Comparing PMI-based to Cluster-based Arabic Single Document Summarization Approaches

Madeeh Nayer El-Gedawy
Computer Center
Institute of Public Administration (IPA) – Jeddah

***Abstract*—In this paper, two extractive techniques are applied to handle Arabic Single Document Text summarization problem (SDS); the first uses a K-Means clustering approach and the other uses mutual information (MI) which is broadly used to measure the co-occurrence between two words in text mining. A successful Arabic document summarization algorithm should identify noteworthy sentences in the documents as accurately as possible. The terms used in the document (the distinct words) represent the document's identity, and instead of Bag of Words (BoW); a Term-Sentence Matrix (TSM) is utilized. In the first approach, the text themes are extracted using K-Means then one sentence per Cluster is chosen to be part of the summary using TFIDF weights. In the other approach, the pointwise mutual information (PMI) is used to assign weights for each cell in the TSM. The matrix generated from this TSM, is used to extract a summary of the document. experimentations prove that the cluster-based methodology performs slightly better than the first one, but if the end user could tweak the summary percentage to appropriate level then, the PMI-based approach will be slightly better.**

***Keywords*—Text Summarization, PMI, K-Means, Khoja Stemmer, Similarity Measures, TFIDF, Pre-processing, Clusters, Sentence Ranking.**

I. INTRODUCTION

According to an IDC report [1], the information all over the world is getting twice every two years. In 2011, the information around the globe was mostly 1.8 millions of petabytes. By 2020 the world will produce 50 times the amount of information while IT specialists will grow nearly to 1.5 times. It will be impossible for humans to manually summarize these huge repositories of textual data; so new methodologies for automatic text summarization is urgently needed.

Text summarization [2] is the process of transforming the original text into a shorter abridged one that preserves its meaning. Text summarization methods can be classified into extractive and abstractive summarization [3] [4]. An extractive summarization method selects significant sentences from the original document. An abstractive summarization method constructs new sentences based on understanding the original. A summary can be generic or user-focused. A generic summary tackles all themes detected in the original document. A user-focused highlights specific themes based on a query-oriented methodology. Experiments [5] have shown that summaries containing 20% or 30% of the original document could be effective reflection of the text.

In this paper, two methods are compared, both are very well candidate approaches to solve the problem of SDS. One method is based on Pointwise Mutual Information (PMI); Mutual information (MI) which is broadly used to measure the co-occurrence between two words; a high PMI score refers to a frequent item set (word pair). Knowing item sets is useful for many text mining applications such as lexical substitution [6] and feature selection [7]. The other method is an implementation for K-means clustering algorithm [8]. Due to the complex nature of the Arabic language, it is noteworthy to give appropriate consideration to the pre-processing phase as it may affect the results as shown in the experimentation section.

The rest of this work is organized as follows. Section 2 presents some related work. Section 3 describes the full methodology and procedures used. Experimentations are presented in section 4, and finally, conclusions are stated in section 5.

II. RELATED WORK

Extractive summaries are formulated by extracting significant text sentences from the text, based on statistical analysis of surface features such as term and N-gram frequencies, sentence location in the text and cue. Thus, as the “most frequent” or the “most favourably positioned” content are proxies for the “most significant”.

Text summarization procedure [9] can be divided into two phases: pre-processing phase and processing phase. pre-processing includes: tokenization, stop word removal and stemming. In Processing phase, features influencing the choice of sentences are selected and then weights are

assigned to these features. Final score of each sentence is determined using a feature-weight formula. Top ranked sentences are selected for a final summary.

There are some challenges facing the extractive summary [10] approach, such as:
1. Some sentences are much longer than the others. Therefore, summaries could include some text which is not significant enough.

2. Important information is usually spread all over the text sentences, and usually summaries will not capture all these important fragments.

3. Statistical summarization could lead to dangling anaphora problem [11]; where the summary has a pronoun that refers to something missing.

There are many approaches for extracting summaries from texts [12] [13], such as:

A. Term Frequency-Inverse Document Frequency (TF-IDF) method:
Excellent for query-based summaries. At first, Bag-of-words (BOW) is built (sentence level) and the traditional term frequency and inverse sentence frequency is used, where sentence frequency is the number of sentences in the document that contain that term. The sentences are scored by comparing similarity to the query and those of highest scores are chosen.

B. Cluster based method:

The text to be summarized addresses some topics or themes. By clustering the sentences and dividing them into bundles; these themes are extracted. A good summary must address all the themes addressed in the text.

C. Machine Learning approach

Given a set of training texts and their corresponding abstracts, the summarization procedure is treated as a classification task (CT): sentences are classified as either summary sentences or non-summary sentences. In the testing phase, the classification probabilities learnt in the training data are used to detect probable sentences that could be included in the summary.

D. LSA Method

Singular Value Decomposition (SVD) or Latent Semantic Analysis (LSA) can find principal orthogonal dimensions of multidimensional data. Terms that usually exist in related contexts are also positioned in the same singular space. LAS could be used to extract the topic-words and content-sentences from documents.

III. METHODOLOGY

The methodology is divided into 2 phases: preprocessing and processing.

A. Pre-processing

For preprocessing, we followed these 3 procedures:

1) Normalization:
Some extra letters correspond to different forms of some specific letters of the alphabet. These letters contain: various forms of alef, their presence depends on the morphology and the context of the word, but they are usually exchanged by mistake. To solve this problem, the following rules listed in the next table are applied for normalizing the Arabic tokens [14].

TABLE 1
Normalization rules

Rule	Example
Tashkeel	removed
Tatweel	Aaaaaaaaalah> Allah
Hamza	ؤ ىء ئ -> ء
Alef	ا أ إ <- آ
lamalef	لآ لأ لإ <- لا
yeh	ي ى -> ي
heh	ة <- ة or ه

2) Stop Word List Removal:
In this task, functional words that do not add any useful meaning to the analysis such as pronouns, auxiliary verbs, prepositions and determiners are removed. An approach that depends on entropy is developed as explained in [15]. The dataset used for extracting the stop word list is a newspaper corpus that contains 1,000 from El-Shorouk electronic newspaper website1 which are written in modern standard Arabic (MSA). These articles are fetched through a .NET library for screen scraping named NetScraper2. In a nutshell, the entropy-based works in two steps as follows:
Step 1: Word frequency is the number of times a word appears in a document. The list is sorted in descending order of frequency.
Step 2: we measure the likelihood Li,j of the term wj in document Di:

$$L_{i,j} = \frac{\text{frequency in the document Di}}{\text{the total number of words in document Di}}$$

Then we calculate entropy that measures the information value of the word w_j:

$$H(w_j) = \sum L_{i,j} * \log(1/L_{i,j})$$

[1] http://www.shorouknews.com/columns/
[2] http://netscraper.codeplex.com/

3) Stemming:

Stemming is meant to remove the inflections decorating the root or the stem, there are two approaches used for stemming: aggressive stemming and light stemming, aggressive stemming tries to reach the root of the word while the light stemming tries to find the fewest letters of the word that are sufficient to keep the word meaning. In this research, the Khoja [16] stemmer is used; it is a very well-known aggressive stemmer; it removes all diacritics, determiners, punctuation marks, the conjunction prefix 'waw' and numbers. All words are then checked against its exhaustive list of prefixes and suffixes, if there is a match, the longest match will be cut, finally the word is compared to some patterns, if there is a match, and then the root is determined. The Khoja stemmer could be downloaded from here3.

B. Processing Phase

1) PMI-based approach:

Sentences Representation:

A term sentence matrix (TSM) is constructed where there are w rows (unique words) and t columns (sentences); each cell measures the importance of the word within each sentence; the initial values are set to the frequencies.

Weighing and Ranking:

The modified weights for the TSM is based on pointwise mutual information which denotes the probability that x and y co-occur. In the TSM matrix, an element Mxy (the Xth word in Yth sentence); the PMI-based calculations will be:

$$PMI\ (X,Y) = \log\left(\frac{\frac{f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)}}{\frac{\sum_{x=1}^{n} f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)} * \frac{\sum_{y=1}^{t} f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)}}\right) \quad [17]$$

Where $\frac{f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)}$ is the probability that word x exists in sentence y; $\frac{\sum_{x=1}^{n} f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)}$ is probability that word x exist in the whole text; $\frac{\sum_{y=1}^{t} f(X,Y)}{\sum\sum_{x=1,y=1}^{t,n} f(X,Y)}$ is probability of sentence y within the whole text. The last term is multiplied by the whole PMI (X,Y) to get the relative rank of sentence y. the sentences are arranged in order according to this rank.

2) Clustering approach:

Sentences Representation:

The initial values of the TSM cells are set to the TFISF (Term Frequency Inverse Sentence Frequency).

Similarity Measures, Clustering and Ranking:

Three similarity measures have been tested: Euclidean distance, Jaccard distance and Cosine similarity [18] [19] [20]. The similarity is measured between sentences. These are the 3 similarity measures formulas used:

$$D(SA,SB) = \sqrt{\sum_{t=1}^{n} |W(t\ in\ A) - W(t\ in\ B)|} \quad \textit{(Euclidean)}$$

$$Cos(SA,SB) = \frac{\overrightarrow{SA}.\overrightarrow{SB}}{|\overrightarrow{SA}| \times |\overrightarrow{SB}|} \quad \textit{(Cosine Similarity)}$$

$$Jac(SA,SB) = \frac{\overrightarrow{SA}.\overrightarrow{SB}}{(|\overrightarrow{SA}|)^2 + (|\overrightarrow{SB}|)^2 - \overrightarrow{SA}.\overrightarrow{SB}} \quad \textit{(Jaccard Similarity)}$$

Where SA and SB are sentences a and b; W(t in A) is the weight (TFISF) of the term t in sentence a where there are number of terms from 1 to n. to fit in the algorithm Cosine and Jaccard similarities are converted to distance measures by calculating the new distance as (1-Cos) or (1-Jac). K-Means clustering algorithm was used as implemented by Apache Mahout software4; a detailed description of K-Means algorithm is explained in [21]. This paper depended on next equation to control the number of required clusters based on word distribution:

$$K = n\ \frac{|D|}{\sum_{i=1}^{n} |Si|} \quad [22]$$

In other words, it means that K clusters is equal to the number of sentences multiplied by the total number of terms in the document divided by the aggregated number of terms in sentences individually.

Lastly, one sentence is picked out of each of the clusters. To choose the best sentence within a cluster to be included in the summary; the sentences are given a rank value based on this equation as proposed in [23]:

$$Rank\ of\ Sx\ in\ Ck = \frac{1}{|CK|} \sum Dissimilarity(sx,sy)$$

That means to get the rank of sentence x in centroid k, the dissimilarity between this sentence and other sentences within the same centroid must be calculated and summed.

[3] http://sourceforge.net/projects/arabicstemmer/

[4] https://mahout.apache.org/

IV. EXPERIMENTATIONS

A. *Dataset*

For experimentation, Essex Arabic Summaries Corpus (EASC) is used5; it is an Arabic corpus that contains 153 Arabic articles and their associated summaries. It is used because it fits for the purpose of testing the single document summarization task.

B. *Experimentation setup*

For comparing the two approaches discussed in this paper; the F-measure is used; where each of the corpus articles summary is considered the reference one to which the suggested summary is compared. The F-measure is calculated as:

$$F = \frac{2PR}{P + R}$$ [6]

Where the Precision (P) is calculated as:

$$P = \frac{|Summary\ reference\ \cap Summary\ suggested|}{Summary\ suggested}$$

And Recall (R) is calculated as:

$$R = \frac{|Summary\ reference\ \cap Summary\ suggested|}{Summary\ reference}$$

For each run, three configurations have been considered: the first without removing stop word list and without using the Khoja stemmer (NoWL_NoStem), the second without the stemmer but stop words have been removed (WL_NoStem) and finally both stop word list and stemming have taken place (WL_Stem).

C. *Experimentation Results*

The comparison result between the two methods are shown in table 2. As shown, there are three variations for the cluster-based approach due to trying three different distance measures. For the two methods compared to be coherent, the number of sentences chosen for the summary, according to the PMI-based score is set to the same number of sentences resulted from the cluster-based approach which is set to the number of clusters. In table 3, we have tried to set the number of sentences contained in the summary using the PMI-based approach to the same number of sentences contained in the summary reference, to check if this will improve the results or make even worse; as the percentage of summarized text could be a parameter controlled by the end user.

TABLE 2
Comparison Matrix using F-measure

Method	NoWL_NoStem	WL_NoStem	WL_Stem
PMI-based	0.42	0.43	0.44
Cluster-based (Euclidean)	0.46	0.48	0.47
Cluster-based (Cosine)	0.46	0.47	0.47
Cluster-based (Jaccard)	0.45	047	0.46

TABLE 3
PMI-based modified by setting number of summarized sentences to number of human summarized sentences

NoWL_NoStem	0.46
WL_NoStem	0.49
WL_Stem	0.48

V. CONCLUSION

The cluster-based approach gives better results than those of PMI-based. The difference is not big. Moreover, stemming and removing stop word list make the PMI-based summarization results better, on the other hand stemming has a negative effect on the cluster-based summarization results. Euclidean distance could give better results than both Cosine and Jaccard similarities. If the PMI-based summarization is guided by the end user; the results could be promising; actually, the results could be better of those obtained by the cluster-based approach.

REFERENCES

[1] John Gantz and David Reinsel, Ext ract ing Value from Chaos, I D C I V I E W, EMC Corporation, 2011.
[2] Karel Jezek and Josef Steinberger, "Automatic Text summarization", Vaclav Snasel, pp. 1-12, 2008.
[3] Farshad Kyoomarsi, Hamid Khosravi, Esfandiar Eslamiand Pooya, and Khosravyan Dehkordy, "Optimizing Text Summarization Based on Fuzzy Logic", In proceedings of Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE, UK, pp. 347-352, 2008.
[4] G. Erkan and Dragomir R. Radev, "LexRank: Graph-based Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research, Re-search, vol. 22, pp. 457-479, 2004.
[5] A. Morris, G. Kasper, and D. Adams. "The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance". Information Systems Research, vol. 3, pp. 17–35, 1992.
[6] Dagan, O. Glickman, A. Gliozzo, E. Marmorshtein and C. Strapparava, "Direct Word Sense Matching for Lexical Substitution". In Proc. of COLING/ACL-06, pp. 449-456, 2006.
[7] Y. Liu, "A Comparative Study on Feature Selection Methods for Drug Discovery", J. Chem. Inf. Comput. Sci., vol. 44, pp. 1823-1828, 2004.
[8] Harshal J. Jain, M. S. Bewoor, and S. H. Patil, "Context Sensitive Text Summarization Using K-Means Clustering Algorithm", International Journal of Soft Computing and Engineering, vol. 2, May 2012.

[5] http://sourceforge.net/projects/easc-corpus/

[6] http://en.wikipedia.org/wiki/Precision_and_recall

[9] Madeeh Nayer El-gedawy, "Using Fuzzifiers to solve Word Sense Ambiguation in Arabic Language", International Journal of Computer Applications, vol. 79, October 2013.
[10] Jackie Cheung, "Comparing Abstractive and Extractive Summarization of Evaluative Text :Controversiality and Content Selection", B. Sc. (Hons (.Thesis in the Department of Computer Science of the Faculty of Science, University of British Columbia, 2008.
[11] Josef Steinberger, Massimo Poesio, Mijail A. Kabadjov, and Karel Jeek, "Two uses of anaphora resolution in summarization", Information Processing and Management: an International Journal, vol.43, pp.1663-1680, November 2007.
[12] Ani Nenkova and Kathleen McKeown. Mining Text Data, A Survey of Text Summarization Techniques, Springer, pp 43-76., 2012.
[13] Vishal Gupta and Gurpreet Singh Lehal. "A Survey of Text Summarization Extractive Techniques", JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 2, AUGUST 2010.
[14] Madeeh Nayer El-gedawy. " Orthogonal Processing for Measuring the Tonality of Egyptian Microblogs". International Journal of Computer Applications, vol. 87, pp. 20-25, February 2014.
[15] Zhou Yao and Cao Ze-wen. "Research on the Construction and Filter Method of Stop-word List in Text Preprocessing", Proceedings of the 2011 Fourth International Conference on Intelligent Computation Technology and Automation, vol. 1, 2011.
[16] Shereen Khoja and Roger Garside. "Stemming Arabic text". Computer Science Department, Lancaster University, Lancaster, UK, http://www.comp.lancs.ac.uk/ computing/users/khoja/stemmer.ps, 1999.
[17] S Aji and Ramachandra Kaimal. "DOCUMENT SUMMARIZATION USING POSITIVE POINTWISE MUTUAL INFORMATION". International Journal of Computer Science & Information Technology (IJCSIT), vol 4, April 2012.
[18] Anna Huang, "Similarity Measures for Text Document Clustering", NZCSRSC, Christchurch ,New Zealand, April 2008.
[19] Michael Steinbach, George Karypis, and Vipin Kumar, "A comparison of document clustering techniques", In KDD Workshop on Text Mining, 2000.
[20] Hanane Froud, Abdelmonaime Lachkar, and Said Alaoui Ouatik, "Arabic text summarization based on latent semantic analysis to enhance Arabic documents clustering", International Journal of Data Mining & Knowledge Management Process (IJDKP), 2013.
[21] Madeeh Nayer El-Gedawy, "TARGETING POVERTY IN EGYPT USING K-MEANS ALGORITHM", IDSC-WPS, Working Paper No. 12, 2010.
[22] RM Aliguliyev, "A new sentence similarity measure and sentence based extractive technique for automatic text summarization", Expert Systems with Applications, vol. 36, pp. 7764-7772, 2009.
[23] M. Pavan, M., and M. Pelillo, " Dominant sets and pairwise clustering", IEEE Transactions on Pattern Analysis and Machine Learning, vol. 29, pp. 167 – 172, 2007.