there is no segmented corpus of sufficient size available for this task but only a propositions to build such corpus and to assess the quality of human judgments [4][13][24].

- By comparison to marks deposed in the text by the reader (this method is unreliable because any segmentation is subjective [24], the position of segmentation marks depends on the point of view of the reader);

- By Comparison to “some” marks to find in the text (for example: boundaries between documents of a corpus).

- Through its impact on a particular task like information retrieval (functional evaluation).

The results of the evaluation of each algorithm are shown in the following section.

3.2. The Arabic texts Segmentation Test Corpora

The analyzed segmentation systems were evaluated using a set of five Arabic texts. We compare the obtained results with the judgments of a group of readers who did a manual segmentation. We based our evaluation on seven readers' judgments. After reading, each reader makes a manual segmentation on the five texts. The texts can fall on two categories: literature and medicine. The average length of the texts used for this evaluation is between 600 and 2000 words. The readers are simply invited to define the paragraphs in which there is a topic shift. This operation remains subjective for every reader.

3.3. Method of Reader Judgments

Figure 2 shows the boundaries made by the seven readers on the texts. This diagram helps us to illustrate the general trends of the reader's evaluation, and also to show where and how often they agree or disagree a boundary. For example, all readers except the fourth marked a boundary in paragraph 7. This reader disagrees with other and marked the boundary in paragraph 10. The Boundaries where readers are all in agreement are: {12, 20, 22, 31, 33, 37, 38, and 50}, Readers are in disagreement for the following boundaries: {1, 15, 18, 41.43, 44 and 45}, examples of agreement and disagreement are shown by a top and bottom arrows respectively in the Figure 4.

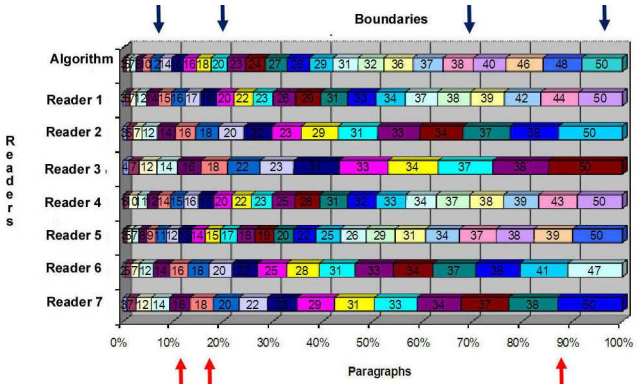

Figure 2 : Boundaries of Readers versus Algorithm.

According to [24], if four of seven readers mark the boundary at the same position, the segmentation is good. [18] Has shown that three readers are considered sufficient to classify this boundary as "main boundary". [4] and [14] Specify the importance of taking into account the expected and unexpected agreement by calculating whether readers agree significantly. To this end, they advise to use the *kappa coefficient (K)*. According to [4], *K* measures by the paired agreements among a group of readers making judgments categories, it is calculated according to

equation. 6:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \qquad (6)$$

Where P(A) is the proportion of times that readers agree and P(E) is the proportion of times we expect them to agree by chance. The coefficient can be calculated by paired comparisons against an expert or comparing to a decision of a group. [4] also states that if K>0.8, this indicates that the segmentation is good, and if K>0.67 and K<0.8 this can provide acceptable experimental conclusions. The coefficients found by [14] have extended from 0.43 to 0.68 for three readers, and those found by [4] are extended from 0.65 to 0.90 for four readers segmenting sentences.

In our evaluation, we set that three judgments in agreement are acceptable to take the boundary as correct. From Figure 4, acceptable boundaries are: {1, 3, 5, 7, 12, 14, 15, 16, 18, 20, 22, 23, 29, 31, 33, 34, 37, 38 and 50}. We calculate the Kappa coefficient as shown in Table 1. The comparison of our results for "K" on Arabic corpus with those obtained by Hearst [13] from the application of the Text Tiling algorithm on an English corpus "K(H)" has shown that our segmentation is acceptable.

P(A)	P(E)	K	K(H)	Remark
0.7894	0.2106	**0.7332**	0.647	**Acceptable**

Table 1 : Results of calculating Kappa coefficient

3.4. Method of Recall / Precision:

In the following experiments, the two standards recall and precision, classically used in information retrieval, detailed in [1], were often employed to evaluate segmentation algorithms. In the context of topic segmentation, precision is defined as (equation 7):

$$\boldsymbol{P} = \frac{\text{Number of correctly system detected boundaries}}{\text{Total number of system generated boundaries}} \qquad (7)$$

While recall is defined as (equation 8):

$$\boldsymbol{R} = \frac{\text{Number of correctly system detected boundaries}}{\text{Total number of real boundaries}} \qquad (8)$$

The Recall value for Text Tiling gives us a prime example of how traditional IR metrics, precision and recall, fail as informative measures of segmentation performance [11]. Figureure 3 shows precision and recall values for five texts segmented with Text Tiling algorithm. This Figureure shows that Text Tiling's recall values are very low, 0%, 33.34% and 60% respectively, precision values are high, 40%, 66.66% and 100%.

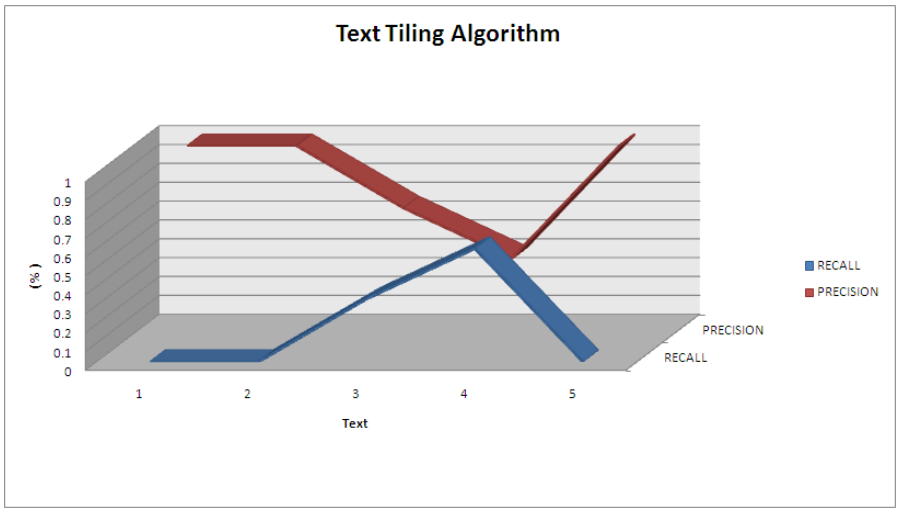

Figure 3: Precision and Recall values for 5 texts segmented with Text Tilling algorithm

However, these values take no account of the fact that Text Tiling is producing 'just' missing boundaries rather than failing to detect them at all. Figureure 4 shows precision and recall values for five texts segmented with C99 algorithm. The interesting observation from this Figureure is that C99 algorithm has a high recall values, 33.34%, 40%, 50% and 100% respectively. Precision values are between 50% and 66.66%.

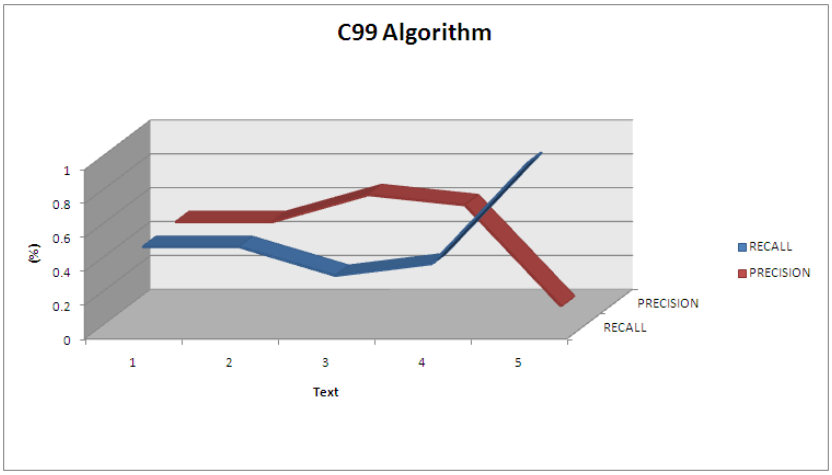

Figure 4 : Precision and Recall values for 5 test texts segmented with C99 algorithm

Table 2 shows the result of comparison between the two algorithms. Text Tiling has the best value on precision; it passes 84% but it has the worst value on recall 15.79%. C99 has the worst value on precision 45.40% but it has the best value on recall; it passes 54%.

Segmentation	**Recall**	**Precision**
Human judges	18.66%	81.33%
Texttilling	15.79%	**84.27%**
C99	**54.60%**	45.40%

Table 3 : Comparative between the two algorithms.

TextTiling and C99 seem to have difficulties to adapt themselves with the number of boundaries to retrieve; the length of the text has a great impact on their number of detected boundaries. Figure 5 shows that TextTiling seems to be more efficient to Arabic texts.

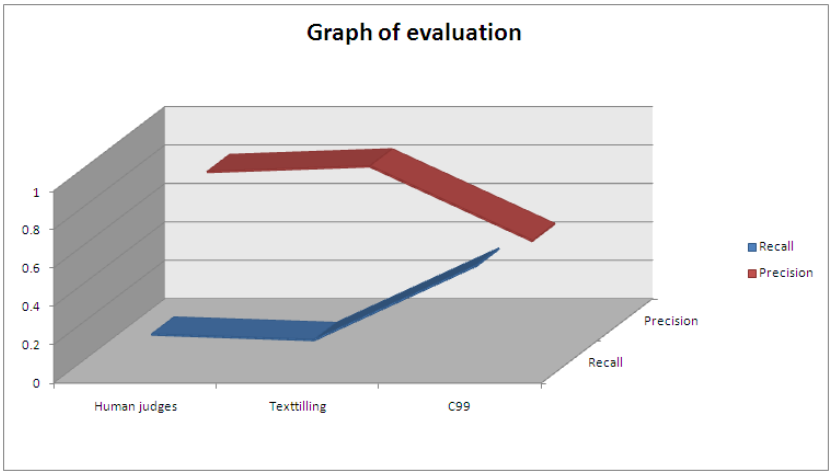

Figure 5 : Evaluation of algorithms with Human judges

Conclusion

In this paper a comparative analysis of two different text segmentation algorithms on Arabic texts is presented. To assess how well each algorithm works on Arabic corpus, each one was applied on an Arabic texts dataset and the results were compared. We confirmed in this paper that segmentation task is hard to evaluate because the objective can vary. Globally C99 algorithm looks to be more efficient. To go further in the experimentation, we should try a new algorithm mixing supervised method with unsupervised, and make new comparisons between statistic and symbolic methods. Eventually, our work shows that with only little improvements, existing algorithms for segmenting English texts are efficient on Arabic texts.

Références

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

[2] D. Beeferman, A. Berger, and J. Lafferty, "*Statistical models for text segmentation*," Machine Learning, vol. 34, pp. 177 - 210, 1999.

[3] T. Brants, F. Chen, and I. Tsochantaridis, "*Topic-based document segmentation with probabilistic latent semantic analysis*," presented at CIKM, McLean, Virginia, USA, 2002.

[4] J. Carletta. "Assessing agreement on classification tasks: The kappa statistic". Computational Linguistics, 22(2):249-254. 1996.

[5] F. Choi, "*Advances in domain independent linear text segmentation*," presented at the first conference on North American chapter of the Association for Computational Linguistics (NAACL), Seattle, Washington, 2000.

[6] K. Darwish, "*Building a Shallow Arabic Morphological Analyzer in One Day*," Proceedings of the workshop on Computational Approaches to Semitic Language, in the 40th Annual Meeting of the Association for the Computational Linguistics, (ACL-02), pp. 47 - 54. 2002.

[7] M. A. El-Shayeb, S. R. El-Beltagy and A. Rafea, "Comparative Analysis of Different Text Segmentation Algorithms on Arabic News Stories," *Proc. IEEE International Conference on Information Reuse and Integration*, pp. 441 - 446, Aug, 2007.

[8] O. Ferrat, B. Grau and N. Masson, "*Thematic segmentation of texts: two methods for two kinds of texts*," In Proceedings of the 36th Annual Meeting of the ACL, 1998.

[9] M. Galley, K. McKeown, E. Fosler-lussier, and H. Jing. Discourse segmentation of multi-party conversation. In: Proceedings of the 41st Annual Meeting of ACL, Sapporo, Japan, 2003.

[10] G. Grefenstette, and P. Tapanainen. What is a word, what is a sentence? Problems of tokenization. In: Proceedings of the 3rd Conference on Computational Lexicography and Text Research (COMPLEX-94), Budapest, Hungary, 1994.

[11] B. J. Grosz and C. L. Sidner, "*Attention, Intentions and the Structure of Discourse*," Computational Linguistics, vol. 12, pp. 175 - 204, 1986.

[12] A. Hasnah, "*Full Text Processing and Retrieval: Weight Ranking Text Structuring, and Passage Retrieval for Arabic Documents*," Ph.D. thesis, Illinois Institute of Technology. 1996.

[13] M. A. Hearst, "*TextTiling: Segmenting text into multiparagraph subtopic passages*," Computational Linguistics, vol. 23, pp. 33 - 64, 1997.

[14] A. Isard and J. Carletta "Replicability of transaction and action coding in the map task corpus". In Johanna Moore and Marilyn Walker, editors, Empirical Methods in Discourse: Interpretation & Generation, AAAI Technical Report SS-95~06. AAAI Press, Menlo Park, CA. 1995.

[15] M. Y. Kan, J. L. Klavans, and K. R. McKeown, "*Linear segmentation and segment relevance*," presented at the International Workshop of Very Large Corpora (WVLC 6), Montreal, 1999.

[16] D. Kauchak and F. Chen, "*Feature-based segmentation of narrative documents*," presented at the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing, Ann Arbor, MI, USA, 2005.

[17] H. Kozima, "*Text Segmentation Based on Similarity between Words*," In Proceedings of ACL'93, pp. 286 - 288, Ohio, Japan, 1993.

[18] D. J. Litman and R. J. Passonneau. “Combining multiple knowledge sources for discourse segmentation”. In Proceedings of the 33rd Meeting of Association for Computational Linguistics., pages 108-115, June. 1993.

[19] O. Manabu and H. Takeo, “*Word sense disambiguation and text segmentation based on lexical cohesion*,” presented at The International Conference on Computational Linguistics, Kyoto, Japan, 1994.

[20] N. Masson, “*An Automatic Method for Document Structuring*,” In Proceedings of the 18th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, 1995.

[21] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, “*Five papers on Wordnet*,” Cognitive Science Laboratory, Technical report 1990.

[22] J. Morris and G. Hirst, “*Lexical cohesion computed by thesaurus relations as an indicator of the structure of text*,” Computational Linguistics, vol. 17(1), pp. 21 - 48, 1991.

[23] D.D. Palmer and M. A. Hearst, “*Adaptive sentence boundary disambiguation*,” In Proceedings of the 4th Conference on Applied Natural Language Processing, Stuttgart, Germany, October. 1994.

[24] J. R. Passonneau and D. J. Litman. “Intention-based segmentation: Human reliability and correlation with linguistic cues”. In Proceedings of the 31st Annual Meeting, pages 148-155. 1993.

[25] J. Reynar, “*Topic Segmentation: Algorithms and Application*,” Ph.D. thesis, Computer and Information Science. University of Pennsylvania, Pennsylvania, USA, 1998.

[26] N. Stokes, J. Carthy, and A. F. Smeaton, “*SeLeCT: a lexical cohesion based news story segmentation system*,” AI Communications, vol. 17, pp. 3 - 12, 2004.