

# A Trainable Arabic Bayesian Extractive Generic Text Summarizer

Ibrahim Sobh<sup>1,2</sup>, Nevin Darwish<sup>1</sup>, Magda Fayek<sup>1</sup>

<sup>1</sup>The Department of Computer Engineering, Cairo University, Giza, Egypt.

<sup>2</sup>The Research & Development International Company (RDI@).

## Abstract

Summarization is the process of producing shorter presentation of the most important information from a source or multiple sources of information according to particular needs. Summarization is not applied only on text documents but also on any multimedia facility. This paper introduces Bayesian method for Arabic extractive text summarization. We developed a trainable summarization program that based on manually labeled corpus and Bayesian classification. System is evaluated in terms of recall, precision and F-measure.

**Keywords:** Extractive summary, Bayesian classification, Training corpus, Arabic documents.

## 1. Introduction

The process of summarization is becoming very important in the presence of large number of information sources available in every field. Summarization work has been started as early as in the 1950's. Edmundson presents a survey of the existing methods to automatic summarization in [1] and a systematic approach to summarization which forms the core of the extraction methods even today in [2]. *Extractive* summarizers extract text by selecting from original document important pieces to produce shorter result. Human summaries often rely on cutting and pasting of the full document to generate summaries. By decomposing human summary, we can learn the kind of operations which are usually performed to extract and edit sentences and then develop automatic programs to simulate the most successful operations. A Hidden Markov Model solution to the decomposition problem is proposed in [3] and it founds that 78% of summary sentences produced by humans are based on cut-and-past. Granularities of extraction could be phrases [4, 5] and sentences [6, 3]. *Abstraction*, on the other hand, generates summaries at least some of whose material is not present in the input text. Abstraction of documents by humans is complex to model as is any other information processing by humans. The abstracts differ from person to person, and usually vary in the style, language and detail. The process of abstraction is complex to be formulated mathematically or logically [3]. Dealing with *multiple documents* summarization is challenging taking into consideration the possible large input corpus and the possibility of repeated and/or conflicted data.

Summary can be used to be *indicative* to produce a reference function to select documents for more in-depth reading or *informative* to cover all or most salient information in the source text documents. Summary can be *general* where there is no focus on some topic or view point provided by the user or it can be *user-focused* where summaries, are guided by user view point statement, topic or question to be answered. Summarization can make use of document structure (titles, subtitles, table of content, etc.) and layout (font size, boldness, underline, etc.) to produce more relevant shorter text. Size of produced summary can be very shot (*Headline*) or relatively short typically 20% to 25% of original document size. Hand-held devices such as personal digital assistants (PDA) and cell-phones provide an interesting application for summarization technologies due to limited screen size.

Summary evaluation is a challenging process because there in no one ideal correct answer and it depends on the purpose of the summary.

Bayesian classification approach for Arabic text extractive generic summarizer is presented in section 2 including features, classifier and corpus. System evaluation and results are discussed in section 3. Conclusions and future work in section 5.

## 2. System Structure

Typically extractive summarizers compute a score for each sentence in the original document and then select the highest scoring sentences as summary. Rules of scoring are heuristic; however given a training corpus it would be possible to approach the problem as statistical classification to classify a sentence to be in summary or out of summary given its feature vector. Kea system implemented in [5] was used to extract keyphrases using naïve Bayesian algorithm for classification. Kea system was evaluated against author specific keyphrases. Similar technique could be applied to extract sentences for summarization.

The proposed system structure requires sentence feature, classification method and training corpus to be identified.

### 2.1 Arabic Stemming

An important step in summarization process is stop word removal and stemming. Arabic as high inflected language requires good stemming for information retrieval and summarization. There is a choice between word roots or

stems as the desired level of analysis. Different approaches for Arabic stemming can be identified, manually constructed dictionaries, algorithmic light stemmers which remove prefixes and suffixes, morphological analyzers which try to find the roots and forms of words. Stemmers can be weak, fail to conflate related forms that should be grouped together, or strong, where unrelated forms are conflated. [9] introduced an Arabic stemmer and a list of 168 stop words. Implementation of [8] is used in this paper for extracting roots and stop word removal.

## 2.2 The Features

The input document is parsed into sentences. Each sentence is parsed into words. Feature vectors are extracted for each sentence. Term Frequency times Inverse Document frequency (*tf-idf*) is commonly used in information retrieval systems to assign weights to terms in a document and used by [10, 5] to assign weights to keyphrases. Similar concept is used here to assign weight to sentence. Distance of the phrase from document start feature is used by [5]. Sentence location in document is considered important features in [10, 6]. Sentence location in document feature is expanded to the location in the paragraph that the sentence belongs to. Also paragraph length is considered. Used features are:

**Sentence Weight:** After stop word removal, each word is transformed into its root as stemming option. Then the frequency for each root is computed in the current document. For each sentence the summation of non stop word frequencies is computed and normalized.

**Sentence Length:** Is the number of the words in a sentence after removing stop words. This feature is normalized making the length relative to the longest sentence in the current document.

**Sentence Absolute Position:** Is the order of the sentence in the document. This feature is normalized where the maximum value is one for the first sentence in the current document.

**Sentence Paragraph Position:** Is the normalized order of the sentence in the paragraph in which the sentence located in.

**Sentence Paragraph Length:** Is the normalized length of the paragraph in terms of number of sentences in which the sentence is located in.

All normalized feature vectors are converted into discrete six values from zero to five in order to simplify the Bayesian classifier.

## 2.3 The Classifier

The Bayesian classifier will classify each sentence to be in summary or out of summary classes based on its feature vector and a training corpus. For each sentence the probability it will be included in summary can be computed as follows:

$$P(s \in S | V_1, V_2, \dots, V_n) = \frac{P(V_1, V_2, \dots, V_n | s \in S)P(s \in S)}{P(V_1, V_2, \dots, V_n)}$$

Where  $s$  is the sentence,  $S$  is the Summary class,  $V$  is the feature vector and  $n$  is the number of features. Assuming that features are statistically independent:

$$P(s \in S | V_1, V_2, \dots, V_n) = \frac{\prod_{i=1}^n P(V_i | s \in S)P(s \in S)}{\prod_{i=1}^n P(V_i)}$$

$P(V_i | s \in S)$  and  $P(s \in S)$  can be estimated directly from the training corpus.  $P(V_i)$  is a normalization factor. The sentence is classified into summary class if the following condition is fulfilled:

$$\prod_{i=1}^n P(V_i | s \in S)P(s \in S) > \prod_{i=1}^n P(V_i | s \in NS)P(s \in NS) + \alpha$$

Where  $NS$  is the non summary class, and  $\alpha$  is a safety threshold or confidence score typically equals to zero. Positive  $\alpha$  will produce less sentences in summary class with increased precision and confidence score, however negative  $\alpha$  will produce more sentences in summary class with increased recall.

## 2.4 The Corpus

The corpus is collected from the BBC<sup>1</sup> Arabic recent Middle East news. The documents are in plane text. The total corpus size is 51 documents divided into training set 46 documents and testing set 5 documents. The corpus is processed by a hand labeling tool figure 1, where each document is parsed into sentences. Each sentence is represented into a single line. An Arabic language specialist is then asked to select the most important sentences in the document. Number of selected sentences for each document is left to the language specialist this assumes to increase the generality of the classifier. Selected sentences are labeled as in summary class; unselected sentences are labeled as not in summary class and features vectors are calculated for all sentences.

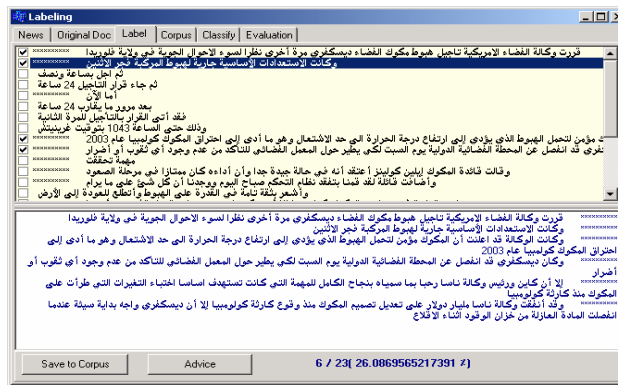


Figure 1. Labeling tool screen caption

<sup>1</sup> [http://news.bbc.co.uk/go/rss/-/hi/arabic/middle\\_east\\_news](http://news.bbc.co.uk/go/rss/-/hi/arabic/middle_east_news)

### 3. System Evaluation

There are several serious challenges in evaluating summaries. Summarization involves a machine producing output that results in natural language communication. If a summary was performed to answer a question then there may be correct answer, otherwise it will be somehow hard to tell. Human judges on summary are very expensive and hence an automated process is required to evaluate summaries. Classification approach for summarization makes it easier for evaluating extractive summaries. Two important measures are used, precision and recall [11, 7]. Precision is a measure of how much of information that the system returned is correct.

Precession = Number of system correct summary sentences / Number of system summary sentences

Recall is a measure of the coverage of the system.

Recall = Number of system correct summary sentences / Total number of summary sentences

Recall and precision are antagonistic to one another. A system strives for coverage will get lower precision and a system strives for precision will get lower recall. F-measure balances recall and precision using a parameter  $\beta$ . The F-measure is defined as follows:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

When  $\beta$  is one, Precision P and Recall R are given equal weight. When  $\beta$  is greater than one, Precision is favored, when  $\beta$  is less than one, recall is favored.

In the following experiments  $\beta$  equals one. Since the corpus available is small, a *cross validation* strategy is used [6]. The corpus is divided into training set (90% of the corpus) and testing set (10% of the corpus). The testing is repeated ten independent rounds, each round with different testing set considering the rest of the corpus as training set. Table 1 shows each round results details.

Round	Training set Size	Recall	Precision	F-measure
1	93.6 %	0.678	0.556	0.611
2	91.3 %	0.625	0.606	0.615
3	88.5 %	0.782	0.818	0.800
4	90.3 %	0.8	0.667	0.727
5	92.0 %	0.969	0.667	0.790
6	94.2 %	0.884	0.696	0.779
7	85.3 %	0.586	0.79	0.673
8	91.9 %	0.909	0.625	0.741
9	84.7 %	0.667	0.612	0.638
10	89.9 %	0.925	0.77	0.840
<b>Average</b>	90.2 %	0.7825	0.6807	0.721
<b>Dev</b>	3.207 %	0.137	0.087	0.083

Table 1. System Results

According to F-measure, the best was round 10 and the worst was round 1. On average the system produces

78.25% of human selected sentences. On the other hand, the system average precision is 68.07%. Figure 2 shows a screen caption of the Bayesian classification tool.

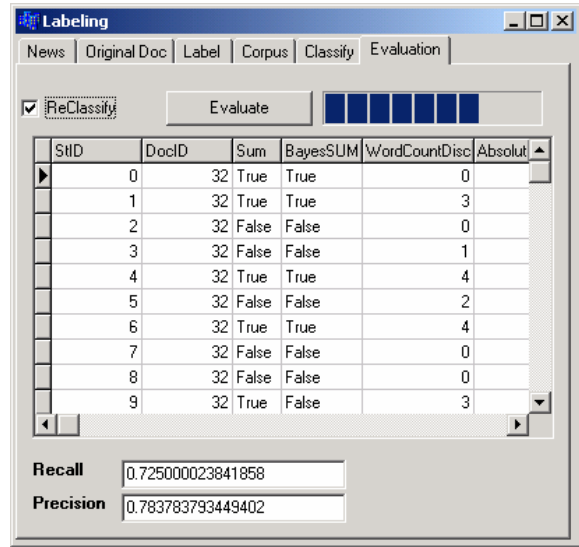


Figure 2. Bayesian classification tool screen caption

Table 2 shows how performance varies as features are successively combined together in order. Testing is performed on round four where it has the closest F-measure to the average of the system.

Features	Recall	Precision	F-measure
Weight	0.675	0.729	0.701
Length	0.8	0.639	0.710
Absolute Position	0.824	0.647	0.725
Paragraph Position	0.825	0.647	0.725
Paragraph Length	0.8	0.667	0.727

Table 2. Accumulative features Results

#### 4.1 Ad-hoc System

Ad-hoc system was implemented to generate summaries. The system uses heuristic scoring function for ranking sentences. Then the system selects the highest scores of  $m$  sentences to be in summary. Then the system re-orders the selected sentences as they appear in the original document. The scoring function is as follows:

$$Score = \sum_{i=1}^n w_i F_i$$

Where  $n$  is the number of features,  $w_i$  is the weight for feature  $i$ ,  $F_i$ . Weights can be positive, negative or zero according to how the feature will influence the final score of the sentence. Sentence length, sentence order in document, and sentence paragraph length features were used in the ad-hoc system. Heuristic weights were given for each feature. Then the ad-hoc system was asked to produce summaries of size 25% of the original documents which is the same percentage found in the corpus. Table 3

shows four versions of ad-hoc system with the different weights assigned to corresponding features.

Feature	ad-hoc1	ad-hoc2	ad-hoc3	ad-hoc4
Length	1	0	1	1
Position	-1	1	1	1
Paragraph Length	1	-1	0	1

Table 3. *Ad-hoc systems*

**ad-hoc1:** Prefers long sentences, sentences that come at the end of a document, and sentences that belong to short paragraph.

**ad-hoc2:** Prefers sentences that come at the start of a document, and sentences that belong to long paragraph. Sentence length is ignored.

**ad-hoc3:** Prefers long sentences, sentences that come at the start of a document. Sentence paragraph length is ignored.

**ad-hoc4:** Prefers long sentences, sentences that come at the start of a document and sentences that belong to short paragraph.

Figure 3 shows a comparison between ad-hoc systems and the Bayesian classification system performances in terms of F-measure.

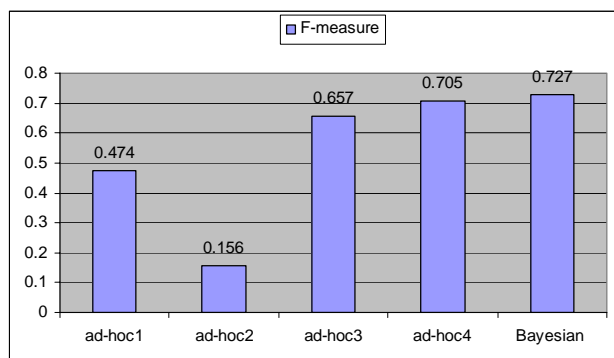


Figure 3. *Systems Comparison*

## 5. Conclusions and future work

In this paper, a trainable Bayesian approach for Arabic extractive text summarization has been introduced. On average the system produces 78.25% of human selected sentences; system average precision is 68.07%; these results are considered acceptable for a wide range of applications. The trainability feature of the system makes it possible to be customized for specific domains. System performance overcomes four ad-hoc systems. System performance was increasing when combining sentence weight, sentence length, and sentence absolute position. Addition of sentence paragraph position and sentence paragraph order results in slight change in system performance, this due to the fact that most of paragraphs in the corpus are of length only three or less sentences and hence the paragraph features are not discriminative enough.

Final results showed a very good potential for improvements. Number of techniques can be applied to enhance the results. Arabic word stemming that depends

on Arabic stem (root + form) [8] instead of using only the root is expected to improve the sentence weight feature contribution. Using similarity measure between sentences to reduce redundancy also is expected to be powerful where humans tend to produce summaries with minimum redundancy. Cosine similarity measure can be applied between sentences; a sentence with minimum similarity measure will be candidate to be in summary. Larger corpus will enhance the overall system precision and recall. Selecting more features like user defined key words, or indicator phrases will increase the system controllability. Adding semantic information from comprehensive lexical resource such as WordNet [12] but for Arabic language will enhance output cohesion.

## 6. Acknowledgments

Special thanks are posed to The Research & Development International Company (RDI®) for its support. We must also mention valuable efforts of Natural Language Processing technology and linguistic support teams at RDI®.

## References

- [1] Edmundson, H.P. and R.E. Wyllys, "Automatic Abstracting and Indexing-Survey and Recommendations". Communications of the ACM, 4(5): p. 226-234, 1961.
- [2] Edmundson, H.P., "New Methods in Automatic Extracting". Journal of the ACM, 16(2): p. 264-285, 1969.
- [3] Jing, H. and K.R. McKeown, "The Decomposition of Human-Written Summary Sentences". In proceedings of SIGIR'99, University of Berkely, CA, USA
- [4] Tureny, P.D., "Learning Algorithms for Keyphrase Extraction", Information Retrieval, 2(4), p. 303-336, 2000
- [5] Witten, I.H., Paynter, G.W., Frank E., Gutwin, C., and Nevill-Manning, C.G., "KEA: Practical Automatic Keyphrase Extraction" Department of computer science, The University of Waikato, 2000
- [6] Kupiec, J. , Pederson, J. O., Chen, F. "A Trainable Document Summarizer" In proceedings of the 18th SIGIR' 95 Conference, Association of Computing Machinery, p. 68-73 , 1995.
- [7] Steve J. Stephen L. and Gordon W. "Interactive Document Summarization Using Automatically Extracted Key phrases" Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35), 2002
- [8] Mohammed Atteya. "A Large-Scale Computational processor of the Arabic Morphology, and Applications" A Master's Thesis, Faculty of Engineering, Cairo University, Egypt, 2000
- [9] Khoja, S. and Garside, R. "Stemming Arabic Text" Computing Department Lancaster University, Lancaster, 1999
- [10] Chikashi Nobata, Satoshi Sekine "CLR/NYU Summarization" DUC-2004
- [11] Yihong Gong and Xin Liu "Creating Generic Text Summaries" Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01).
- [12] Miller, G. "WordNet: A Lexical Database for English." Communications of the Association for Computing Machinery (CACM) 38, 11, 39-41.