http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (7): 1009-1015, 2008
ISSN 1812-5638

Semantic-Based Segmentation of Arabic Texts

Ameur A. Touir, Hassan Mathkour and Waleed Al-Sanea
Department of Computer Science, King Saud University, Riyadh 11543, Saudi Arabia

Abstract: In this study, we present an automatic technique to help segment the Arabic texts while preserving the semantics. The technique is based on an empirical study on the sentences and clauses connectors. It has evolved from tedious analysis of various Arabic texts and from observations that have been noted over a long period of time. The analysis made it possible to realize the functionality of each connector in terms of separating standalone segments in the Arabic texts. This has lead to a categorization of active and passive connectors. We used the introduced notion of active and passive connectors to develop an algorithm that respects the semantic of the text to identify the segments of a given Arabic text. The algorithm has been implemented and experimented with. Various Arabic essays were segmented using the algorithm and the results were compared to that of manual segmentations performed by linguistic experts. The performance of the algorithm was in line with the manual segmentations that were performed by the linguistic experts.

Key words: Text segmentation, Arabic text processing, computational linguistics, information retrieval

INTRODUCTION

Text segmentation is one of the important units that many language processing applications need (Agichtein, and Ganti, 2004; Al-Sanie *et al*., 2005b; Beeferman *et al*., 1997; Golcher, 2006; Marcu, 2000a, b). For example, applications that process bibliographic texts need text segmentation in order to identify the fields (author, title, reference, date) and store the records in a table. Such cases are manageable, the references are written in standard formats making it possible to track. This is not the case when dealing with free unstructured texts. Applications like information retrieval look for certain facts that might be found in parts of sentences to be extracted. In this case the sentence which is a compound of more than one meaningful part has to be broken into standalone pieces without breaking the semantic of each piece.

The importance of text segmentation and its wide range of applicability in various automation activities ignited research in the segmentation process in different languages such as English, French, Chinese, Polish and Spanish (Wu and Tseng, 1995; Yang and Li, 2005; Mazur, 2005; Sebastian and Costa, 1997). However, limited research has been done in Arabic text segmentation. This may be due to the special flavor and characteristics of the Arabic language. We report here on a technique to segment the Arabic texts while respecting their semantics. The technique is based on the connecting words between sentences and clauses as they are usually used by Arabic writers in known literature. For this purpose, an extensive analysis of various Arabic texts has been conducted. The analysis is to realize the functionalities of connecting words and their variant usages in terms of separating standalone segments in the Arabic texts. This has lead to a categorization of active and passive connectors. The introduced notion of active and passive connectors has been used to develop an algorithm that respects the semantic of the text to identify the segments of a given Arabic text. The algorithm has been implemented. It has been employed in segmenting various Arabic essays. The results were compared to that of manual segmentations performed by linguistic experts. The performance of the algorithm was in line with the manual segmentations that were performed by the linguistic experts.

Different purposes drive the work on text segmentation. Some works are done to segment the texts based on the topic (Beeferman *et al*., 1997; Lamprier *et al*., 2007). In this approach, each part of the text that addresses a certain topic is identified and put into a unique segment. Another approach is to segment the text based on a reference table (Agichtein and Ganti, 2004). The potential segments that fit under the table attributes are identified and then added to the table. It is common that statistical approaches are used in text segmentation (Beeferman *et al*., 1999; Golcher, 2006; Utiyama and Isahara, 2001). In Haouam *et al*. (2003) texts are segmented based on the RST technique then indexed according to their contents, to allow their retrieval in line with some semantic search criteria. Chang and Choi (2005, 2006) attempt to segment texts based on cue phrases. They use the cue phrase probability

Corresponding Author: Ameur A. Touir, P.O. Box 92014, Riyadh 11653, Saudi Arabia

as causality pattern for causality extraction. In (Le Thanh *et al*., 2004), the text is segmented into elementary discourse units based on syntactic information and cue phrases. In Al-Sanie *et al*. (2005a, b) and Mathkour *et al*. (2005), the texts are segmented into sentences, clauses, or clause-like units and then each unit is classified based on its importance as nucleus or satellite (Marcu, 1997, 1999, 2000a; Sparck-Jones, 1999). Cristea *et al*. (2005) utilize segmentation based on discourse structure for the purpose of text summarization. Villatoro-Tello *et al*. (2006) take advantage of the n-gram method to represent sentences using word sequences. Our technique that we pursue here is based on understanding the functionalities of the connectors in the sentences and clauses in Arabic corpus. It introduces the notion of active and passive connectors to draw a methodology to segment Arabic texts while preserving the semantic of its constituents.

BACKGROUND

Punctuation marks had not been known and used maturely in Arabic language until 1912 when Ahmed Zaki Basha first wrote his book Altarqeem wa Alamatuh fi Al-loqah Al-arabiah (Punctuation and its marks in Arabic Language). He defined the rules of writing the different punctuation marks based on his study on French. Punctuation marks had not been known in the Arabic writing culture until the beginning of the last century. Since then, Arabic writers started using the punctuation marks in their writings. However, many writers still do not give them high attention. Because of this, a lot of Arabic manuscripts are still written without enough attention to the punctuation marks. Since such marks are important indicators of the text segments boundaries and are normally used in text segmentation, the absence of them in Arabic texts makes the segmentation process of such texts more challenging. It has been observed, however, that meaningful parts in the text are not isolated and that they exist together with some connectors (Al-Sanie *et al*., 2005b; Mann and Thompson, 1988; Marcu, 2000a). The technique presented in this study makes use of this observation and deals with punctuation marks as normal characters. Therefore, no special attentions are made to the punctuation marks. The proposed technique looks for the potential connectors to identify the standalone parts. The idea is illustrated with the following example, referred to hereto after as example 1:

لم يحضر محمد إلى الاجتماع لأنه كان مسافرا.

The above Arabic sentence, which is translated to:

Mohammed didn't come to the meeting because he is out of the country.

has two facts (appear underlined) which can be broken into two segments as follows:

[لم يحضر محمد إلى الاجتماع] [لأنه كان مسافرا.]

[Mohammed didn't come to the meeting] [because he is out of the country]

The above example illustrates the sentence after it has been broken into two standalone segments.

With this in mind, two propositions have been considered for the sake of segmenting Arabic texts. The first proposition assumes dividing the text into sentences that satisfy the formal definition of the Arabic sentences. The second proposition segments the text semantically. Definition 1 details the first proposition:

Definition 1: An Arabic sentence is either:

- Noun sentences which consists of starter "مبتدأ" and complement "خبر". Example:
  Mohammed is a student — محمد طالب.
  (Starter) (complement)
- Verb sentences which consists of verb + subject, or verb + subject + object. Example:
  Mohammed has left — غادر محمد.
  (subject) (verb)
  Mohammed ate the apple. — أكل محمد التفاحة.
  (subject) (verb) (object)

This proposition has two disadvantages:

(i) There is an ambiguity in identifying the starter and the complement in the first category and identifying the verbs, subjects and objects in the second category. The following examples illustrate.

Mohammed is a diligent student. — محمد طالب مجتهد.

Can we identify the complement using a surface parser?

I saw Mohammed's car. — شاهدت سيارة محمد.

Here the object is a compound sentence consisting of two words. Again, can the object be identified using a surface parser?

(ii) It does not take into consideration the completeness of the meaning. Consider this example:

I fell down and got pain. — سقطت فتألمت.

If it is divided according to the definition of the Arabic sentence (Definition 1) we will have meaningless segments:

[I fell down] [and got pain] [سقطت] [فتألمت.]

On the other hand, the second proposition segments the text semantically. Here, the aim is to divide the text into complete meaningful parts which can exist independently without their prefix or postfix parts (it assumes that the references in the segments can be substituted by their referees). To illustrate consider example 1 again:

[لم يحضر محمد إلى الاجتماع] [لأنه كان مسافرا.]

[Mohammed didn't come to the meeting] [because he is out of the country]

It is observed that, regardless of the connector because, there are two segments which can be processed independently, Mohammed didn't come to the meeting$_1$ and Mohammed is out of the country$_2$.

In this study, the second proposition is adopted in identifying the Arabic text segments to overcome the disadvantages in the first proposition. Our technique is built upon surface linguistic processing in an attempt to break the text into standalone segments (i.e., no sentence is cut in the middle).

CORPUS ANALYSIS

In order to identify the connectors that divide complex sentences into standalone and meaningful segments in the Arabic text, a list of candidate segments connectors is extracted based on the work done in (Al-Sani *et al*., 2005a, b; Mathkour *et al*., 2005). The list is expanded by other candidate connectors, from the famous Arabic references Mughni Al-labeeb An Kutub Al-aareeb and Al-gana Al-dani Fi Huroof Al-maani (Al-Ansari, 2003; El-Masri, 2001), which identify the meaning and functionality of the Arabic words. After obtaining the complete list of candidate segments connectors, we collected a corpus containing 100 articles. Each article ranges between 450 and 800 words. An engine (a program) is run to process these articles looking for the elements in the candidate segments connectors list. After identifying such elements in the text samples, the engine produces an output containing the text samples with each candidate connector identified in terms of the position and the preceding and the following sentences. Finally, the output of the engine is analyzed to study the functionality of each candidate in terms of separating standalone segments. The following definition, which introduces the notion of active and passive connectors, is concluded.

Definition 2: Let L be a list of candidate segments connectors, each element c in L is classified based on its effects on the text segmentation as either active or passive, where:

- Active: words that indicate the beginning of a new segment, the end of a segment or a complete segment.
- Passive: words that don't indicate a new segment, an end of a segment or a complete segment by themselves, but when they come with active elements, they contribute in determining the position of the start or the end of the segments.

To illustrate the above definition, here presents the following example (it should be noted that the usage of the language in this way is intended to illustrate the ability of the proposed technique to deal with texts that are not necessarily well-written):

[تعتزم إدارة الجامعة إنشاء قسم جديد في الكلية]
[هنالك بعض التقارير التي تؤكد إنشاء هذا القسم.]

[The university administration intends to establish a new department in the college] [there are some reports which confirm this intention].

In this example, the connector "هنالك" (underlined) is an active candidate which indicates the beginning of a new segment. If here add "و" (and) before the connector "لكن" it gives the following:

[تعتزم إدارة الجامعة إنشاء قسم جديد في الكلية و]
[هنالك بعض التقارير التي تؤكد إنشاء هذا
القسم و [لكن لم يحدد بعد موعدا لذلك.]

[The university administration intends to establish a new department in the college] [and there are some reports which confirm this intention] [but no specific time is set yet].

In the above example, there are three connectors: "و", "هنالك" and "لكن". The connector "و" is passive. It doesn't indicate a new segment, but its appearance with the active connector "هنالك" makes it possible for the segmentation processor to detect that the new segment boundary starts before the active connector "هنالك", precisely, at the new position which is before the passive connector "و". Similarly, for the connector "لكن" (Fig. 1). Each structure holds the connector together with its type and position.

This will result in obtaining the above two segments instead of the following segments, where the segmentation process fails to detect the passive connector:

[تعتزم إدارة الجامعة إنشاء قسم جديد في الكلية و]

[هنالك بعض التقارير التي تؤكد إنشاء هذا

القسم و [لكن لم يحدد بعد موعدا لذلك.]

[The university administration intends to establish a new department in the college and] [there are some reports which confirm this intention,] [but no specific time is set yet].

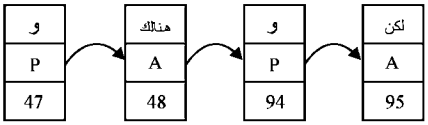

Fig. 1: Identifying the active and passive connectors

Table 1: An example of the list of segments connectors

SegConnector	In English	Type	Seg boundary
\s(و\s*إذا)\s	Therefore	A	B
،\s*(و?\s*مهما)\s	Even though	A	B
،\s*(و?\s*إلا)\s	Unless	A	B
(؛)	;	A	A
.	.	A	A
:	:	A	A
(\}.*?\{)	Curly braces	A	S
(\].*?\[)	Square brackets	A	S
\s(و?\s*في)\s	In	P	
\s(و?\s*مع)\s	With	P	
\s(و?\s*لو)\s	If	P	

Some connectors indicate the end of a segment; an example is "؟". This connector instructs the segmentation processor to put the segment boundary after it.

As mentioned in the definition, some active connectors indicate a complete segment. For example, the two curly braces "}{" along with the text it contains is considered a complete segment.

The result of the corpus analysis is a list of strings extracted from the list of candidates segments connectors represented in a regular expression that matches all the existence found in the corpus. Each string has two properties, the first property indicates whether the element is an active or a passive segments connector (A or P) and the second property indicates the position of the segment boundary in case this element is an active connector, where B indicates before the connector, A indicates after the connector and 'S' indicates surrounding the connector-in case the connector indicates a complete segment. Table 1 shows a part of the corpus analysis results to illustrate the entries of the segment connector list, where the segments connectors are represented in the java regular expressions. This is considered a preprocessing step to set the stage for the segmentation process.

THE SEGMENTATION PROCESS

The segmentation process shown in Fig. 2 is obtained through the following steps:

- Identifying the connectors that indicate complete segments (with S instances in the SegBoundary property in Table 1).

```
segmentText (in text: String; out segments: Array of String)

Active: List of active connectors;
Passive: List of passive connectors;
Segments: Array of text segments;
Blocks: List of blocked segments;

begin
        identifyCompleteSegments (text, Active, Blocks);
for each marker in Active
begin
      if marker. segBoundary <> 'S'
            text.replaceAll (marker, #marker#); //change each occurrence
                                                // of marker to #marker#
        end
        resolveAdjacentActiveConnectors (text);
        setSegmentsBoundaries (text, Passive);
        createSegments (text, Blocks, Segments);
end
```

Fig. 2: The Arabic text segmentation process

- Locating the active connectors.
- Resolving the case where adjacent active connectors exist.
- Setting the segments boundaries.
- Creating the final list of segments.

The first step is to keep the complete segments from being broken by the subsequent steps. As an example, consider the process of the following sentence.

لتعريف المصطلحات أعلاه يمكن الرجوع
إلى لسان العرب [المجلد الأول: صفحة 70-90]

The definition of the above terms can be found in **Lisan Alarab** [Volume 1: pp 70-90].

As shown in Table 1, the square brackets along with the text embodied inside them constitute complete segments. When these segments are not identified and the process attempts to locate the active connectors and segments the text accordingly, the result will be the following (Note: in this example we used '<' and '>' as segments separators, because ']' and '[' used in previous examples are included in the body of the text):

لتعريف المصطلحات أعلاه يمكن الرجوع
إلى لسان العرب [المجلد الأول:> <صفحة 70-90]>

<The definition of the above terms can be found in Lisan Alarab [Volume 1:> < pp 70-90]>.

As shown in Table 1, the active connector ':' specifies the end of the segments, when the parser goes over the above text, it breaks the sentence inside the square brackets as shown above. To overcome this problem, identifyCompleteSegments in the segmentation process identifies the complete segments first to protect them from such incident (Fig. 3).

The second step in the segmentation process is locating the active connectors. This is done in the iteration block in Fig. 1. It goes over each element in the list of active connectors and then searches for it in the text. It ignores the active connectors responsible for the complete segments, since, they have been processed in the previous step.

When all the active connectors are identified and before the final segments are created, the segmentation process resolves the case where adjacent active connectors exist. Whenever such a case is found, the processor will consider the first connector and ignore the second. e.g.:

```
….. word1 word2 word3 activeConn1 activeConn2 word4 word5 ….
                        ↑           ↑
                     consider     ignore
```

The result shown here is obtained through resolveAdjacentActiveConnectors (Fig. 4):

[….. word1 word2 word3] [activeConn1 activeConn2 word4 word5 ….]

The first statement means "## " is followed by zero or more spaces, followed by ##marker2##, which indicates two adjacent active markers. An example is ##marker1## ##marker2##. The statement text.replaceAll (pattern, #marker2) means ignore the second marker. The result will be #marker1# marker2.

Finally, the processor sets the segments boundaries and creates the final list of segments which are performed in the consecutive steps setSegmentsBoundaries and createSegments (Fig. 5).

```
identifyCompleteSegments (in and out text: String; in active: List; out Blocks: List)

begin
        for each marker in active
        if marker exists in text and marker.setBoundary == 'S'
        begin
                text.replaceAll (marker, @@@@@);
                add marker to Blocks;   //Blocks is a list of markers replaced by
        end                             // @@@@@
end
```

Fig. 3: The process of identifying the complete segments

```
resolveAdjacentActiveConnectors (in and out text:String)
begin
        pattern = "##\s*#marker2##";
        if pattern exists in text
                text.replaceAll (pattern, #marker2);
end
```

Fig. 4: The process of resolving adjacent active connectors

```
createSegments (in and out text:String; in Blocks: List; out Segments: Array)

Blocks: List of blocked segments;
begin
        pattern = "@@@@@";                          //return back the blocked segments
        while pattern exists in text
        begin
                text.replace (pattern, Block.getFirst);
                Block.removeFirst;
```

Fig. 5: The Process of creating the final segments

Table 2: Matches between the judge and the segmentation process

Essay	Correct hit	Incorrect hit
1	33	0
2	15	1
3	25	0
4	23	1
5	20	0
6	29	1
7	26	1
8	33	2
9	26	0
10	22	0

DISCUSSION

In order to evaluate the segmentation process, we collected ten essays. Each essay ranges between 500 and 700 words. Since the segmentation process is semantic based, a subjective evaluation is used in our experiment. After implementing the segmentation process, it is run on the collected essays. We then, gave the output to judges (linguistic experts in the Arabic language) to evaluate them in terms of two factors: correct hit and incorrect hit. Correct hit represents the position marked by the process as a segment boundary and agreed by the judge. Incorrect hit represents the position marked by the process as a segment boundary and the judge disagrees with it. Table 2 shows the result for the ten essays. It is used the three factors as a measure instead of other measures such as precision to further analyze the technique. It can be shown from Table 2 that the proportion of the incorrect hit is very low. This is a consequence of the strict selection of the segments connector from the corpus analysis, which is limited to words that appear as segments connectors in all the texts in the corpus.

CONCLUSION

Text segmentation is needed by many high level text processing applications. This paper demonstrates a segmentation technique based on a linguistic empirical study. The technique is based on the analysis of the Arabic corpus to extract words that appear frequently as connectors of two standalone text segments (active), or words that assist in connecting two standalone segments (passive). The approach of extracting such connectors is shown and the result of using these connectors in the segmentation engine is presented.

Based on the output analysis, some active connectors might appear in other texts as passive. The effect of each connector as either active or passive is based on the context. However, the frequency of the effect of each connector in our experiment derives its classification. The segmentation could be improved by performing an empirical study on a larger corpus to identify segments connectors using the N-gram model. The accuracy of identifying the segments is based on the accuracy of classifying the connectors and the number of connectors in the list.

ACKNOWLEDGMENTS

This research is partially supported by the research center of the college of computer and information sciences in King Saud University.

REFERENCES

Agichtein, E. and V. Ganti, 2004. Mining reference tables for automatic text segmentation. Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'04), Seattle, Washington, USA, August 2004, ACM Press, pp: 20-29.

Al-Ansari, I.H., 2003. Mugni Al-Labeeb An Kutub Al-Aareeb. Al-Maktabah Al-Asriah for Publishing and Printing.

Al-Sarie, W., A. Touir and H. Mathkour, 2005a. Towards a suitable representation of Arabic text summarization. Proceedings of the 7th International Conference on Information Integration and Web-based Applications and Services, Kuala Lumpur, Malaysia, September, CIMCA/IAWTIC 2005. pp: 535-542.

Al-Sarie, W., A. Touir and H. Mathkour, 2005b. Towards a rhetorical parsing of Arabic text. Proceedings of the International Conference on Intelligent Agents, Web Technology and Internet Commerce, Vienna, Austria, November, iiWAS 2005, pp: 1086-1091.

Beeferman, D., A. Berger and J. Lafferty, 1997. Text segmentation using exponential models. Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing, Providence, Rhode Island, USA, August 1997, pp: 35-47.

Beeferman, D., A. Berger and J.D. Lafferty, 1999. Statistical models for text segmentation. Mach. Learning, 34: 177-210.

Chang, D.S. and K.S. Choi, 2005. Causal Relation Extraction Using Cue Phrase And Lexical Pair Probabilities. Natural Language Processing - IJCNLP 2005, Lecture Notes in Computer Science, Vol. 3248, Springer, Berlin, Heidelberg, Germany, pp: 61-70.

Chang, D.S. and K.S. Choi, 2006. Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. Inform. Process. Manage., 42: 662-678.

Cristea, D., O. Postolache and L. Pistol, 2005. Summarisation Through Discourse Structure. Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, Vol. 3406, Springer, Berlin, Heidelberg, Germany, pp: 632-644.

El-Masri, B.H., 2001. Al-Gana Al-Dani Fi Huroof Al-Maani. Al-Maktabah Al-Asriah for Publishing and Printing.

Golcher, F., 2006. Statistical text segmentation with partial structure analysis. Proceedings of 8th Conference on Natural Language Processing (KONVENS 2006), Konstanz, Denmark, October 2006, Universität Konstanz, pp: 44-51.

Haouam, K., A. Touir and F. Marir, 2003. Towards a framework design of a retrieval document system based on rhetorical structure theory and cue phrases. Proceedings of the International Intelligent Information Systems/Intelligent Information Processing and Web Mining Conference, Zakopane, Poland, June 2003, Springer, pp: 139-148.

Lamprier, S., T. Amghar, B. Levrat and F. Saubion, 2007. SegGen: A genetic algorithm for linear text segmentation. Proceeding of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07), Hyderabad, India, January 2007, pp: 1647-1653.

Le Thanh, H., G. Abeysinghe and C. Huyck, 2004. Automated discourse segmentation by syntactic information and cue phrases. Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), Innsbruck, Austria, February 2004, pp: 411-415.

Mann, W.C. and S. Thompson, 1988. Rhetorical structure theory: Toward a functional theory of text organization. Text, 8: 243-281.

Marcu, D., 1997. The rhetorical parsing of natural language texts. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, July 1997, pp: 96-103.

Marcu, D., 1999. Discourse Trees are Good Indicator of Importance in Text. In: Advances in Automatic Text Summarization, Mani, I. and M. Maybury (Eds.). MIT Press, Cambridge, MA, pp: 123-136.

Marcu, D., 2000a. The Theory and Practice of Discourse Parsing and Summarization. 1st Edn. The MIT Press, UK .

Marcu, D., 2000b. The rhetorical parsing of unrestricted texts: A surface-based approach. Comput. Linguistics, 26: 395-448.

Mathkour, H., A. Touir and W. Al-Sanie, 2005. Automatic information classifier using rhetorical structure theory. Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM´05 Conference, Gdansk, Poland, June 2005, pp: 229-236.

Mazur, P.P., 2005. Text segmentation in Polish. Proceedings of 5th International Conference on Intelligent Systems Design and Applications, Wroclaw, Poland, September 2005, pp: 43-48.

Sebastian, N. and A. Costa, 1997. Metrical information in speech segmentation in Spanish. Language and Cognitive Processes, 12: 883-887.

Sparck-Jones, K., 1999. Automatic Summarising: Factors and Directions. In: Advances in Automatic Text Summarization, Mani, I. and M.T. Maybury (Eds.). The MIT Press, UK., pp: 1-13.

Utiyama, M. and H. Isahara, 2001. A statistical model for domain-independent text segmentation. Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics and 10th Conference of the European Chapter of the Association for Computational Linguistics (ACL2001), Toulouse, France, July 2001, pp: 491-498.

Villatoro-Tello, E., L. Villasenor-Pineda and M. Montes-Y-Gomez, 2006. Using Word Sequences for Text Summarization. Text, Speech and Dialogue, Lecture Notes in Computer Science, Vol. 4188, Springer, Berlin, Heidelberg, Germany, pp: 293-300.

Wu, Z. and G. Tseng, 1995. ACTS: An automatic Chinese text segmentation system for full text retrieval. J. Am. Soc. Inform. Sci., 46: 83-96.

Yang, C.C. and K.W. LI, 2005. A heuristic method based on a statistical approach for Chinese text segmentation. J. Am. Soc. Inform. Sci. Technol., 56: 1438-1447.