Comparative Analysis of Different Text Segmentation Algorithms on Arabic News Stories

Michael A. El-Shayeb & Samhaa R. El-Beltagy
Computer Science Department,
Faculty of Computers and Information,
Cairo University, Giza, Egypt
mikeazmy@yahoo.com, samhaa@computer.org

Ahmed Rafea
Computer Science Department,
American University in Cairo
Cairo, Egypt
rafea@aucegypt.edu

Abstract

The task of text segmentation represents an important step in many applications and while much work has been carried out to address this task for the English language, work on text segmentation for other languages is still lagging behind. In this paper a comparative analysis of three different text segmentation algorithms on Arabic news stories is presented. To assess how well each algorithm works on Arabic news stories, each was applied on an Arabic Reuters news story dataset and the results were compared. The work in this paper also describes a combination of two of these algorithms that was found to produce better results than any of the presented individual algorithms. It also presents a set of error reduction filters that were found to significantly reduce segmentation errors in the detection of borders in Arabic based news stories.

1. Introduction

Text in long documents or that obtained from continuous text streams needs to be separated into topically coherent units in order to enable effective querying, analysis, and usage. 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 which the user has to examine carefully in order to find the object of his/her 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 [17]. Recently, unsegmented or continuous news streams have also become an important area where text segmentation can be applied since the success of tasks such as topic tracking depends heavily on the accuracy of the detection of distinct news stories.

While extensive research has targeted the problem of determining boundaries in English news streams, few have studied the problem in other languages and almost no one has addressed it for the Arabic language. In this paper three existing text segmentation algorithms (developed for English documents) are implemented and adopted for Arabic and a comparative analysis between them on Arabic news stories is presented. The paper also presents a combination between two of these, and shows that combining them can produce better results. All three presented systems are based on measuring the lexical cohesion between textual units.

The presented algorithms were evaluated using an Arabic Reuters newswire dataset which is composed of one thousand concatenated news stories. The evaluation experiments show that adding some error reduction features to the presented algorithms is very effective in improving performance.

This paper is organized as follows: section 2 presents related work, section 3 presents an overview of the analyzed systems; results and discussion are reported in section 4 and finally section 5 concludes the paper.

2. Related Work

Existing approaches to text segmentation 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.

An example of the former approach is the TextTiling system [13] which uses the cosine similarity metric between term vectors to measure the cohesion strength between adjacent blocks. Another example is that of the

C99 [3] algorithm which 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. Another system that utilizes a similarity based approach is presented in [18]. This system uses the probabilistic latent semantic analysis (PLSA) 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 [16] [9] and then in [7]. 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 [14] [12]. Work in [14] 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 [4]. In [12] 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.

The second main category in text segmentation is that of feature based approaches in where 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 [2]. 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 [1] and [6].

3. Implemented Approaches

In this section three text segmentation systems are described. These systems are SeLeCT [15], LCseg [12], and TextTiling [13]. All of the three systems are based on lexical cohesion. Four phases are used to describe the three systems:

1. Preprocessing.
2. Similarity Determination.
3. Boundary Identification.
4. Error reduction filtering.

3.1. The Preprocessing Phase

In SeleCT and LCseg, sentences are considered to be the basic textual units on which work can be carried out. So an important step in the preprocessing phase is to identify sentence boundaries through the use of ".", "?" and "!". Exceptions to this rule include abbreviations.

In TextTiling pseudosentences of a predefined fixed size of 20 tokens rather than actual sentences are used to avoid normalization problems.

In both SeleCT and TextTiling nouns and adjectives need to be extracted from the input sentences. So in the second step of the preprocessing phase in these systems, part-of-speech (POS) tagging is carried out in order to tag required terms and to also determine conjunctions as they are needed in the error reduction phase. Determining full proper nouns is also needed in the error reduction phase, so the third step in the pre-processing phase is the application of a chunk parser. For carrying out steps two and three, the Arabic POS tagger and chunk parsing tool presented in [10] were used. Since LCseg uses all terms not only nouns and adjectives, these two steps are not needed for its basic operation, but they were applied for experimentation with the error reduction filters.

In all three systems, stop words are removed and all remaining words are stemmed. Finally candidate terms are saved with their locations in the text for usage in the next phase of similarity determination. In SeleCT and LCseg, locations in the text are represented by sentence numbers, whereas in TextTiling pseudosentence numbers are used.

3.2. The Similarity Determination Phase

In TextTiling, similarity is computed using the cosine similarity metric between term vectors of adjacent windows of pseudosentences. For each pseudosentence break, the similarity between windows (*A* and *B* each of fixed size *K* = 6 pseudosentences) is computed with:

$$\cos ine\left(A,B\right)=\frac{\sum_t w_{t,A}\cdot w_{t,B}}{\sqrt{\sum_t w_{t,A}^2\sum_t w_{t,B}^2}}$$

Where t is all valid terms and $w_{t,\Gamma}$ is the frequency of term t within window $\Gamma\in\{A,B\}$. Thus a high similarity score means that the two windows have many terms in common. Cosine similarity values are then smoothed using a moving average filter.

SeleCT and LCseg depend on the lexical chaining [5] technique to connect sentences that are related to each other through the detection of the existence of semantic relationships between vocabularies appearing within these sentences. Specifically, a lexical chain is defined as a cluster of semantically related terms. Each chain is tagged with the locations of the sentences where it begins and ends. Work on text segmentation using lexical chaining has almost always used an external resource such as Wordnet, to capture relationships between terms in the text. However, recently two systems [12][14] have shown that the best performance achieved for segmenting text with a lexical chainer was obtained when only patterns of repetitions were used. Based on these findings, only repletion patterns of terms across sentences are used in the chaining process thus ignoring synonymy and other semantic relationships. Starting from the first sentence in the input text, whenever a term (singular or compound) is found to repeat in adjacent sentences, a chain for that term is created. The chain start tag is always assigned to the sentence location where the term first occurs. The chain is said to continue until a gap that exceeds an allowable gap length exists between the last occurrence of the term in the chain and its next occurrence in the text, or if the last term in the chain is also the last term in the document. The best allowable gap length g was determined experimentally as 11 sentences [12]. In case of an occurrence of a gap, a new chain is created for the same term when it appears at a later point in the document. The reason behind cutting the chain when a gap occurs between two occurrences of the same term is because the probability for these two occurrences belonging to the same topical entity decreases as the distance between the two occurrences increases. So, termination of a chain in such a case avoids the creation of weak chains.

In addition to this basic process, LCseg also provides a weight to each created chain based on the assumption that chains that contain more repeated terms and span shorter number of sentences are highly indicative of a lexical cohesion structure. The chain weight is computed with:

$$weight(C_i) = freq(t_i) \cdot \log\left(\frac{L}{L_i}\right)$$

Where t_i is the term of the underling chain, L_i is the length of the chain in sentences, and L is the length of the text.

LCseg then uses the cosine similarity metric (as in TextTiling) to compute the similarity between each two adjacent windows (of fixed size K =2 sentences). Instead of using term frequencies within the two windows to compute similarity. LCseg uses lexical chains that overlap with the two windows. The similarity between windows A and B is computed with:

$$\cos ine\,(A,B) = \frac{\sum_i w_{i,A} \cdot w_{i,B}}{\sqrt{\sum_i w_{i,A}^2 \sum_i w_{i,B}^2}}$$

Where

$$w_{i,\Gamma} = \begin{cases} weight(C_i) & \text{if } R_i \text{ overlaps } \Gamma \in \{\mathrm{A},\mathrm{B}\} \\ 0 & \text{otherwise} \end{cases}$$

Cosine similarity values are then smoothed using a moving average filter.

SeleCT, on the other hand, uses a different way to indicate similarity between sentences in the text. It depends on the hypothesis that the existence of a low number of chains that end at a sentence ***n*** and a low number of chains that begin at the following sentence ***n+1*** is an indicator of the existence of lexical cohesion between the two sentences. Conversely, the existence of a high number of chains that end at a sentence ***n*** and a high number of chains that begin at the following sentence ***n+1*** is an indicator of a possible boundary between the text segment ending with sentence ***n*** and the text segment beginning at sentence ***n+1***. So boundary scores are computed for each two adjacent sentences by summing the number of chains that end at sentence ***n*** and the number of chains that begin at sentence ***n+1*** .

3.3. The Boundary Identification Phase

In TextTiling and LCseg, a depth score is computed for each break between two adjacent windows by summing the change between the similarity score at that break and the maximum similarity scores in both sides of this break. High values of these depth scores represent drops in similarity and so they are good indicators of segment boundaries. Mean (μ) and standard deviation (σ) are then computed for these depth scores. A boundary is determined if the depth score exceeds $\mu - \alpha \cdot \sigma$. α is determined experimentally by the two systems as ½. In LCseg, mean and standard deviation are computed only for these depth scores that are higher than a threshold $PLimit = 0.1$ .In TextTiling, when the determined boundary is not located at a sentence break, the boundary is moved to the nearest sentence break. In SeleCT, boundary scores exceeding a threshold are used as indicators of segment boundaries. The threshold is set as the mean of all boundary scores.

3.4. The Error Reduction Filtering Phase

The aim of this phase is to “correct” the news story boundaries that were determined from the previous phase by eliminating noisy boundaries and thus increasing boundary precision. In the context of this work, three cases define noisy boundaries.

In the first case, an identified boundary point only separates a small number of sentences the individual collection of which is unlikely to form a distinct news story. To clean this type of noisy boundary, a distance filter is utilized. This filter removes boundaries that are close to a boundary with a higher boundary strength as has been done in SeleCT and TextTiling. By experimenting in this work, a distance boundary of 80 words was found to yield the best results.

In the second case, a boundary point is followed by a conjunctive sentence, i.e. a sentence that begins with a conjunction. In Arabic news stories, examples of conjunctions include "و", "كذلك", "كما", "فيما", "من جانبه", "من جهة أخري", "الي ذلك" [1], etc. A conjunctive filter is used to eliminate these boundaries as they separate a segment that begins with a sentence related to the previous segment with a conjunction. When this type of filter was used on English stories in SeleCT, it was found to affect the results adversely. However, as will be shown, when applied on Arabic news stories this type of filter has been found to result in significant error reduction. Two complementary methods are proposed in this work to determine conjunctions. The first method makes use of terms that have been tagged as conjunctions through POS tagging performed in the preprocessing phase. The second, uses a list of conjunctions that seem to occur with a high frequency in the beginning of the inner sentences of each news story. The second method is used as a fail safe when POS tagging can not tag a conjunction appropriately or when the conjunction takes the shape of a noun phrase like, "الموقف الذي يقضي"[2].

In the third and last case, a boundary point separates a sentence that has a full proper noun from a sentence containing a short form of this full proper noun. For example, Ludwig van Beethoven is a full proper noun, while Beethoven, is the short form for this full proper noun. Similarly, "الأمين العام للأمم المتحدة كوفي أنان"[3], is a full proper noun which usually only appears in its full form the first time it is referenced, and is then referred to using short proper noun instances such as "أنان"[4]. To detect this kind of occurrence, it is assumed that first appearance of a proper noun is a full appearance of this proper noun. A rule is then used to connect the full source proper noun with its referrers. This rule specifies that if after the appearance of a full proper noun, a term is found which is a substring of the full proper noun after removing the first word in that full proper noun, then it is a reference or a short form of the original full proper noun. A proper noun fi1lter that applies this rule which was proposed by us in [11] was used. Full proper nouns were identified in this work using the chunk parsing tool [10].

3.5. The Proposed System

In this section, ModSeleCT, which combines ideas from both SeleCT and LCseg is described. The main objective of this modified system is to make use of the good features in both of the two systems which are:

1. Chain weights and depth scores that are used in LCseg.
2. The similarity determination method that is used in SeleCT (that replaces the cosine similarity measure of LCSeg).

So, the main difference between ModSeleCT and SeleCT is the way boundaries are determined. Instead of summing chain ends and chain beginnings between each two adjacent sentences and using the mean of these sums as a threshold to determine boundaries, ModSeleCT determines boundaries in three steps:

1. Chain weights are computed for every ending and beginning chain between two adjacent sentences in the same way as LCseg computes chain weights. The sum of the weights rather than their count is then summed to determine boundary strength (this is a variation on SeleCT)
2. A depth score is computed for each break between two adjacent sentences by calculating the change between the boundary strength at this break and the highest boundary strengths in both sides of this break (similar to LCseg).
3. Mean and standard deviation of depth scores are used as a threshold for determining segment boundaries in a similar way to LCseg.

4. Results and Discussion

In this section the results of evaluating the three presented systems as will as the proposed modified system using an Arabic Reuters news story dataset, are presented. In addition, an analysis of the effect of introducing each of the proposed error reduction filters is also provided.

4.1 Evaluation Metrics

In the evaluation experiments, four segmentation evaluation metrics were used. The first two of these are the standard precision and recall metrics. In the context of text segmentation, precision is defined as the number of correctly system detected boundaries divided by the total number of system generated boundaries, while recall is defined as the number of correctly system detected boundaries divided by the total number of actual boundaries in the used dataset. The precision and recall metrics have been criticized by a number of researchers [1] [14] for their failure to take into account near-boundary misses. As a result, Beeferman [1] proposed a new metric P_k, which is a probabilistic evaluation metric that tries to address the inadequacies of precision and recall. However Pevzner and Hearst [8] recently criticized this metric for being biased, because it penalizes false

[1] Translated to English these would be: and, also, in addition, with, for his part, on the other hand, & so.
[2] The translation of this is: 'the position whereby'
[3] Translation: Secretary-General of the United Nations Kofi Annan
[4] Translation: Annan

negatives more than false positives and over penalizes near-misses. Pevzner and Hearst [8] also suggested a new metric called WindowDiff which handles the criticisms of all previous metrics. So our third used metric is the WindowDiff metric which uses a sliding window over the text to measures the difference between the number of hypothesized boundaries and the actual boundaries within the window. An additional metric called RSeg was proposed by us in [11]. The RSeg metric is used to calculate the exact number of correctly extracted news stories (not boundaries) which reflects the accuracy of the text segmentation algorithm. It should be noted however, that the RSeg metric is a very strict measure of accuracy so it should never be used on its own, but rather used in conjunction with other metrics like those presented here.

4.2. The News Story Segmentation Test Corpora

The analyzed segmentation systems were evaluated using an Arabic Reuters corpus. To construct the test dataset, a collection of 1000 randomly selected news stories were obtained form a huge Arabic Reuters dataset and concatenated into one big file. The advantage of using a single file of concatenated news stories as a dataset is that it does not require human intervention for judging topic shifts as a segment in this context refers to a distinct news story. Our dataset is available online for future comparison by any one who develops an Arabic text segmentation system [19].

4.3. Segmentation results

The results of evaluating SeLeCT, LCseg, and TextTiling as well as the proposed system ModSeleCT using the Arabic Reuters dataset are shown in table 1. The results of applying various error filters to all four systems are presented in table 2. To obtain the results shown in table 1, the parameters of the four systems were tuned with the aim of minimizing the WindowDiff error metric which is considered the most reliable metric. To get the optimal performance out of the four systems with respect to WindowDiff metric, parameters were tuned as follows:

For SeleCT, the best performance was achieved with μ+6 (μ = mean). For ModSeleCT the best performance was achieved with $\mu + 0.96 \cdot \sigma$ (σ = standard deviation). For LCseg the best performance was achieved with $\mu + 0.23 \cdot \sigma$, For TextTiling the best performance was achieved with $\mu + 0.27 \cdot \sigma$, and $k = 3$ (window size).

From table 1, it can be observed that ModSeleCT outperforms all the other three systems with respect to all metrics except for precision for which LCseg outperforms ModSeleCT but only by 1.9%.

When applying the error reduction filters to the main algorithms, highest performance was achieved when the systems parameters were tuned to initially maximize the values of recall.

Table 1. Results for segmentation on concatenated Arabic Reuters news stories

	Recall	Precision	Rseg	WD
SeleCT	64.2%	76.5%	35%	0.224
ModSeleCT	69.3%	76.8%	41.1%	0.2
LCseg	54.2%	78.7%	28.8%	0.232
TextTiling	43.1%	54.4%	16.1%	0.273

The reason is that the error filters ultimately increase the values of precision and so high values for both precision and recall are obtained. To examine the effect of the different error reduction filters on the performance of the four systems, the results of the systems before and after using every filter are presented in table 2. To get the optimal performance out of the four systems, their parameters were tuned as follows:

For SeleCT, the best performance was achieved with μ+2 (as a threshold for determining segment boundaries). For ModSeleCT the best performance was achieved with $\mu + 0.02 \cdot \sigma$. For LCseg the best performance was achieved with $\mu - 0.9 \cdot \sigma$. For TextTiling the best performance was achieved with $\mu - 0.8 \cdot \sigma$, and $k = 3$ (window size). It can be observed that the threshold was reduced. The reason of this reduction in is to allow for more boundaries to be hypothesized by the systems and thus increase the chance for detecting more actual boundaries.

Table 2 shows that for the four systems, the use of the distance error filter reduces the WindowDiff error value and increases the number of correctly extracted distinct news stories compared with the results of the systems before using this filter. The table also shows that recall decreases after using the distance filter. This is because some correct boundaries can be falsely removed as the size of the news stories in the used data set vary from very short to very long. Table 2 also shows a significant enhancement in the segmentation results for the four systems when the conjunction error filter is used. By introducing this filter for ModSeleCT for example, the error (indicated by WindowDiff) decreases from 0.553 to 0.087 and the number of correctly extracted news stories increases from 31.2% to 71.4%. The advantage of this filter is that it rarely removes correct boundaries compared with the distance error filter. It can also be deduced from this table that proper noun filtering achieves a slight improvement in approximately all the metrics except of the recall value which is slightly reduced. A possible reason for this result is that the used POS tagger and chunk parser do not always tag proper nouns appropriately which may have led to this very slight improvement when using this filter.

Table 2. Results of applying error filters to all four systems (D= distance filter, C= conjunction filter, P= proper noun filter)

	SeleCT				ModSeleCT				LCseg				TextTiling			
	R%	P%	Rsg%	WD	R%	P%	Rsg%	WD	R%	P%	Rsg%	WD	R%	P%	Rsg%	WD
N	90.6	36.1	27.5	0.734	87.6	41.7	31.2	0.553	82.8	33.3	10.3	0.776	70.9	18.9	8.7	1.3
C	**89.5**	**89.7**	**75.3**	**0.084**	**86.5**	**90.9**	**71.4**	**0.087**	81.5	85.6	57.7	0.138	69.7	74.8	41.8	0.207
D	77	55.4	35.4	0.31	77	53.5	33.4	0.332	67.6	60	34.3	0.244	53.2	31.9	15.5	0.444
P	89.5	39.1	31.1	0.641	86.5	44.6	33.7	0.493	81.9	35.8	14.4	0.691	70	21.2	10.9	1.111
C+D	81.3	94.5	63.4	0.096	79.5	94.4	60.2	0.103	**74.9**	**94.5**	**53.5**	**0.123**	**64**	**81.8**	**37.1**	**0.187**
C+P	88.4	90.9	74.4	0.084	85.4	91.6	70.2	0.09	80.6	86.5	57.3	0.138	68.8	77.2	41.9	0.203
P+D	76.7	58	37.5	0.288	76.3	56.1	35.4	0.309	67.7	61.8	35.4	0.235	53.4	33.8	17.5	0.419
All	80.4	95.3	62.5	0.098	78.5	94.9	59	0.106	74.1	94.5	52.5	0.127	63.4	83.6	36.7	0.189

5. Conclusion

In this paper we have presented a comparative analysis of different text segmentation algorithms on Arabic news stories. The paper has shown that significant enhancement in results has been achieved when combining two of these systems. It has also illustrated that considerable improvement in the results can be achieved for all presented systems when using conjunctions for error reduction filtering. The overall best results obtained were those of the SeleCT system very closely followed by the proposed ModSeleCT, but only after the application of the conjunction filter to both.

6. References

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

[2] 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.

[3] 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.

[4] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Five papers on wordnet," Cognitive Science Laboratory, Technical report 1990.

[5] J. Morris and G. Hirst, "Lexical cohesion by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, vol. 17, pp. 21- 48, 1991.

[6] J. Reynar, "Topic Segmentation: Algorithms and Application," in *Computer and Information Science*. Pennsylvania: UPenn, 1998.

[7] Kan, Min-Yen, 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.

[8] L. Pevzner and M. A. Hearst, "A critique and improvement of an evaluation metric for text segmentation," *Computational Linguistics*, vol. 28, pp. 19 -36, 2002.

[9] M. A. Stairmand, "A Computational Analysis of Lexical Cohesion with Applications in Information Retrieval," in *Department of Language Engineering*. Manchester, England: UMIST, 1996.

[10] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic Tagging of Arabic Text: From raw text to Base Phrase Chunks," presented at HLT/NAACL, Boston, 2004.

[11] M. A. El-Shayeb, S. R. El-Beltagy, and A. Rafea, "An Arabic News Story Segmentation System," Center of Excellence for Data Mining and Computer modeling, Cairo, Egypt, Technical report 2007.

[12] M. Galley, K. McKeown, E. Fosler-lussier, and H. Jing, "Discourse segmentation of multi-party conversation," presented at 41st Annual Meeting of ACL, Sapporo, Japan, 2003.

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

[14] 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.

[15] N. Stokes, J. Carthy, and A. F. Smeaton, "Segmenting broadcast news streams using lexical chains," presented at STarting AI Researchers Symposium, 2002.

[16] 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.

[17] R. Barzilay and M. Elhadad, "Using lexical chains for text summarization," presented at Intelligent Scalable Text Summarization Workshop, Madrid, Spain, 1997.

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

[19] http://www.claes.sci.eg/coe_wm/ar_seg/segdataset.zip