Available online at https://www.asjp.cerist.dz/en/PresentationRevue/134

RIST

Information Processing at the Digital Age Journal

Impact of Stemming Techniques on Topic Segmentation of Arabic Texts

Belahcene Bahloul[a,b*], Hassina Aliane[c], Mohamed Benmohammed[b]

[a]*University of Khemis Miliana, Ain Defla, Algeria*
[b]*LIRE Laboratory, University of Constantine 2, Algeria*
[c]*CERIST Research center, Algiers, Algeria*

Abstract

In this paper, we propose a topic segmentation approach for Arabic texts, through which we have studied the effect of the application of two different stemming techniques, root-based and light stemming. The approach we propose is global, distributional, non-linear. It is global since it considers a comparison of all text segments and not only neighboring segments. It is non-linear in the sense that it can rank segments situated in different positions in text in same groups (subtopics). The approach is based on the calculation of lexical cohesion between segments basing on a combination of repetitive lexical semantic criteria. For terms weighting, we have used OKAPI (BM25) measure after an operation of stemming using both root-based stemming and light stemming. The semantic repetitions of terms are calculated using Arabic WordNet lexical database. A similarity matrix is created where rows and columns are the text segments and the elements of the matrix are COSINE scores between pairs of segments. Subtopics are finally formed using a strict clustering technique in order to eliminate redundancy in the segment groups. For experimentation, we tested our system on a collection of economic and web news articles using Recall, Precision, F-measure and WindowDiff. The obtained results are very promising.

Keywords: Arabic language processing, Subtopic segmentation, Stemming, Lexical cohesion, Terms weighting;

1. Introduction

Topic segmentation is used to identify the maximum possible relationships between different segments of a document, in order to group together the most consistent ones. It's a rudimentary step for several treatments

* Corresponding author. Tel.: +213542809158
E-mail address: d.bahloul@univ-dbkm.dz.

like automatic summarization, information retrieval and document indexation. According to [Choi, 2000], topic segmentation has as object to find in a text a set of portions thematically coherent and distinct of the neighboring portions. According to [Hearst, 1997], the objective is to segment the text into several contiguous, nonoverlapping blocks that are thematically coherent. For the last years, several approaches have been proposed for the topic segmentation and they can be classified in endogenous approach and exogenous approach. The first approach exploits the information contained in the text to be segmented such as lexical repetition. The second approach uses external resources like: thesaurus, dictionary and co-occurrence network [Chaibi et al, 2014]. In the literature, several segmentation algorithms have been proposed. We can classify them into two categories, linear or non-linear (hierarchical). Linear segmentation algorithms such as TextTiling [Hearst, 1997], Segmenter [Kan et al, 1998], DotPlotting [Reynar, 1998], C99 [Choi, 2000], produce contiguous segments according to their order of appearance in the source document. The hierarchical algorithms such as Probabilistic segmenter [Utiyama and Isahara, 2001] and [Simon et al, 2013], Minimum cut segmenter [Malioutov and Barzilay, 2006], Bayesian segmenter [Eisenstein and Barzilay, 2008], HAPS (Hierarchical Affinity propagation for segmentation) [Kazantseva and Szpakowicz, 2011] and [Kazantseva and Szpakowicz, 2014] aim to group not necessarily neighboring segments into the same thematic groups. In the two families of algorithms, segmentation is based mainly on the census of the lexical and semantic relations between segments. Topic segmentation is an important topic in the treatment of natural language because it is closely related to morphological analysis and it is much more with the case of rich and complex languages morphologically such as Arabic. Works on Arabic language are too little. The only works identified are: [El-Shayeb et al, 2007] who developed a system named ModSeleCT. The system is based on a linguistic technique called lexical chaining which measures the cohesion between textual units. ModSeleCT was compared against three algorithms: SeLeCT [Stokes et al, 2004], LCseg [Galley et al, 2003] and TextTiling [Hearst, 1997]. The works of [Harrag et al, 2010] and [Chaibi et al, 2014] consist on a comparison between the two algorithms TextTiling and C99 implemented on Arabic texts. [Harrag et al, 2010] developped ArabTiling and TopSegArab (which are respectively, an adaptation of TextTiling and C99 segmenters). The two systems are implemented on IR where they confirmed that the use of topic segmentation gives more better results than without. [Chaibi et al, 2014], on their part, developed ArabC99 and ArabTextTiling (which are respectively an adaptation of Text- Tiling and C99). Their evaluation results show that ArabC99 is better than TextTiling Arabic version. In this paper, we present a subtopic segmentation approach based essentially on maximizing lexical cohesion between text segments. We combine lexical-semantic criteria to calculate distributional similarity. For the calculation of terms repetitions in the text, we considered the two stemming techniques, root-based stemming and light-stemming, where the objective is to test the effect of each of the two techniques on the overall performance of the segmentation system. The rest of this paper is organized as follows. Section 2 states potential interesting contributions of lexical cohesion detection in topic segmentation problems. Section 3 presents an overview on distributional similarity and some of its techniques. We described the proposed approach in Section 4 and Section 5. The evaluation and experimental results are discussed in Section 6 and finally, we end up with a conclusion in Section 7.

2. Lexical cohesion

Automatic detection of lexical cohesion between two segments in a text consists to identify and valorize the relationships that can link them in the same context. It is essentially based on repetition of words or detection of semantic links. In literature, Cohesion and text are inseparable terms. Cohesion is the glue that binds the relations of meaning within a text to make it an integrated unit [Al-Janabi, 2013]. Cohesion can be split into grammatical and lexical cohesion. Lexical cohesion (in Arabic التماسك المعجمي / a'txxam'Asok Almo‘ˆcame) is the most widely used cohesive tie [AL-Shurafa, 1994]. It can be defined as the cohesive effect achieved by the selection of vocabulary [Halliday and Hasan, 1976]. Detecting lexical cohesion between segments in a text constitutes a rudimentary step in the segmentation process. Lexical cohesion can

be defined as semantic relationship between elements forming a text. The marks of lexical cohesion between two segments can be generally spotted by the six following relationships: repetition, synonymy, hyponymy, antonymy, metonymy and equivalence between terms [AL-Shurafa, 1994]. An experiment on Arabic and English texts [Al-Janabi, 2013] showed that the ratio of reiteration in Arabic literary texts is higher than in English literary texts (84.21% to 70.10%) and the recurrence of Arabic words repetition is higher than that of English (52.22% to 43.47%) and the occurrence of synonymy in English text is higher than that in Arabic texts (15.21% to 9.71%). This clearly shows that lexical cohesion detection by reiteration (particularly repetition and synonymy) is predominantly the most appropriate in automatic processing. The detection of lexical cohesion is mainly based on a calculation of similarity between segments. According to the types of relationships mentioned above, the distributional similarity calculation uses repetition of terms and tracking of semantic links.

3. Distributional similarity

Distributional similarity technique has been used since a long time for the construction of semantic classes of words. The method assumes that semantically close words tend to appear in same contexts [Bannour et al, 2011]. To measure the semantic similarity between words, we can use lexical resources like Arabic WordNet. Before applying any similarity measure using the distribution of words, we must firstly, quantify the occurrences (repetitions) of the same terms in the text and on the other hand, identify semantic links between terms. To apply any similarity measures, it is necessary to calculate the distribution of terms in different segments of the text. Instead of calculating simple occurrences of a term, we can weight its importance according to its repetitions, in this way we can not only measure the similarity between segments but also identify the rank of this similarity according to all pairs of segments in a text. The most used weighting method is TF-IDF (Term Frequency / Inverse Document Frequency) which is a weighting method often used in information retrieval, particularly in text mining. This statistical measure is used to evaluate the importance of a term contained in a document in relation to a collection or corpora. The weight is incremented with the number of occurrences of the word in the document. Lot of variations of the original formula are often used to evaluate the relevance of a document basing on the users search criteria [Bougouin, 2013]. For example, the equation of [Jones, 1972] quoted below, compares the behavior of a candidate term in the analyzed document with its behavior in a document collection:

$$TF - IDF(terme) = TF(terme) \times \log\left(\frac{N}{DF(terme)}\right) \quad (1)$$

TF represents the number of occurrences of a term in the analyzed document. *DF* is the number of documents in which it is present and N is the total number of documents. An alternative measure to TF-IDF is OKAPI (or BM25) [Robertson et al, 1999], It is still considered one of the methods in the state of the art in this field. OKAPI is described as a TF-IDF taking better into account the length of documents. This measure is used to normalize the *TF* (which becomes *TFBM25*) [Bougouin, 2013]:

$$Okapi(terme) = TF_{BM25}(terme) \times \log\left(\frac{N - DF(terme) + 0{,}5}{DF(terme) + 0{,}5}\right) \quad (2)$$

$$TF_{BM25} = \frac{TF(terme) \times (k_1 + 1)}{TF(terme) + k_1 \times \left(1 - b + b \times \frac{DL}{DL_{moyenne}}\right)} \quad (3)$$

k_1 and b are constants. By experimentation [Bougouin, 2013], the best values of k_1 and b are set to 2 and 0,75 respectively. *DL* is the length of the analyzed document (number of its words) and $DL_{avgLength}$ the average length of the document collection. Note that OKAPI can also be adapted to consider a segment (sentence, paragraph) relatively to a text. In our approach, we use an updated version of OKAPI which takes into account the distribution of synonyms. OKAPI seems also the most appropriate technique according to the

nature of the text segments in our experimental dataset which have generally different lengths. To calculate similarity between two sets, several measures exist, the most widely used in NLP applications is COSINE. Several COSINE versions exist, the most used in the calculation of similarity between sentences is that proposed by [Choi, 2000] which was used in its C99 thematic segmentation algorithm. The similarity between two sentences x and y is calculated by the following equation:

$$sim(x,y) = \frac{\sum_j f_{x,j} \times f_{y,j}}{\sqrt{\sum_j f_{x,j}^2 \times \sum_j f_{y,j}^2}} \quad (4)$$

fi,j denotes the frequency of term *j* in the sentence *i*. COSINE gave the best performance according to the prior studies and works relatively in Arabic language [Al-Anazi et al, 2016]. In our work, COSINE measure is updated to take into account the terms weighting using OKAPI metric.

4. Proposed approach

The approach we propose to automatically subdivide text in subtopics is a global, non-linear approach. It combines a statistical distributional calculation (terms repetition) with a lexical semantic model based on the calculation of synonymy between terms. Our approach involves five main steps. A preprocessing step which includes text segmentation of the text into paragraphs, filtering, normalization and stemming. The second one consists on the calculation of OKAPI weighting score basing on frequency and synonymy calculation. The third step concerns similarity calculation, in which we calculate the COSINE scores between all pairs of segments. The fourth step consists to group with each segment the most related segment. The last step consists to apply a strict clustering in which we group in same subgroups (subtopics) all common segments. The approach is a global since it compares a segment with all other text segments, unlike the local approaches that compare only neighboring segments. It is also non-linear, because it can gather in the same group, coherent segments located in different positions in the text. Fig. 1 shows the different processing steps of our system.

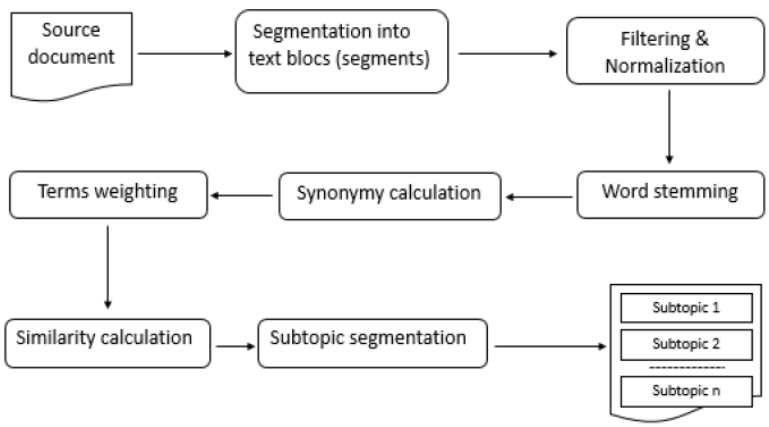

Fig. 1. Subtopic segmentation system architecture

5. Description of the system main modules

5.1. Text segmentation

The objective of This module is to divide the text into text segments (paragraphs) using the stop point (.), the colon (:) and the line break. We avoid also the use of other punctuation signs and some connectors as the (**و** /and) because of ambiguity that can be generated in segmentation.

5.2. Filtering and normalization

This step consists to remove all non-significant words. We compare each recognized word with one of the elements in the stop words list (**كان** , **بعد** ,**أن** , ...). This step consists also to remove special characters and numbers and eliminate diacritics in the case of partially vowelized words. Finally, to prepare the words to the next treatment (stemming), these are standardized by replacing letters such (**ة** , **ي** ,آ, إ ,أ) by (**ه** , **ى** , ا) respectively.

5.3. Stemming and synonyms calculation

For each normalized significant word (verb, noun, adjective), we apply a stemming process using both root based stemming (using Khoja stemmer) and light stemming (using Assem stemmer).The objective is to remove inflected elements (prefix and suffix) to obtain its root and its stem, this allows to return a list of tokens called items. Calculation of roots and stems is very important for the calculation of distribution of occurrences of same words in the text. Roots and stems are also used for synonyms calculation. Semantic repetitions are calculated using the lexical database Arabic WordNet (AWN), which is a lexical-semantic network that for each term it gathers all its synonyms in groups called Synsets. Table I shows the results of stemming and synonymy calculation of the word (**المفاهيم** / AlmafAhym / Concepts).

Table 1. An example of stemming and synonymy calculation

Input: المفاهيم	Output: synset
Root: فهم	سمع، أخذ، فسر، أدرك، تيقن، عقل، استفسر، اتفق، صواب، استعلم، استفهم، سأل، فطنة، معرفة، فكرة عامة، مفهوم، تصور، وافق، تفاهم، رشد، بصيرة، دراية
Stem: مفاهيم	فكرة عامة، مفهوم، تصور

5.4. Terms weighting

This step consists in associating a score to each word according to the distribution of its occurrences and its synonyms in the text. The distributional repetitive score is calculated by OKAPI metric. Noting that OKAPI in its standard version, calculates weights (weighted frequencies) of a term in a document over a collection of documents. This metric can be adapted to consider a term in a segment over a set of segments (a text). OKAPI score is calculated by equations (2) and (3) and considering: k_1 and b are constants set to *2* and *0,75* respectively. DL is the length of the current segment (number of its words) and $DL_{avgLength}$ the average length of the text segments. TF represents the number of occurrences of a term in the current segment (number of occurrences of the word itself or one of its synonyms. N represents the number of segments in the

text. *DF* is the number of segments in which the term or one of its synonyms appears. OKAPI score shows the importance of a term based on its repetitions in the text.

5.5. Similarity calculation

This step consists to calculate similarity (lexical cohesion degree) between pairs of segments using COSINE measure. The standard version of COSINE (see equation (4)) calculates similarity basing on the words frequencies in the pairs of sentences. We use an updated version of COSINE which considers the weighted frequencies (OKAPI) instead of simple frequencies *f*. We think that this new version measures not only the number of repetitive relations between segments but also the importance of these relationships. The updated equation of COSINE that calculates similarity between two segments (S1,S2) is defined as follows:

$$Cosine_{S1,S2} = \frac{\sum_{j \in S1 \cap S2} OKAPI_{S1,j} \times OKAPI_{S2,j}}{\sqrt{\sum_{j \in S1} OKAPI_{S1,j}^2 \times \sum_{j \in S2} OKAPI_{S2,j}^2}} \quad (5)$$

OKAPISi,j denotes OKAPI score of a term j in the segment Si. After calculation of COSINE values between pairs, a similarity matrix is constructed. Its lines and colons are the segments of the text. An element *SIM[i, j]* represents the COSINE similarity value between segments i and j. The diagonal values *SIM[i, i]* is *1* which represents the similarity of each segment with itself. Fig. 2 shows an example of a similarity matrix of *9* segments.

	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	1.0	0.199	0.114	0.094	0.135	0.188	0.104	0.02	0.0
S2	0.199	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
S3	0.114	0.0	1.0	0.093	0.009	0.09	0.0	0.0	0.0
S4	0.094	0.0	0.093	1.0	0.0	0.0	0.0	0.035	0.0
S5	0.135	0.0	0.009	0.0	1.0	0.197	0.171	0.027	0.0
S6	0.188	0.0	0.09	0.0	0.197	1.0	0.0	0.0	0.0
S7	0.104	0.0	0.0	0.0	0.171	0.0	1.0	0.021	0.073
S8	0.02	0.0	0.0	0.035	0.027	0.0	0.021	1.0	0.0
S9	0.0	0.0	0.0	0.0	0.0	0.0	0.073	0.0	1.0

Fig. 2. An example of a similarity matrix

5.6. Subtopic segmentation

According to the similarity matrix, we associate with each segment *i* the most related segment *j*. The selected segment *j* is the segment having the highest COSINE value in the row of the segment *i*. After the creation of segment pairs, we create a set of subgroups by application of a strict clustering technique. Each subgroup contains a set of most related segments. We rank all pairs having common segment in the same subgroup and we eliminate the redundant segments. The set of segments in the same subgroup are ordered according to their apparition in the source text in order to form subtopics. Fig. 3 shows an example of segments clustering. Fig. 4 shows the subtopic segmented text corresponding to.

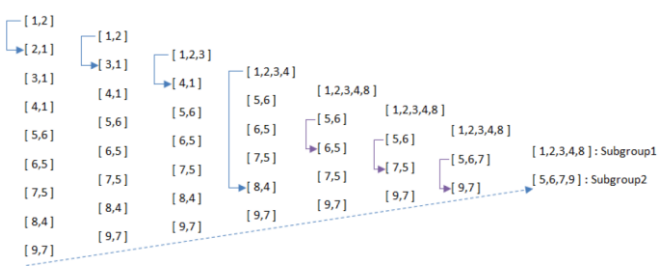

Fig. 3. An example of a strict clustering

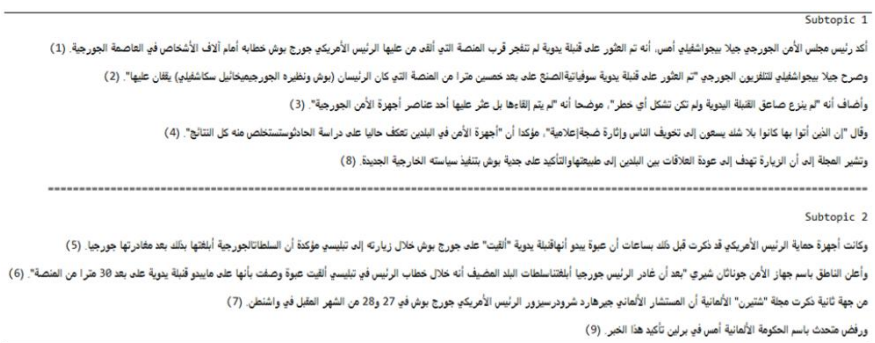

Fig. 4. An example of a subtopic segmented text

6. System evaluation

Evaluate an automatic topic segmentation system is a delicate task. Many issues are raised and can roughly be reduced to two questions: (A) What reference? (B) What evaluation score? [Adam and Morlane, 2009]. To evaluate a system, it must be compared to a reference segmentation. Evaluation can be done manually by using manual annotations, but it usually returns a very low agreement between annotators. The automatic evaluation measures used Pk [Beeferman et al, 1999] and WindowDiff [Pevzner and Hearst, 2002]. The first counts the number of times that the two words chosen at random at a distance *k* are in the same segment both in the reference and in the hypothesis. The second calculates the difference in the number of breaks in a sliding window. Automatic evaluation is generally quantitative and consistency between segment is often ignored. A semi-automatic evaluation can be made by comparing the results produced automatically (judged consistent segment groups) by the system and other manually built by a human expert, in this case, the recall and precision scores are calculated. Our experimental dataset consists of a collection of economic and political articles written in undiacritized Modern Standard Arabic collected from web news and others from an international economic magazine. The corpus consists of about 50 articles with an average length of about 2000 words per article. Our evaluation consists of two tasks. In the first one, we have presented a collection of 50 articles with different lengths previously segmented by the system (separate text segments) to a human expert (graduate in economics) and asking him to group with each segment the most related segments to build consistent distinct groups (which constitutes reference dataset). The segment groups constructed by the human expert are compared with those constructed automatically by the system. For each group of segments, we have calculated correctly grouped segments, incorrectly grouped segments and forgotten segments. The measures Recall and Precision are calculated by the following equations:

$$Recall = \frac{Correct}{Correct+Forgotten} \quad (6)$$

$$Precision = \frac{Correct}{Correct+Incorrect} \quad (7)$$

- *Correct*: Number of segments correctly grouped by system and by human expert.
- *Incorrect*: Number of segments grouped by system and not by human expert.
- *Forgotten*: Number of segments grouped by human expert and not by system.

For example, to calculate Recall and Precision for two segment groups, one produced by our system and the other by human expert, we must do the comparison presented in Table 2.

Table 2. Calculation of Recall and Precision

System group	Human expert group
[1,2,5,8,11,12,16]	[1,2,3,7,8,12,13,15,16]
Recall = 5/(5+4) = 0:55	3, 7, 13, 15 : Forgotten segments
Precision = 5/(5+2) = 0:71	5, 11 : Incorrectly grouped segments

This comparison is repeated for all segment groups produced from a text segmentation. The final evaluation value is calculated by F-measure:

$$F - measure = \frac{2\times(Recall \times Precision)}{Recall+Precision} \quad (8)$$

To see the effect of using root-based stemming (RB) and light stemming (LS) on the process of segmentation, the evaluation is done for the two cases. Table 3 shows the global results for the first evaluation task.

Table 3. Recap of semi-automatic evaluation results

Number of articles	50
Avg. number of segment per article	20
Avg. number of total words per segment	95
Avg. number of significant words per segment	58
Recall (RB)	0.62
Precision (RB)	0.69
F-measure (RB)	0.65
Recall (LS)	0.63
Precision (LS)	0.72
F-measure (LS)	0.67

The second evaluation task consists to evaluate the linear and non-linear segmentation (Since the non-linear include the linear segmentation). We formed a set of new articles from different text portions (each portion contains a number of segments) extracted from different articles discussed different subjects. In the first evaluation step, we put the different portions in a sequential order to form a new article which will be segmented by the system. The system must rank each portion in an independent group and detect the correct boundaries. In the second evaluation step, we put the different segments of each portion in a non-sequential order to form a new article. The system must also rank each segment in its original portion (correct group). We calculated Recall, Precision and F-measure in the two cases linear and non-linear segmentation. Table 4 shows the results of this evaluation.

Table 4. Evaluation results for the linear and non-linear segmentation

Segmentation technique	Recall	Precision	F-measure
Linear segmentation (RB)	0.73	0.75	0.73
Non-linear segmentation(RB)	0.72	0.73	0.72
Linear segmentation(LS)	0.75	0.81	0.77
Non-linear segmentation(LS)	0.73	0.79	0.75

The results of the first evaluation (Table 3) show that our system produces fairly good results compared to the difficulty of this type of intra-document topic segmentation. Indeed, the human expert does not base only on distributional criteria to detect subtopics but he also involves other criteria which touch on the pragmatics of the texts. We notice also that the performance of the segmentation process with using light stemming is

better than with root-based stemming. In the second evaluation (Table 4), the results show that our system generally makes the right dependencies between segments and their original articles in both cases, linear and non-linear segmentation. The results in both cases are almost the same which proves once again that non-linear segmentation is very powerful because it includes linear segmentation. Also, we conclude once again that light stemming is more performant than root-based stemming.

In order to evaluate the performance of our system against other topic segmentation systems. We made a comparison with three systems using WindowDiff (WD) metric. The three systems are: ModSeleCT [El-Shayeb et al, 2007], ArabTextTiling and ArabC99 [Chaibi et al, 2014]. The experimental dataset consists of 100 news articles collected from Arabic reuters[†] news stories. It should be noted that since we could not access to the authors' corpus, we created our own corpus using the same category of articles (news) used by the three systems. We created 10 new articles from the initial dataset. Each one is formed by concatenation of 10 articles selected randomly from the initial dataset. The objective is to compare the performance of our system in detecting correct boundaries. A sliding window is used to compare the automatically topic segmented text with the articles formed by concatenation. The system is evaluated with its two versions, with the light-stemming (LS) and the root-based stemming (RB). Table 5 shows the comparison results of our system against the other systems.

Table 5. Comparison results of the proposed segmenters with other systems

Segmenter	WindowDiff	Dataset
Proposed system (RB)	0.33	100 articles (Arabic Reuters
Proposed system (LS)	0.27	100 articles (Arabic Reuters)
ModSeleCT	0.2	1000 articles (Arabic Reuters)
ArabTextTiling	0.55	120 articles (Arabic Newspaper)
ArabC99	0.42	120 articles (Arabic Newspaper)

According to the comparison results (Table V), we see that ModSelect presents the best WindowDiff value (0.2) followed by the proposed segmenter (LS) with (0.27). This is probably due to the same category of articles used. It should be noted that to make a worthy evaluation, a same test corpus must be used. Despite the different results and corpora, we can say that our segmenter has well performed against other systems.

7. Conclusion

Currently, automatic topic segmentation is mainly based on repetition of words having similar morphological structures. The new segmentation techniques consider the meaning of terms instead to taking them as simple strings. In this context our approach aims to enhance not only the relationship that may exist between segments but also the importance of these relationships. The evaluation results show clearly that our approach gives good results. We notice that distribution of terms as well as the use of well-defined vocabularies to express ideas in text are always considered as good factors for making good intra-document classification and of course a good topic segmentation. Moreover, as improvement of this work, we propose the study of other relationships such as collocations for example and the use of other distributional similarity techniques such as word2vec and Latent Semantic Analysis (LSA). Finally, we think that testing the approach

[†] https://ara.reuters.com

we have proposed in information retrieval and automatic summarization systems constitutes a good opportunity.

References

Choi, F. Y. Y., 2000. Advances in domain independent linear text Segmentation. In: Proceedings of NAACL-00. pp. 26–33.

Hearst, M. A., 1997. TextTiling: Segmenting Text into Multiparagraph Subtopic Passages. Computational Linguistics 23, pp. 33-64

Chaibi, A. H., Naili, M.,Sammoud, S.: Topic segmentation for textual document written in Arabic language. In: Proceedings of 18th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2014. Procedia Computer Science 35, pp. 437–446.

Kan, M.Y., Klavans, J.L., McKeown K.R., 1998. Linear segmentation and segment significance. In: Proceedings of 6th international Workshop of Very Large Corpora (WVLC-6).

Reynar, J. C., 1998. Topic segmentation: Algorithms and applications. PhD thesis, Computer and Information Science, University of Pennsylvania.

Utiyama, M., Isahara, H., 2001. A statistical model for domain-independent text segmentation. In: Proceedings of the 39th Annual Meeting on the Association for Computational Linguistics, pp. 499-506.

Simon, A., Gravier,G., Sébillot, P., 2013. Un modèle segmental probabiliste combinant cohésion lexicale et rupture lexicale pour la segmentation thematique. In: Proceedings of TALN-RCITAL, Les Sables d'Olonne, France.

Malioutov, I., Barzilay, R., 2006. Minimum cut model for spoken lecture segmentation. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pp. 25-32.

Eisenstein, J., Barzilay, R., 2008. Bayesian Unsupervised Topic Segmentation. In: proceedings of the Conference on Empirical Methods in Natural Language Processing.

Kazantseva, A., Szpakowicz, S., 2011. Linear Text Segmentation Using Affinity Propagation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 284-293.

Kazantseva, A., Szpakowicz, S., 2014. Hierarchical Topical Segmentation with Affinity Propagation. COLING, pp. 37-47.

El-Shayeb, M. A., El-Beltagy, S. R., Rafea, A., 2007. Comparative Analysis of Different Text Segmentation Algorithms on Arabic News Stories. In: Proceeding of IEEE International Conference on Information Reuse and Integration, pp. 441–446.

Stokes, N., Carthy, J., Smeaton, A. F., 2004. SeLeCT: a lexical cohesion-based news story segmentation system. AI Communications, 17, pp. 3–12.

F. Harrag, F., Hamdi-Cherif, A., Al-Salman, A. S., 2010. Comparative study of topic segmentation Algorithms based on lexical cohesion: Experimental results on Arabic language. The Arabian Journal for Science and Engineering, 35 (2C), pp. 183–202.

Al-Janabi, M. K. H., 2013. Lexical Cohesion in Two Selected English and Arabic Short Stories. AL-USTATH Journal, 2 (702), pp. 61–88.

AL-Shurafa, N. S. D., 1994. Text Linguistics and Cohesion in Written Arabic. JKAU Arts and Humanities Journal, 7, pp. 17–30.

Halliday, M.A.K., Hasan, R., 1976. Cohesion in English. London, Longman.

Bannour, S., Audibert, L., Nazarenko, A., 2011. Mesures de similarité distributionnelle entre termes. Travail réalisé dans le cadre du programme Quaero, IC.

Bougouin, A., 2013. Etat de l'art des méthodes d'extraction automatique de termes clés. In: Proceedings of TALN-RECITAL, Les Sables d'Olonne.

Jones, K., 1972. A Statistical Interpretation of Term Specificity and its Application in Retrieval.

Galley, M., McKeown, K., Fosler-lussier, Jing, H., 2003. Discourse segmentation of multi-party conversation, The 41st Annual Meeting of ACL.

Robertson, S., Walker, S., Beaulieu, M., Willett, P., 1999. Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive Track

Al-Anazi, S., AlMahmoud, H., Al-Turaiki, I., 2016. Finding similar documents using different clustering techniques. In: Proceedings of Symposium on Data Mining Applications, SDMA2016, Riyadh, Saudi Arabia. Procedia Computer Science 82, pp. 28–34.

Adam, C., Morlane-H., F., 2009. Détection de la cohésion lexicale par voisinage distributionnel : application à la segmentation thématique. In: Proceedings of RECITAL'09, Senlis, France.

Beeferman, D., Berger, A., Lafferty, J., 1999. Statistical models for text segmentation. Machine Learning, 34 (1-3), pp. 177–210.

Pevzner, L., Hearst, M., 2002. A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics, 28, pp. 1–19.