A Proposed Textual Graph Based Model for Arabic Multi-document Summarization

Muneer A. Alwan[1], Hoda M. Onsi[2]
Information Technology Department
Faculty of Computers and Iinformation, Cairo University
Cairo, Egypt

***Abstract*—Text summarization task is still an active area of research in natural language preprocessing. Several methods that have been proposed in the literature to solve this task have presented mixed success. However, such methods developed in a multi-document Arabic text summarization are based on extractive summary and none of them is oriented to abstractive summary. This is due to the challenges of Arabic language and lack of resources. In this paper, we present a minimal language-dependent processing abstractive Arabic multi-document summarizer. The proposed model is based on textual graph to remove multi-document redundancy and generate coherent summary. Firstly, the original text, highly redundant and related multi-document, will be converted into textual graph. Next, graph traversal with structural rules will be applied to concatenate related sentences to single ones. Finally, unwanted and less weighted phrases will be removed from the summarized sentences to generate final summary. Preliminary results show that the proposed method has achieved promising results for multi-document summarization.**

***Keywords*—Text Summarization; Arabic Abstractive Summary; Textual Graph; Natural Language Processing;**

I. Introduction

The increasing amount of data on the Internet today has led to various trends towards automatic text summarization tools. There are two types of text summarization, Extractive and Abstractive. Extractive summarization aims to select important sentences from the original text and organize these sentences to generate a summary. On the other hand, Abstractive summarization attempts to generate human-like summary and may even produce new sentences. This means that the important ideas in the original text are rewritten to generate coherent summaries. Abstractive methods require a more sophisticated process, involving information fusion, sentence compression, and/or language generation [1]. Due to the difficulty associated with the generation of abstracts, most text summarization techniques only focus on the first type.

According to the literature, great works have been made to build a text summarization system for English language. However, few of these have targeted Arabic language. Moreover, all existing work in Arabic multi-document Summarization used Extractive techniques [2]. This lack or absence of such systems is due to challenges presented by the Arabic language.

Arabic is an inflectional, morphologically complex, highly derivational language. Moreover, Arabic is rich in the use of affixes and clitics and, usually, disambiguating short vowels and other orthographic diacritics in standard orthography are omitted [3]. In addition, for text summarization there is absence of automatic and manual Arabic gold-standard summaries and lack of Arabic natural language processing resources like text generators, corpora, machine-readable dictionaries, lexicons and ontologies.

There are two types of documents to be summarized, single and multi-document. Single document summarization produces summary for one document about a specific subject whereas multi-document summarization aims to generate a single summary of a group of related documents. Online user reviews, tweets in Twitter and comments in YouTube or Facebook websites are the most prominent examples of multi-documents.

The problem with Extractive methods in multi-document summarization is that it should select only the most important sentences along the related documents. This means that there are several sentences that beneficial meanings to be conveyed could be missed in the final summary. To address this problem we proposed a minimal language-dependent processing Abstractive Arabic summarization model. Our model aims to remove the redundancy from highly redundant multi-documents and concatenate the related documents to a single one.

The rest of this paper is organized as follows: Section 2 presents the previous work; Section 3 presents the proposed model; in Section 4 we discuss the evaluation and experimental results; finally, in Section 5, we introduce the conclusion of our work and propose some future work.

II. Previous Work

In English language several pieces of research have been proposed in Text summarization. We are interested in multi-document abstractive summarization approaches that almost can be applied to Arabic language.

In [4] K Ganesan et al. proposed multi-document abstractive text summarizer. The system used a graph data structure that relied on the structural redundancies in the text to discover informative phrases. This work known as Opinosis used graph to get all possible sentences related to a specific query. In [5] Hai-Tao et al. the original text was converted into textual graph and they got the final summary by applying English text syntax rules.

A recent work in [6] Liu et al. have proposed a model that focused on the graph-to-graph transformation to generate abstractive summary. They mapped the source text into

Abstract Meaning Representation (AMR) graphs, and then transformed them into a summary graph to generate final summary. In [7] L Bingis et al. proposed method that generates new sentences by extracting noun phrases and verb-object phrases from the documents. They generate the final summary by merging informative phrases to new sentences.

Multi-document summarization in Arabic language is still in its infancy compared to the literature on English [8] and all existing work use extractive techniques.

In [9] KSAL Harazin et al. used a single document summarization approaches for multi-document summarization, also they provided a model for multi-document summarization that relied on cross document structure theory. In [10] El-Haj and Rayson proposed extractive language-independent summarizer for single and multi-document. A corpus-based technique for both English and Arabic language was applied. They compared lists of word frequencies between two corpora in both languages to compute the log-likelihood score for each word. Summaries were built by selecting sentences that had the highest log-likelihood scores.

In [11] Oufaida et al. presented summarization system for a single and multi-document. In the proposed system, the sentences to be summarized were selected based on the ranks of their terms. To extract summary sentences, the system ranked the terms by using the minimal-redundancy maximal-relevance method (mRMR) [12] and clustering algorithm.

For Abstractive Arabic text summarization, S.Ismail et al. [13] are working on single document summarization. Their proposed system consisted of three modules, first they convert the input Arabic text into a semantic graph called Rich Semantic Graph (RSG). The second and third modules of this proposed model are, performing graph reduction and generating the summary from the reduced graph, respectively.At the present time this research still ongoing.

III. Proposed Model

The proposed model to remove text redundancy and generate Abstractive Arabic text summary consists of 4 stages as shown in Figure 1: 1. Preprocessing to remove text noises. 2. Representing the multi-documents by directed weighted graph. 3. Traversing the graph and applying structural rules to generate the summary sentences. 4. Refining the sentences which contain unwanted parts and adding them into the final summery.

A. Preprocessing

In order to map the original multi-documents into the textual graph, it is preferable to remove a different set of attachable punctuation, diacritics, prefixes and suffixes from the word. For Arabic text Preprocessing we use AraNLP [14] which is a free Java-based library that covers various text preprocessing tools.

Diacritics removal: It removes three forms of diacritics, the Shadda, Nunation Diacritics and Vowel Diacritics. *Punctuation removal*: this tool has been used to remove number of punctuations like Arabic semi-colons (؛), commas (,), Arabic exclamation marks (!), Arabic question marks (؟). *Light stemmer*: this

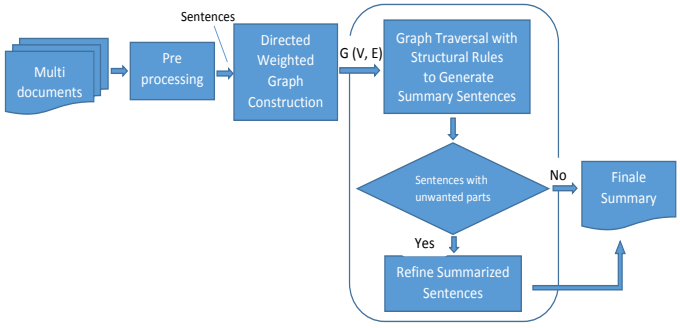

Fig. 1: Overview for the Proposed Model.

is the most important tool, it has been used to remove suffixes and prefixes from the original word. For example, "الجهاز" and "جهاز" have the same conceptual meaning and should map into one word "جهاز". This tool significantly helps to reduce the amount of the processed text. *Stop word recognition*: In this step we do not use AraNlp stop words removal to remove stop words, instead we use it to determine if the word is a stop word or not. The stop words are any word without semantic meanings and are used as an auxiliary words in the sentence, such as "من", "على", "ان", "في". Finally we determine the part of speech, using POS-tagger, for each single word using stanford Arabic word segmenter and POS tagger [15].

B. Constructing the Directed Weighted Graph

Our work exploits textual graph and attempts to enrich Arabic text summarization by new technique in Abstractive summary. Graphs have been commonly used for extractive summarization for example, LexRank by G Erkan et al. [16] and TextRank by R Mihalcea et al. [17] and also for Abstractive summarization for example, Opinosis by K Ganesan et al. [4]. Constructing the textual graph is similar to Opinosis with some differences.

To construct the graph $G(V, E)$, the unique stem (light stemming) for every word in the original multi-documents should map into single node or vertex (V) in the graph. Words with the same stem should map into the same node. The graph is a directed graph where the edge (E) between two nodes (words) in the graph indicates the adjacency (sequential flow) relationship between those words in the sentence. Unlike Opinosis, every stop-word in the original text should map into a single node. To ensure that each node has a unique word, sentence index and word index should attach into every stop-word. The textual graph construction could be summarized as follows:

- For each word:
  - Check:
    - If it is in graph, then do nothing.
  - Otherwise check:
    - If it has adjacent word in the graph, then it becomes next or previous node.
  - Otherwise:

- The word should map into a new initial node that will connect to its potential neighbors.

Figure 2 is an illustration of a simple textual graph construction. The node that contains word mentioned several times through the document, has several next adjacent nodes. For example the word "التابلت" (the red node) has 4 nodes directly connected to it. However, for a word that occurs in a single sentence should has one next adjacent node. Inherently, the graph removes redundancy from the text so the same words in different sentences are mapped into one node in the graph.

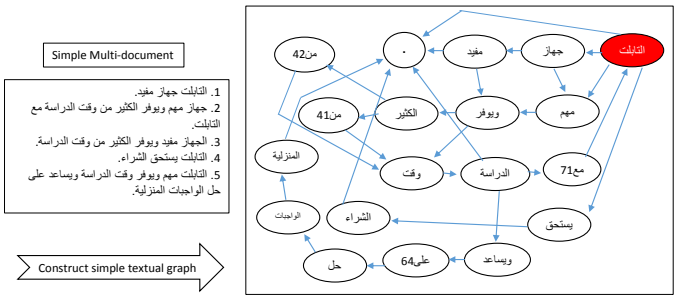

Fig. 2: Simple high redundant text converts into textual graph.

Node Attributes and Weight:

Each node in the graph should keep nine attributes, the original word, word stem, word type, word index, sentence index, sentence length, part of speech tag, word frequency and sentence weight. The word type either stop-word or unstop-word. Sentence index and word index are sentence id in the document and word position in this sentence respectively. For word that occurs in several sentences the node keeps only the id of the first sentence in which this word appears and that is true for the word position as well. Sentence length is the number of tokens (words) in the sentence. Node also keeps part of speech tag for each word either noun, verb, adjective, etc. The weight attribute can be calculated using the following equation:

$$TotalWeight = \frac{1}{m}\sum_{i=1}^{m} W \tag{1}$$

Where m is the length of the sentence and W is the weight of word that can be calculated from the following equation:

$$W = TFIDF * POS * StopWord \tag{2}$$

Where

$$TFIDF = N(1 + log\frac{D}{n}) \tag{3}$$

Where N is the frequency of the word in the multi-document, D is the total number of documents (sentence) and n is the documents that contain this word. POS , empirically, gives 1 for nouns, 0.6 for verbs and 0.3 for others. $StopWord$ is 0 for any stop-word and 1 for others.

C. Graph traversal with Structural Rules to Generate Summary Sentences

We want to generate summary sentences of that have high redundancy (thus summarize the major meaning). Up till now, the graph has removed the redundancy. We need extract the new summarized sentences out of the graph. Depth first traversal search along with structural rules have been used to do as follows:

1) First, we retrieve the words according to their sequence in the original sentences. For this step word index and sentence index should be checked.
2) Mark every node (word) added to the summary as visited.
3) Check if the word has several next neighbors, then this means that there are several related sub-sentences that could be concatenated together and form a new sentence. For example: the word "التابلت" in the simple graph in figure 2 has four next adjacent nodes, so, there are four sub-sentences related to this word that could be concatenated together to form a new sentence. Each sub-sentence should begin by next adjacent node and ends by either full-stop or visited node.
4) Check if the node has no previous node and the next has already been marked as visited then ignore it. This means that the current node (word) is not important enough and it should be avoided to be added to the summary. This leads to reducing the size of the final summary.
5) Check if the node is a stop-word and its next adjacent node has been visited then ignore it. This means that this node is a terminal stop-word and has no meaning to add to the summary.

D. Refining Summarized Sentences and Generating the Summary

At the end of graph traversal we end up with three types of sentences:

1) Sentences that result from merging sentences or sub-sentences together.
2) Sentences that are trimmed from original sentences after unwanted word(s) has/have been removed or sentences that part(s) of them has/have been added to the merged sentences.
3) Sentences without any change in their original body.

In order to make sure that sub-sentences or trimmed sentences have enough meaning to add to the summary the following conditions have been applied:

1) Check if sub-sentence or trimmed sentence weight is greater than a specific threshold t, then it will be included in the summary. Experimentally:

$$t = \frac{1}{6}(TotalWeight) \tag{4}$$

Such that $TotalWeight$ is the weight of the original sentence (equation 1).
2) Check if sub-sentence or trimmed sentence contains more than four words and its weight is less than the

threshold, then add to the summery. This means that the original sentence is too long and the part which we have trimmed out of it conveys enough meaning to be added to the final summary.

3) Avoid adding to the summery those sub-sentences or trimmed sentences that contain only single word or single word with stop word only.

For the simple graph in figure 2 the new summary is:

التابلت جهاز مفيد ويوفر الكثير من وقت الدراسة ويساعد على حل الواجبات المنزلية يستحق الشراء.

IV. Results and Discussion

Text summarization is a very important issue. According to (Lloret et al. 2012) [18] the evaluation of automatic summarization represents a challenging area. However, the summary that obtained from our model has the properties of abstractive summary and, as mentioned in section 1, there is no previous work in Arabic abstractive text summarization. Moreover, the type of data set that have been used to work with (opinions or user reviews) has not used before for Arabic text summarization. This means that, there is no previous works or technologies to compare with. For this reason, to be able to evaluate our model results we went through two ways: manually by recruiting human reviewers and automatically by compare the amount of meaning in the summaries with the amount of meaning in the original multi-documents.

The dataset that has been used to experiment the proposed model was collected from the well-known online shopping website [1] and Twitter.com. Users reviews about twenty-five different products (mobile cell phones and tablets) and tweets talked about five different subjects was crawled from the first and the second websites respectively. We are following multi-document summarization approach where the total number of documents that we have used are 1651 documents grouped into 30 multi-documents.

For the test, 1651 documents have been inserted to the system as input and it generate 441 sentences as summary. Then, the first 293 summarized sentences obtained to five educated users. The users were then asked to tell how much they agreed with the following statement: ''this is a correct and meaningful sentence''. The volunteers then were asked to rate their degree of satisfaction on a 5-point Likert scale where 1 indicates strong unsatisfaction and 5 indicates strong satisfaction.

Figure 3 shows the average results for the five scales. The Likert scale results of the criteria ''this is correct and meaningful sentence'' shows that the raters agree with that 72% of the summarized sentences are correct and meaningful.

Table [1] shows a comparison between the original documents and their associated summery for 30 multi-documents. ''Original Sentences'' column contains the total number of sentences in each multi-document while ''Summary Sentences'' column contains the number of summarized sentences. ''Reduction Ratio'' column presents the proportion of summary sentences to the multi-document sentences. ''Meaning Amount'' column presents the proportion of meaning conveyed by summary to the total meaning in the original multi-document.

[1] egypt.souq.com/eg-ar

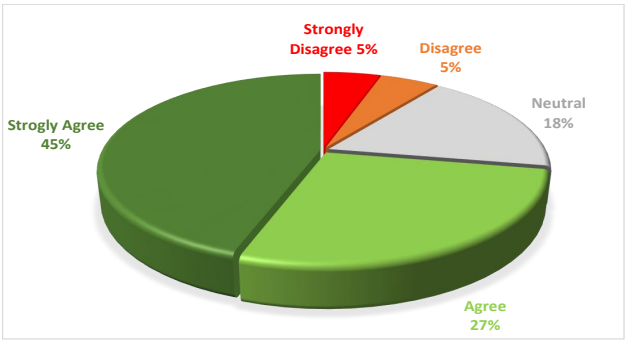

Fig. 3: Results of criteria: ''this is correct and meaningful sentence'' scales.

Multi-Document	Original Sentences	Summary sentences	Reduction Ratio	Meaning Amount
d1	19	8	0.58	0.77
d2	37	15	0.59	0.66
d3	53	15	0.72	0.64
d4	21	8	0.62	0.78
d5	66	19	0.71	0.61
d6	39	9	0.77	0.56
d7	37	15	0.59	0.51
d8	37	9	0.76	0.71
d9	29	9	0.69	0.72
d10	62	21	0.66	0.59
d11	31	8	0.74	0.75
d12	51	14	0.73	0.64
d13	70	11	0.84	0.58
d14	70	20	0.71	0.62
d15	23	6	0.74	0.69
d16	60	19	0.68	0.62
d17	51	27	0.47	0.62
d18	55	22	0.6	0.63
d19	85	28	0.67	0.61
d20	30	10	0.67	0.7
d21	45	18	0.6	0.72
d22	131	27	0.79	0.68
d23	57	11	0.81	0.74
d24	34	5	0.85	0.74
d25	66	12	0.82	0.68
d26	83	10	0.88	0.7
d27	98	20	0.8	0.71
d28	89	20	0.78	0.73
d29	78	15	0.81	0.76
d30	44	10	0.77	0.63

Table 1: A comparison between the original documents and their associated summery for 30 multi-documents

Weight of a document calculated using the following equation:

$$Weight = \frac{1}{n}\sum_{i=1}^{n}(sentenceWeight) \quad (5)$$

where $sentenceWeight$ calculated from equation (1) and n is the total number of sentence in a document. Therefore, weight for both the original document and the summary is

calculated and the meaning amount conveyed by summary is obtained as follows:

$$MeaningAmount = \frac{summaryWeight}{originalDocumentWeight} \quad (6)$$

We have found that the system reduces the original text at an average to 28%. Meanwhile, it keeps in average 67% of the general meaning in the summarized version. This is reasonable because we are only interested in parts with high redundancy along the multi-document. From Table [1], the summary of multi-document (d26) and (d17) are the maximum and minimum reduction respectively.

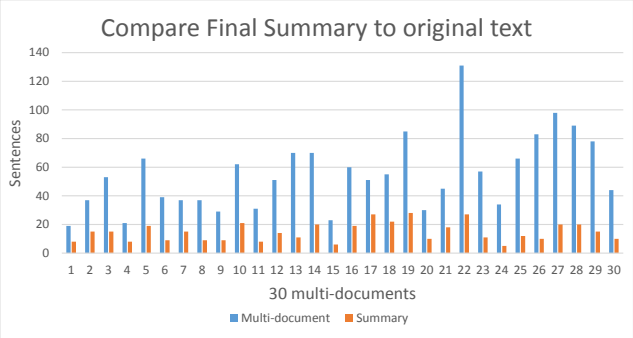

Fig. 4: The proportion of summary sentences to the original multi-document sentences.

V. Conclusion and Future Works

We have proposed a minimal language-dependent processing abstractive Arabic text summarization rule based model. This model depends on textual graph to remove text redundancy and constructs new sentences by concatenate related sentences together.

The proposed model consists of four stages namely; preprocessing, representing the multi-documents by directed weighted graph, traversing the graph and finally applying structural rules to generate summarized sentences.

The proposed model has achieved promising results for multi-document summarization. From the experiment using sample documents reached to 88% reduction ratio.

The future work could be include semantic process to enhance the summarization model. Also, we can add dictionaries, lexicons and ontologies to this model which maybe maximize the reduction ratio and could lead to generate highly readable and meaningful summary.

References

[1] E. Lloret and M. Palomar, "Analyzing the use of word graphs for abstractive text summarization," in *Proceedings of the First International Conference on Advances in Information Mining and Management, Barcelona, Spain*, 2011, pp. 61–6.

[2] A. B. Al-Saleh and M. E. B. Menai, "Automatic arabic text summarization: a survey," *Artificial Intelligence Review*, vol. 45, no. 2, pp. 203–234, 2016.

[3] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 573–580.

[4] K. Ganesan, C. Zhai, and J. Han, "Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions," in *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, 2010, pp. 340–348.

[5] H.-T. Zheng and S.-Z. Bai, "Graph-based summarization without redundancy," in *Web Technologies and Applications*. Springer, 2014, pp. 449–460.

[6] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, "Toward abstractive summarization using semantic representations," 2015.

[7] L. Bing, P. Li, Y. Liao, W. Lam, W. Guo, and R. J. Passonneau, "Abstractive multi-document summarization via phrase selection and," *arXiv preprint arXiv:1506.01597*, 2015.

[8] M. El-Haj, U. Kruschwitz, and C. Fox, "Exploring clustering for multi-document arabic summarisation," in *Information Retrieval Technology*. Springer, 2011, pp. 550–561.

[9] K. S. A. Harazin, "Multi-document arabic text summarization," Ph.D. dissertation, Islamic University, Gaza, Palestine, 2015.

[10] M. El-Haj and P. Rayson, "Using a keyness metric for single and multi document summarisation." Association for Computational Linguistics, 2013.

[11] H. Oufaida, O. Nouali, and P. Blache, "Minimum redundancy and maximum relevance for single and multi-document arabic text summarization," *Journal of King Saud University-Computer and Information Sciences*, vol. 26, no. 4, pp. 450–461, 2014.

[12] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1226–1238, 2005.

[13] S. S. Ismail, M. Aref, and I. F. Moawad, "A model for generating arabic text from semantic representation," in *2015 11th International Computer Engineering Conference (ICENCO)*. IEEE, 2015, pp. 117–122.

[14] M. Althobaiti, U. Kruschwitz, and M. Poesio, "Aranlp: a java-based library for the processing of arabic text," 2014.

[15] M. Diab, K. Hacioglu, and D. Jurafsky, "Automatic tagging of arabic text: From raw text to base phrase chunks," in *Proceedings of HLT-NAACL 2004: Short papers*. Association for Computational Linguistics, 2004, pp. 149–152.

[16] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, pp. 457–479, 2004.

[17] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts." Association for Computational Linguistics, 2004.

[18] E. Lloret and M. Palomar, "Text summarisation in progress: a literature review," *Artificial Intelligence Review*, vol. 37, no. 1, pp. 1–41, 2012.