An Optimized Dual Classification System for Arabic Extractive Generic Text Summarization

Ibrahim Sobh[1, 2], Nevin Darwish[1], Magda Fayek[1]
sobh@rdi-eg.com ndarwish@eng.cu.edu.eg magdafayek@eng.cu.edu.eg
[1]The Department of Computer Engineering, Cairo University, Giza, Egypt.
[2]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. With summaries, we can make effective decisions and get useful information in less time. This paper introduces an Arabic extractive text summarization system. This system integrates Bayesian and Genetic Programming (GP) classification methods in an optimized way to extract the summary sentences. The system is trainable and uses manually labeled corpus. Features for each sentence are extracted based on Arabic morphological analysis and part of speech tags in addition to simple position and counting methods. Initial set of features is examined and reduced to an optimized and discriminative subset of features. Given human generated summaries, the system is evaluated in terms of recall, precision and F-measure.

Keywords: Extractive summary, Bayesian classification, Genetic Programming, Classification optimization, Supervised learning, 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 for automatic summarization [1] and a systematic approach to summarization which forms the core of the extraction methods even today [2]. Generally speaking, summaries could be extractive or abstractive.

1.1 Extractive summarizers

Extractive summarization extracts text by selecting from original document important pieces to produce shorter result. Human summaries often relay 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 was proposed [3] and it found that 78% of summary sentences produced by humans are based on cut-and-past. Granularities of extraction could be phrases (2 or 3 words) as in or sentences [3, 6]. Extraction approach may have the problem of coherence but they are trusted by the users. There are different approaches to implement extractive summaries. The most important ones are: the linear methods that give a score for each sentence depending on heuristic measures, Latent Semantic Analysis (LSA) which is inspired by latent semantic indexing and applying Singular Value Decomposition (SVD) to the document sentence matrix [7], Maximal Marginal Relevance (MMR) which measures the relevance or similarity between each sentence in the full document and the sentences that have been selected and added into the summary [8], and Graph Based methods that models the document into graph where sentences are the vertices, and Machine Learning Approaches [6].

1.2 Abstractive summarizers

Abstraction, on the other hand, generates summaries at least some of whose material is not presented in the input text. Abstraction of documents by humans is complex to model as is any other information processing by humans. The process of abstraction is complex to be formulated mathematically or logically [3]. Abstraction requires text analysis, modeling and language generation techniques.

1.3 Single and Multi-Document summarizers

Dealing with multiple documents summarization is challenging when taking into consideration the possible large input corpus and the possibility of repeated or conflicted information among different documents. A single document summary within a predefined domain is a simpler task. Another challenge is that the multi-documents also could be in different languages.

1.4 How Summarizers Differ

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. Size of produced summary (Compression Ratio) can be very short (*Headline*) or relatively short typically 20% to 25% of original document size.

1.5 Summary Evaluation

Summary evaluation methods attempt to determine how adequate and reliable or how useful a summary is relative to its source. Generally, there are two types of evaluation methods. The first is *intrinsic* evaluation in which users judge the quality of summarization by directly analyzing the summary. Users judge fluency, how well the summary covers stipulated key ideas, or how it compares to an ideal summary written by the author of the source text or a human abstractor. None of these measures are entirely satisfactory. The ideal summary, in particular is hard to construct and rarely unique. In most cases there is no only one correct ideal summary for a given document. The second type of evaluation methods is *extrinsic*. Users judge a summary's quality according to how it affects the completion of some other task, such as how well they can answer certain questions relative to the full source text. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is also used for summary evaluation by counting the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans

Extractive approach for summarization by classification enables us to use recall, precession and F-measure to evaluate summaries.

1.6 Paper Overview

The proposed system uses classification methods to generate general summaries by extracting sentences from a single document. The system structure is presented in section 2 including features, Bayesian classifier, GP classifier, classifiers integration and corpus details. System evaluation and results are discussed in section 3. Conclusions and future work in section 4.

2. Proposed Summarization System

Typically extractive summarizers deal with sentences. Rules of sentence scoring are generally 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 classes given its feature vector. Naïve Bayesian classification method is considered to be simple, easy to implement and dose not require heavy processing. However, it assumes the independence between features and it may fall into local optima. Naïve Bayesian classification method was used for *key-phrases* extraction [5]. Similar technique could be applied to extract sentences for summarization. Genetic Programming is used also for classification and could be used for extractive summarization. GP uses a beam search to try to find global optima. The proposed system uses both classification techniques and combines them in an optimized way to get better results using reduced feature set. The system structure requires labeled (annotated) training and testing corpus, sentence features and classification methods to be identified.

2.1 Arabic Preprocessing

Arabic as high inflected language requires good stemming for information retrieval and summarization applications. Feature extraction requires complex Arabic language processing: Stop words removal, Stemming and Part Of Speech Tagging (POST).

2.1.1 Stop words removal

Introduction of an Arabic stemmer and a list of 168 stop words were presented in [10]. We used the implementation of [11] as a more robust method for extracting stop words where Arabic words are analyzed and categorized into regular derivatives, irregular derivatives, fixed, Arabized or Transliterated. We considered that fixed words are stop words.

2.1.2 Stemming

The purpose of stemming is to cluster different words into shared groups depending on roots or stems (root and form). 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. We used the Arabic Morphological Analyzer [11] for extracting roots.

2.1.3 POS Tagging

POS Tags are also important because they enable recognition of important information about words for example whether they are nouns (single or plural, feminine or masculine, identified or not identified) or verbs (present or past) and so on. Implementation of [12] is used for extracting POS Tags.

2.2 The Proposed Features

The input document is parsed into a number of paragraphs and each paragraph is parsed into sentences. Each sentence is parsed into words. The document is assumed to be in plain text. The parsing process makes use of spaces, commas, parenthesis and new line for identifying words, sentences and paragraphs. 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 [5, 13] to assign weights to keyphrases. Similar concept is used here to assign weight to sentences. Distance of the phrase from the document start feature is used by [5]. Sentence location in document is considered important feature in [6, 13]. Here, sentence location in document feature is expanded to the location in the paragraph that the sentence belongs to. Also, paragraph length (which the sentence belongs to) is considered. Also we used the features in [14] as "basic features" and more additional features as "new features". The basic features are:

Feature 1: Sentence Weight: After stop word removal, each word is transformed into its root. 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.

Feature 2: 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.

Feature 3: 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 document.

Feature 4: Sentence Paragraph Position: Is the normalized order of the sentence in the paragraph in which the sentence is located in.

Feature 5: Sentence Paragraph Length: Is the normalized length of the paragraph in terms of number of sentences in which the sentence is located in.

In addition to the basic features more features are also proposed to take the advantage of the relations between sentences in terms of similarity and to use Part of Speech Tags information for the words in each sentence. The new features are:

Feature 6: Sentence Similarity:

The similarity feature of a sentence is computed as the next formula:

$$Sim_s = \sum_{i=1, i \neq s}^{n} Sim(S_i, S_s)$$

The similarity of sentence *S* is the summation of the similarities between sentence *S* and all other senesces in the document. The similarity is computed using the *cosine* method. The similarities are saved in a similarity matrix where:

- $A_{i,j} = Sim(S_i, S_j)$ Element (*i* , *j*) is the similarity between sentence *i* , *j*
- $A_{i.j} = A_{j,i}$ The similarity between sentence *i* and *j* is equal to the similarity between sentence *j* and *i*
- $A_{i.i} = 1$ The similarity between sentence and itself is 1

Then the values were normalized per document. The similarity between sentences method used in literature before as a non-supervised heuristic method to decide when to consider the sentence in summary or not [8]. Here we are proposing a new method for using similarity as a feature along with other features in a supervised learning approach.

Feature 7: Number of Infinitives:

Is the normalized number of infinitives in a sentence per document.

Feature 8: Number of Verbs:

Is the normalized number of verbs in a sentence per document.

Feature 9: Number of Identified:

Is the normalized number of identified words in a sentence per document.

Feature 10: Number of "Marfoa'at":

Is the normalized number of "Marfoa'at" words in a sentence per document. These are the words that come as a beginning in a noun sentence or as the doer or actor in a verbal sentence.

Features 7 to 10 are extracted by using the Part of Speech tagging [12]. However we found that this implementation has limitations for extracting Number of Identified and Number of "Marfoa'at", it extracts the most common but not all the cases.

After normalization, these features are converted into discrete six levels from zero to five in order to simplify the Bayesian classifier.

Feature 11: Is Digit:

A binary feature equals 1 if the sentence includes a digit and 0 otherwise. The digit could represent for example money, date, quantity or address.

2.3 The Classifiers

2.3.1 Bayesian classifier:
A Bayesian classifier classifies 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 that will be included in summary can be computed as follows:

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

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:

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

$P(V_i\,|\,s\in S)$ and $P(s\in S)$ can be estimated directly from the training corpus. $P(V_i)$ is a normalization factor so, 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 α is a safety threshold or confidence score typically equals to zero. Positive α will produce less sentences in summary class with increased precision, however negative α will produce more sentences in summary class with increased recall. In this paper we used α equals to zero.

2.3.2 GP classifier:
GP is automated learning of computer programs. Originally, Genetic Algorithms (GA) learning is inspired by the theory of evolution. Basically the problem is represented by genes. The first population of genes is initialized and then applying mutation and cross-over operators on the current population results in a new better population. A fitness function is used to evaluate how an individual fits and optimizes the problem. GP represents a problem as the set of all possible computer programs. A program is represented in a gene where GP uses cross-over and mutation as the transformation operators to change candidate solutions (programs) into new candidate solutions. GP uses a beam search where the population size constitutes the size of the beam and where the fitness function serves as the evaluation metric to choose which candidate solutions are kept and not discarded. Typically, a program can have a tree or linear structures. GP was used successfully in many fields for example, financial market, image processing, optimization, signal processing and pattern recognition.
In his 1975 book, Holland [15] mentioned AI as one of the main motivators for the creation of genetic algorithms GA. He did not experiment the direct use of GA to evolve programs. Two researchers, Cramer, 1985 [16] and Koza, 1989 [17] suggested that a tree structure should be used in a program generation in a genome. Koza however was the first to recognize the importance of the GP and demonstrated its feasibility for automatic programming in general. In Koza, 1989 he provided evidence in the form of several problems from five different areas. In his book, Koza, 1992, [18] he sparked the rapid growth of GP.
In this paper we choose to use the Discipulus[1] GP system. Discipulus is considered the world's first and fastest commercial Genetic Programming system. It writes computer programs automatically in Java, C, and Intel assembler code. Discipulus builds two types of models, Regression models and Classification models. We used the downloadable free version with default and recommended values for cross-over and mutation rates when running the tool for classification.

2.3.3 The Dual Classification System
There are many classifier combination topologies. Broadly speaking, there are the parallel (horizontal) and the sequential (cascading) combination methods. We selected an optimized and simple way for combining the two classifiers to get better results as follows:
Bayesian Classifier Union (OR) GP Classifier
Consider sentence in summary if any classifier agrees.

$$Class=Class_{Bayesian}\cup Class_{Genetic\,Programming}$$

[1] http://www.aimlearning.com

Bayesian Classifier Intersection (AND) GP Classifier
Consider sentence in summary if and only if both classifiers agree.

$$Class = Class_{Bayesian} \bigcap Class_{Genetic\,Programming}$$

2.4 The Corpus

The corpus is collected from the "Ahram"[2] web site. Recent "Egypt" and "Arabic Region" news were selected. The documents are transformed from HTML format into plain text. The total corpus size is 213 documents divided into training set (80%) and testing set (20%). The corpus is parsed into paragraphs and sentences. Each sentence is represented into a single line to an Arabic language specialist. Then the specialist is asked to select (check) the most important sentences in the document. Number of selected sentences for each document (Compression Ration) is left to the judgment of the language specialist as it depends on the document. This approach should increase the generality of the system by capturing (learning) the appropriate compression ratio. Selected sentences are labeled as in summary class; unselected sentences are labeled as out of summary class and features vectors are extracted for all sentences. Total number of sentences is 4899 sentences. (Average 23 sentences per document). The human summary size in the training set is 23.3%.

3. System Evaluations and Results

Classification approach for summarization makes it easier for evaluating extractive summaries. Three important measures are commonly used, precision, recall and F-measure [9, 21]. 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 human 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 β. The F-measure is defined as follows:

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

When β is one, Precision P and Recall R are given equal weight. When β is greater than one, Precision is favored, when β is less than one, recall is favored. In the following experiments β equals one. Our target is to have large F-measure and at the same time produce a reasonable summary size according to the training set. The (F-Measure/summary size) ratio is important when comparing systems.
Table 1 shows the result when using all the eleven features for the Bayesian classification and GP classification independent and integrated. The GP is calculated after 9.7 million programs experiment.

System	Recall	Precision	F.Measure	Summary Size
Bayesian	0.711	0.489	0.579	33.99%
GP	0.479	0.716	0.574	15.61%
AND	0.464	0.721	0.565	15.06%
OR	0.725	0.490	0.585	34.55%

Table 1. *Eleven Features Classification Results*

3.1 Features reduction

After studying the probability distribution for the features independently across summary and out of summary classes based on *Figure-1* [19], we found that some of the features are not discriminative due to overlapping across the classes while other features are promising. Guided by these experiments we are able to categorize the features into groups depending on its strength. The following features seem to be not discriminative (*weak*):

- Feature 3: Sentence Absolute Position
- Feature 10: Number of "Marfoa'at"
- Feature 11: Is Digit

[2] http://www.ahram.org

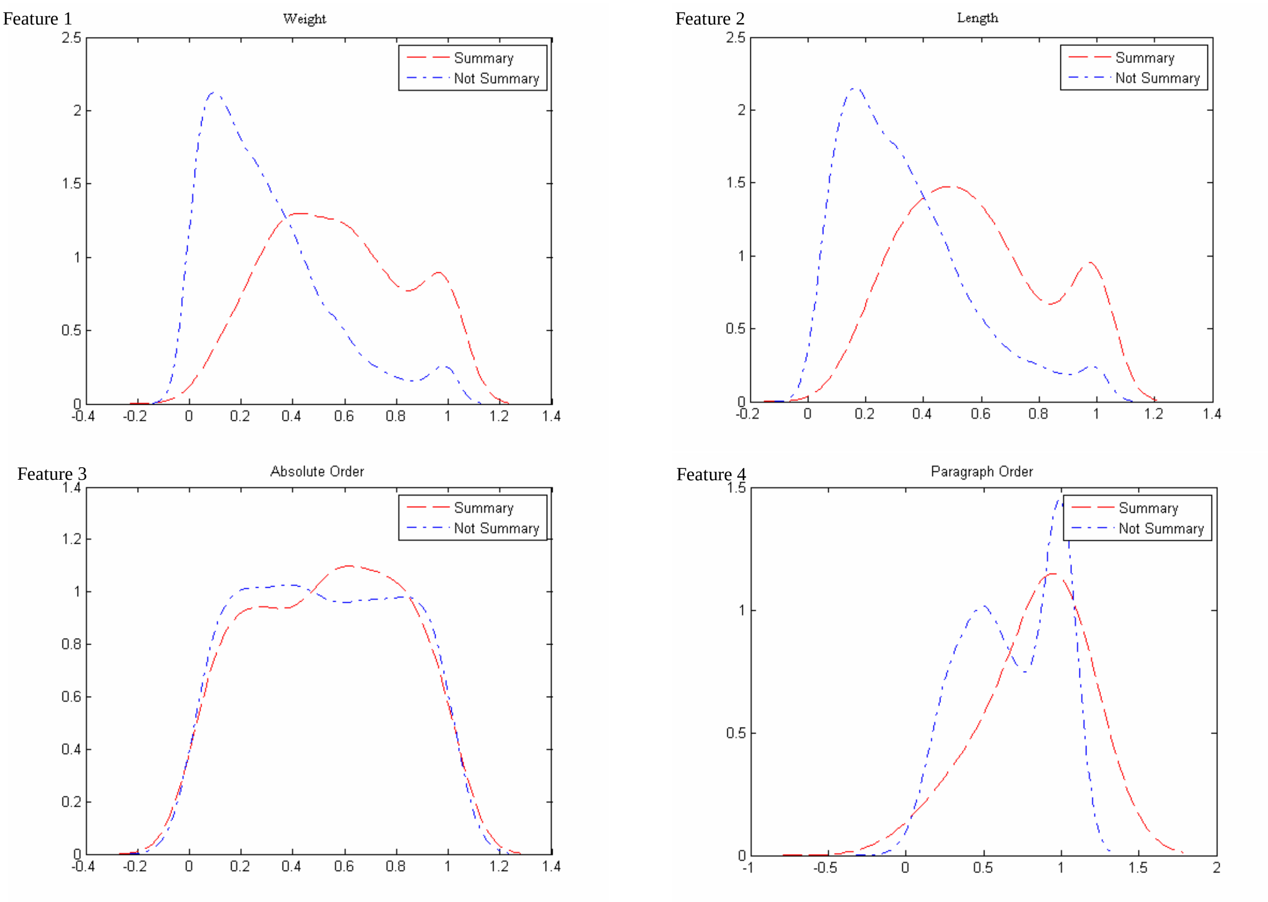
On the other hand, the following features supposed to be discriminative (*strong*):

- Feature 1: Sentence Weight
- Feature 2: Sentence Length
- Feature 6: Sentence Similarity

However, the following features are in between (*Intermediate*):

- Feature 4: Sentence Paragraph Position
- Feature 5: Sentence Paragraph Length
- Feature 7: Number of Infinitives
- Feature 8: Number of Verbs
- Feature 9: Number of Identified

The following *Figure-1* shows probability distributions for each feature independently over the summary and out of summary classes.



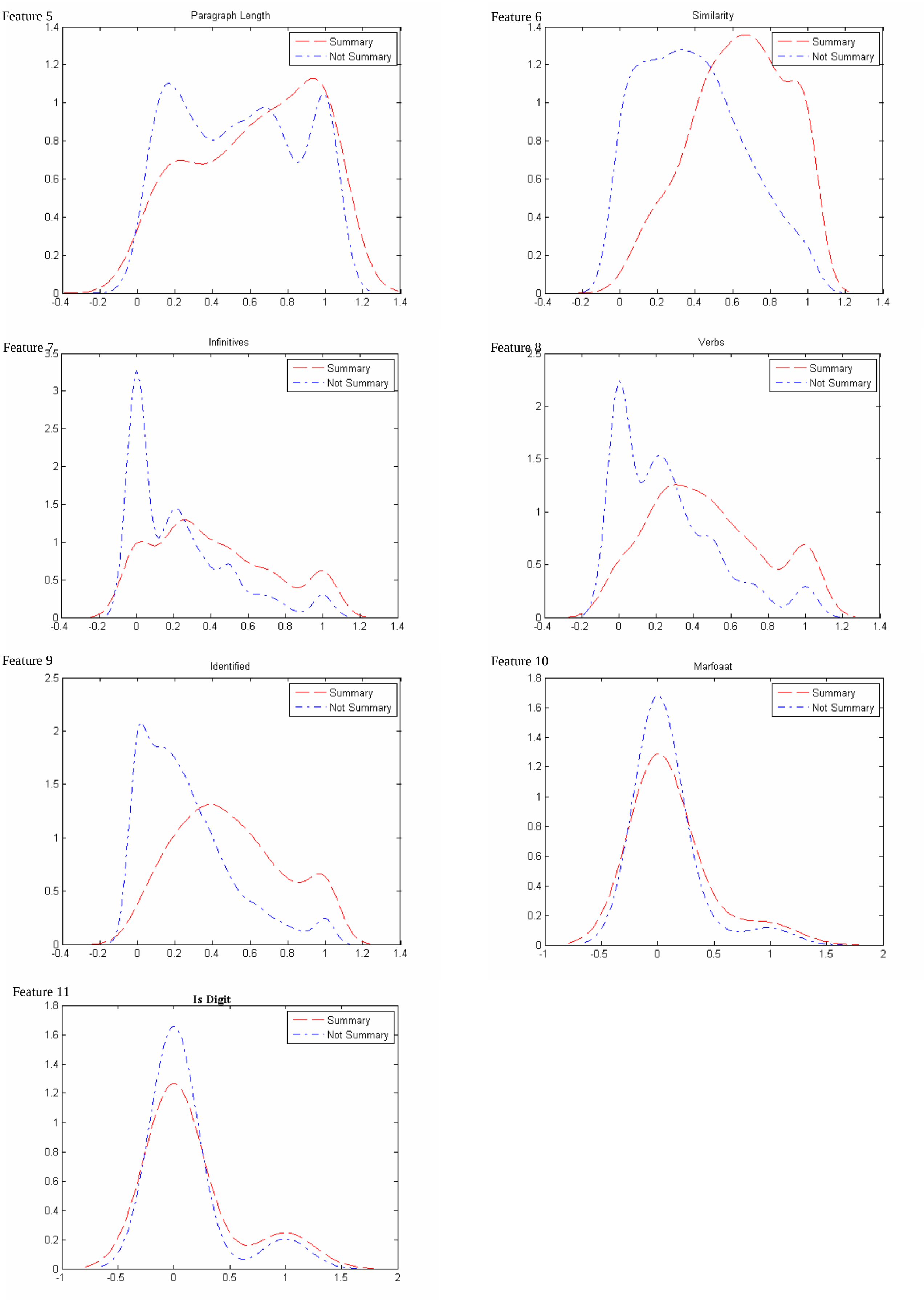


Figure-1 Features Probability Distributions

In the following experiments and guided by probabilities distributions we gradually removed features and observed the system outputs and results using Bayesian classification. The results are presented in the following table:

Features	Recall	Precision	F.Measure	Summary Size
1 to 11 (11 Features)	0.711	0.489	0.579	33.99%
1,2,6,8,9,4,7,5 (8 Features)	0.716	0.490	**0.582**	34.11%
1,2,6,8,9,4,7 (7 Features)	**0.730**	0.487	**0.584**	34.99%
1,2,6,8,9 (5 Features)	0.720	0.476	**0.574**	35.33%
1,2,6 (3 Features)	0.659	0.496	**0.566**	31.01%
1,2 (2 Features)	0.559	0.529	0.544	24.7%
2 (1 Feature)	0.265	0.609	0.370	10.19%
1 (1 Feature)	0.479	**0.706**	**0.515**	20.04%

Table 2: *Feature Reduction effect*

This experiment shows that by removing weak features and using only eight features we have F-Measure of 0.582 which is even better that using all of the features. By removing intermediate features gradually the system results is slightly decreased. Accordingly, by using only 3 strong features we have F-Measure of 0.566. Finally by using only the sentence length feature we have a relativity high precision of 0.706 and F-Measure of 0.515. The best recall was 0.73 (7 features), the best precision was 0.706 (1 feature) and the best F-Measure was 0.584 (7 features).
To improve the seven features experiment above (Best F-Measure), we considered the correlation between features. We found for example there is a high correlation between Sentence Length and Sentence Weight features (0.867) which means that in this specific corpus and domain long sentences tend to contain frequently used words (non-stop words). Also the correlation between Sentence Length and Number of Identified features is (0.808). On the other hand there was a low correlation between Sentence Paragraph Position and Number of Verbs (0.018) and Number of Infinitives (0.023). By eliminating highly correlated features and keeping strong features this will decrease system dimensionality while preserving accuracy. The selected five features are:

- Feature 2: Sentence Length
- Feature 4: Sentence Paragraph Position
- Feature 6: Sentence Similarity
- Feature 7: Number of Infinitives
- Feature 8: Number of Verbs

This reduction saved more than 50% of total processing time compared with the usage of the whole eleven features.

3.2 Classifiers Combination

Table 3 shows the results when using the selected five features by the Bayesian, GP and the combined classifiers results.

System	Recall	Precision	F.measure	Summary Size
Bayesian	0.687	0.533	0.600	30.12%
GP	0.474	0.725	0.573	15.28%
AND	0.464	0.754	0.577	14.40%
OR	0.697	0.525	0.599	31.01%

Table 3. *Five Features Classification Results*

Table 4 summarizes the final results in terms of F-Measure and produced summary size percentage

	11 features		**5 features**	
	F	**Size**	**F**	**Size**
Bayesian	0.579	33.99	**0.600**	30.12
GP	0.574	15.61	0.573	15.28
AND	**0.565**	**15.06**	**0.577**	**14.40**
OR	**0.585**	34.55	**0.599**	31.01

Table 4. *Summarized Total Results*

4. Conclusions and future work

In this paper, an optimized dual classification system for Arabic extractive text summarization has been introduced. Both classification methods have relatively close F-measures, but GP system tends to produce smaller summaries. Bayesian classification method is simple, assumes feature independence and may fall into local optima where GP search is global. Generally, the Bayesian classifier tends to have large recall, on the other hand the GP classifier tends to have large precision. By integrating both classifiers we found that using the *union* for integration increases the recall and the result summary size. However, using the *intersection* for integration increases the precision and decreases the summary size.

When using the eleven features, the experiments show that the best F-measure was 0.585 when using the union between Bayesian and GP classifiers. The best (F-Measure/summary size) ratio came from using the intersection between Bayesian and GP classifiers. When removing features gradually guided by probability distributions the results decreased slightly. Depending on correlation analysis, and to increase system optimization, we used only five features. The best F-measure was 0.600 when using the Bayesian classifier and 0.599 when using union between Bayesian and GP classifiers. The best (F-Measure/summary size) ratio came from using the intersection between Bayesian and GP classifiers. Finally, by selecting the appropriate five features instead of eleven; we get promising and acceptable results for a wide range of applications. The trainability feature of the system makes it possible to be customized and tuned for specific domains and applications. Some applications need high recall others need high precision. The proposed system is tunable to such applications.

For improvements, number of techniques can be applied to enhance the results. Selecting more features like user defined key words, named entities or indicator phrases is expected to increase the system controllability and results.
Adding semantic information from comprehensive lexical resource such as WordNet [20] but for Arabic language will enhance output cohesion. Integrating (LSA) score for each sentence as a feature is expected to enhance the results. Using Linear Discriminate Analysis (LDA) is expected to decrease dimensionality in a more structured way and maintain acceptable results by making a transformation for the current feature space into a new one where the classes are separated in a more discriminative way. Integration between Bayesian and GP classifiers using union and intersection has the advantage of being fast and simple. Using a third classier in a multi-classifier system is expected to enhance the results. Adopting alternative techniques for evaluation will help better understanding the nature of the summarization problem. For example, ROUGE evaluation; Human evaluations for the output summary by judging aspects like coverage, coherence; Another technique is to compare the system summary against number of human generated summaries and see the relation between all of these summaries. We also plan to use and customize the same system for different domains and study the effect of this on the features and overall system performance.

5. Acknowledgments

Special thanks for The Research & Development International Company (RDI®[3]) for its support. We must also mention valuable efforts of Natural Language Processing technology and linguistic research teams.

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

[3] http://www.rdi-eg.com

[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] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in Proc. ACM SIGIR, 2001, pp. 19–25.

[8] J. Goldstain,M. kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in Proc. of ACM SIGIR'99, Aug. 1999.

[9] 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.

[10] Khoja, S. and Garside, R. "Stemming Arabic Text" Computing Department Lancaster University, Lancaster, 1999.

[11] Attia, M. (2000). A Large-Scale Computational Processor of The Arabic Morphology, and Applications, MSc. thesis, Dept. of Computer Engineering, Faculty of Engineering, Cairo University.http://www.RDI-eg.com/RDI/Technologies/paper.htm

[12] Attia, M. (2005), Theory and Implementation of a Large-Scale Arabic Phonetic Transcriptor, and Applications, PhD thesis, Faculty of Engineering, Dept of Electronics and Electrical Communications, Cairo University. http://www.RDI-eg.com/RDI/Technologies/paper.htm

[13] Chikashi Nobata, Satoshi Sekine "CLR/NYU Summarization" DUC-2004.

[14] Ibrahim Sobh, Nevin Darwish, Magda Fayek " A Trainable Arabic Bayesian Extractive Generic Text Summarizer" Proceedings of The Sixth Conference On Language Engineering, ESLEC, 2006. 149-154.http://www.RDI-eg.com/RDI/Technologies/paper.htm

[15] Holland, J. "Adaptation in natural and artificial systems", MIT press, Cambridge, MA, 1975.

[16] Cramer, N.L. "A representation for the adaptive generation of simple sequential programs" Proceedings of an International Conference on Genetic Algorithms and the Applications, 183-187, Carnegie-Mellon University, Pittsburgh, PA. 1985.

[17] Koza, J.R. "Hierarchical genetic algorithms operating on populations of computer programs" Proceedings of the Eleventh International Joint

Conference on Artificial Intelligence IJCAI-89. Volume 1, pages 768-774. Morgan Kaufmann. San Francisco, CA. 1989.

[18] Koza, J.R "Genetic Programming: On the Programming of Computers by Natural Selection", MIT Press, Cambridge, MA. 1992.

[19] Bowman, A. W., and A. Azzalini, Applied Smoothing Techniques for Data Analysis, Oxford University Press, 1997.

[20] Miller, G. "WordNet: A Lexical Database for English." Communications of the Association for Computing Machinery (CACM) 38, 11, 39-41, 1995.

[21] Yihong Gong and Xin Liu "Creating Generic Text Summaries" Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01).