

Evaluation Approaches for an Arabic Extractive Generic Text Summarization System

Ibrahim Sobh^{1,2}, Nevine Darwish¹, Magda Fayek¹

¹ The Department of Computer Engineering, Cairo University, Giza, Egypt.

² The Research and Development International Company, RDI, <http://www.rdi-eg.com>.
sobh@rdi-eg.com, {ndarwish, magdafayek}@eng.cu.edu.eg

Abstract

The advance of technology and extensive use of the web has prompt the need to summarization of text documents. Users tend to extract the most informative or indicative information instead of reading the whole original documents. Naturally, automatic text summarization will save time and effort for the users, and will enable them to make decisions in less time. This paper introduces evaluation methods for 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 annotated corpus. We have introduced methods for evaluating the summary against other human summaries. Moreover, we used human judgement for system output, and finally we tested the system against a commercial Arabic summarization system.

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. (Luhn, 1958) extracted abstracts of scientific articles automatically based on the assumption that frequent words represents the most important concepts of the document. (Edmundson et al. 1961) presented a survey of the existing methods for automatic summarization. Based on cue phrases, title, key words and title (Edmundson, 1969) has implemented document summarization. Basically, these methods form the core of the extraction methods even today.

Uses of Summaries

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

Extractive Summarization

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 (HMM) solution to the decomposition problem was proposed by (Jing, 1999) 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) or sentences (Kupiec et al. 1995). 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 (Gong and Liu 2001), 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 (Carbonell and Goldstein 1998), and Graph Based methods that models the document into graph where sentences are the vertices, and Machine Learning Approaches (Kupiec et al. 1995).

Abstractive Summarization

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 (Jing, H. and McKeown, K.R., 1999). Abstraction requires text analysis, modeling and language generation techniques.

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, precision and F-measure to evaluate summaries.

In this paper, we measured how human summaries may differ, and how our system performed relative to different human summaries. We tested our system using the same measures against a well known commercial summarization system referenced as “S System”. In addition to this, we asked two humans to give each sentence in the system output summary a subjective score to get a measure of summary quality.

System Overview

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. The importance of a sentence within a document can be determined by various heuristics such as position, cue phrases (Edmundson 1969, Kupiec et al. 1995), word/phrase frequency (Luhn 1958, Edmundson 1969, Kupiec et al. 1995), lexical cohesion (Barzilay and Elhadad 1997), discourse structures (Marcu, 1998), and indicator phrases (Hovy and Lin 1999, Kupiec et al. 1995). Naive Bayesian classification method is considered to be simple, easy to implement and does 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 extractive summaries (Kupiec et al. 1995) and key phrase extraction (Witten et al. 1999). Genetic Programming (GP) is used also for classification and could be used for extractive summarization (Turney, 2000). 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 a reduced feature set. The system structure requires annotated training and testing corpus.

Arabic Processing

Arabic as high inflected and derivative language requires stemming for information retrieval and summarization applications. Feature extraction requires complex Arabic language processing: Stop words removal, Stemming and Part Of Speech Tagging (POST). We used the implementation of (Attia, 2005) as a robust method for extracting roots as stems, POST and stop words.

Features

We used only five discriminative features (Sobh, I., Darwish, N., Fayek. M. 2007) for each sentence:

- 1) Sentence length, 2) Sentence position in paragraph, 3) Sentence similarity, 4) Number of infinitives in sentence and 5) Number of verbs in sentences.

The Classifiers

We used two classifiers in parallel. Naive Bayesian classifier and Genetic Programming classifier.

Naive Bayesian Classifier

A Bayesian classifier classifies each sentence to be in summary or out of summary classes based on its feature vector and the training data. For each sentence the probability that will be included in summary can be computed as follows:

$$P(s \in S | V_1, V_2, \dots, V_n) = \frac{P(V_1, V_2, \dots, V_n | s \in S)P(s \in S)}{P(V_1, V_2, \dots, V_n)} \quad (1)$$

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:

$$P(s \in S | V_1, V_2, \dots, V_n) = \frac{\prod_{i=1}^n P(V_i | s \in S)P(s \in S)}{\prod_{i=1}^n P(V_i)} \quad (2)$$

The sentence is classified into summary class if the following condition is fulfilled:

$$\prod_{i=1}^n P(V_i | s \in S)P(s \in S) > \prod_{i=1}^n P(V_i | s \in NS)P(s \in NS) \quad (3)$$

Where NS is the non summary class.

Genetic Programming 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. GP was used successfully in many fields for example, financial market, image processing, optimization, signal processing and pattern recognition. In his book (Holland, 1975), Holland mentioned Artificial intelligence (AI) as one of the main motivators for the creation of genetic algorithms. He did not experiment the direct use of GA to evolve programs. Two researchers, (Cramer, 1985) and (Koza, 1989) 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. (Koza, 1989) provided evidence in the form of

several problems from five different areas. In his book, (Koza, 1992) he sparked the rapid growth of GP.

We choose to use the Discipulus¹ 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 settings for cross-over and mutation rates when running the tool for classification.

The Dual Classification System

There are many classifier combination topologies. 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.

$$Class = Class_{Bayesian} \cup Class_{GeneticProgramming} \quad (4)$$

-Bayesian Classifier Intersection (AND) GP Classifier:

Consider sentence in summary if and only if both classifiers agree.

$$Class = Class_{Bayesian} \cap Class_{GeneticProgramming} \quad (5)$$

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 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 annotated as in summary class; unselected sentences are annotated as out of summary class and features vectors are extracted for all sentences. Total number of sentences is 4899 sentences. (23 sentences per document in average). The human summary size in the training set is 23.3%.

System Evaluation and Results

We used three methods for evaluating the system generated summary:

1. Calculating precision, recall and F-measure.
2. Comparing with other human summaries.
3. Using Human judgment for each sentence in system summary.

Moreover, we compared these results with a well known summarization system referenced as "S System".

Precision and Recall

Classification approach for generating automatic summaries makes it easier for evaluating extractive summaries. Three important measures are commonly used, precision, recall and F-measure for example (Steve et al. 2002) and (Gong and Liu 2001). Precision is a measure of how much of information that the system returned is correct.

-Precision = Number of system correct summary sentences / Total 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:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (6)$$

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 results when using the five features for the Bayesian classification and GP classification independent and integrated.

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 1: Five features summarization evaluation

Comparing independent human summaries

In order to understand how humans may generate different extractive summaries for the same document, we called the main human summarizer the "reference human summarizer". We asked two additional independent human summarizers to extract sentences from the same testing set. Then we computed the summary compression ratio for each one and we computed the common selected sentences between each pair. Table 2 shows the cross-evaluation between summary sizes.

¹ <http://www.aimlearning.com>

² <http://www.ahram.org>

System	Summary Size
Reference Human	23.4%
Human 1	35.8%
Human 2	32.3%

Table 2: Human summaries size comparison

Table 3 shows a comparison between different human summaries intersections (common extracted sentences) percentages. For example, the intersection sentences between "Reference" and "Human 1" is 47.4% relative to reference summary size (this could be the recall of human 1 summary given reference summary, or precision of reference summary given human 1 summary).

Human 1			Human 2			Reference
R	P	F	R	P	F	
0.309	0.474	0.374	0.469	0.649	0.544	
						Human 1
			0.534	0.483	0.507	

Table 3. Human summaries cross-evaluation comparison.

The largest F-measure was 0.544 between Human 2 and the reference summaries. The largest recall 0.534 was between Human 1 and Human 2 summaries. The largest precision was between Human 2 and the reference summaries. This also shows that human summaries may differ in size and the selected extracted sentences. The following figure compares between each pair of summaries. This includes our system: (Bayesian, GP, AND, OR), and human summaries: Reference, Human 1 and Human 2 summaries.

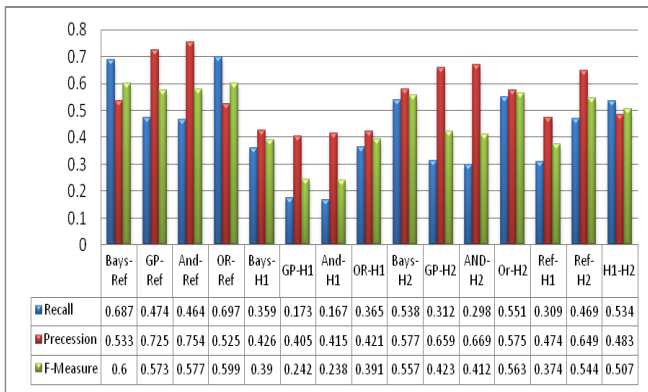


Figure 1: System pairs comparisons

The comparison shows that (our system-Reference) summary pair has the largest F-Measure between all other pairs. Also (our system-Human 2) has average F-Measure of 0.489 which is larger than (our system-Human 1) pair where the average F-Measure of 0.315. On the other hand, the (AND-Human 1) and (GP- Human 1) have the lowest F-Measures of 0.238, 0.242 respectively (It was expected due to the fact that the AND-system summary size is 14.4% and GP-system summary size is 15.28% and hence there is no chance to get high recall for other human summaries).

The (Bayesian-Human 2) and (OR-Human 2) pairs have F-Measures of 0.557, 0.563 respectively which is much better than (Human 1- Human 2), (reference-Human 1) and (reference-Human 1) pairs. These results imply that our system exists in the area of human performance and the difference between the system and the humans is actually comparable to the difference between humans.

Comparing with S System

Figure 2 compares between S System and our system from the reference summarizer point of view.

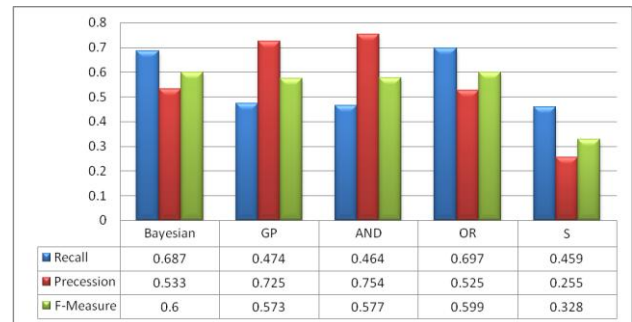


Figure 2: Systems comparisons with reference summary

As expected, our systems results were close to the reference summary (as the system was trained on this reference summary) where the S System did not see the reference summary before. In order to make fare comparison, we compared between S System and our systems from the two new human summaries point of view. Figure 3 shows the results.

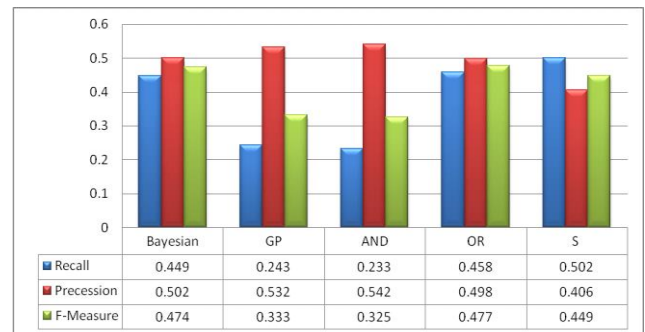


Figure 3: Systems comparisons with two human summaries

This comparison shows that S System had the best recall over all the systems, then the OR system; on the other hand all our systems had better precision than S System. In terms of F-Measure, the Bayesian and the union systems were slightly better than S System. This comparison does not show the summary size. It is usually required to have high F-Measure at relatively small summary size; figure 4 shows the comparison between S System and our systems taking into consideration the F-Measure/summary size ratio.

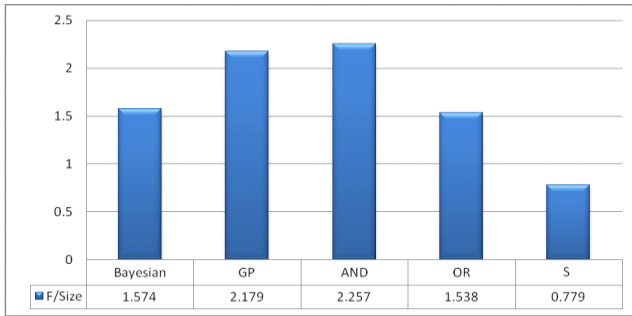


Figure 4: Systems (F/Size) Comparisons with two human summaries

We noted that S System tends to select most of the sentences as summary if the original document is relatively small (8 to 10 sentences). Also in our system we considered a "comma" character as separator between sentences to provide more flexibility for human summarizers when making decisions if the sentence in summary or not. On the other hand we noted that S System did not consider this character as a separator, this makes its results more coherent but produced larger summaries that lowered the F-Measure/summary size ratio.

Human Evaluation

Although we are using automatic techniques for evaluating summaries due to the fact that we have a golden/reference summary, it is still important to evaluate the output summaries using human judgments to have another way of evaluating a summary even that the expensive cost of human judgment. We asked the two human summarizers to evaluate the output of the systems. For each summary, they are asked to assign each sentence given its summary context a label as follows:

-Good: It will be better to add this sentence to be in this summary. This may be because the sentence is informative, important and does not cause ambiguity with surrounding sentences.

-Fair: The sentence could be in or out this summary. This may be because the sentence contains marginal information.

-Bad: It will be worse to put this sentence in this summary. This may be because the sentence contains repeated, incomplete or useless information.

For example, a sentence could be selected as "good" in certain summary and "fair" in another summary. We applied this human judgment for the Intersection system (Bayesian AND Genetic Programming), the Union system (Bayesian OR Genetic Programming) and finally, the S System. The results are showed in figure 5.

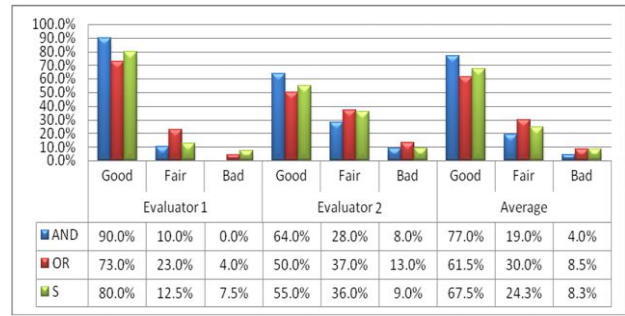


Figure 5: Systems human evaluation comparisons

These results show that even the two evaluators results are different, the best system for both was AND system, then the S System, then the OR system.

Conclusions

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. By integrating both classifiers we found that using the union for integration increases the recall and the result summary size that could be used as informative summary. However, using the intersection for integration increases the precision and decreases the summary size that could be used as indicative summary.

In order to understand the nature of human summaries we asked two additional human to summarize the text. Then we compared each pair in terms of recall, precision and F-Measure. We found that our system performance was in the same area as humans. Moreover, we used the S system and compared it against the additional human summaries. We found that the S system had the best recall; on the other hand all our systems had better precision than the S system. In terms of F-Measure, the Bayesian and the union systems were slightly better than the S system. When taking the size of the summary, our system was much better than the S system. By applying two human subjective judgments for each sentence given its summary context, we found that evaluation tends to prefer the AND system over the S system and OR systems. Our system got average of 69% good sentences.

Finally, our system is optimized, easy to train and customize and able to produce summaries comparable to human generated summaries. We expect the system to be used for a wide range of applications.

Future Work

Applying number of suggested techniques is expected to enhance the system results. Adding semantic information from comprehensive lexical resource such as WordNet (Miller, 1995), but for Arabic language, may enhance output cohesion and help in feature selection. One problem with extracted sentences, they may contain anaphora links to the rest of the text. This has been investigated by (Paice, 1990). Several heuristics have been proposed to solve this problem such as including the sentence just before the extracted one. Anaphora solving seems to be interesting point of research. Adopting alternative techniques for evaluation will help better understanding the nature of the summarization problem. For example; testing the system performance for accomplishing another task such as question answering or document classification. Moreover, we plan to use and customize the same system for different domains and study the effect of this on the recommended features and overall system performance. Using word stem (root + form) instead of root only may enhance the results.

References

- Attia, M. (2005). "Theory and Implementation of a Large-Scale Arabic Phonetic Transcriber, and Applications", PhD thesis, Faculty of Engineering, Dept of Electronics and Electrical Communications, Cairo University. <http://www.RDI-eg.com/RDI/Technologies/paper.htm>
- Barzilay, R., and Elhadad, M., (1997). "Using lexical chains for text summarization", in Proceedings of the ACL Intelligent Scalable Text Summarization Workshop (ISTS), 86-90.
- Carbonell, J., and Goldstein, J., (1998). "The use of MMR, diversity-based reranking for reordering documents and producing summaries", in Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98), 335-336, Melbourne, Australia, August.
- Cramer, N.L. (1985). "A representation for the adaptive generation of simple sequential programs" in proceedings of an International Conference on Genetic Algorithms and the Applications, 183-187, Carnegie-Mellon University, Pittsburgh, PA.
- Edmundson, H.P. (1969). "New Methods in Automatic Extracting". *Journal of the ACM*, 16(2): 264-285.
- Edmundson, H.P. and R.E. Wyllys. (1961). "Automatic Abstracting and Indexing-Survey and Recommendations". *Communications of the ACM*, 4(5): 226-234.
- Evans, D.K., McKeown, K., Klavans, J.L. (2005). "Similarity-based Multilingual Multidocument Summarization", Technical Report CUCS-014-05, Department of Computer Science, Columbia University.
- Gong, Y. and Liu, X. (2001). "Generic text summarization using relevance measure and latent semantic analysis" in proceedings of Special Interest Group on Information retrieval, SIGIR, ACM, 19-25.
- Holland, J. (1975). "Adaptation in natural and artificial systems", MIT press, Cambridge, MA.
- Hovy, E.H., and Chin-Yew Lin. (1999). "Automated text summarization in SUMMARIST" In ACL/EACL summarization workshop, 18-24, Madrid, Spain
- Jing, H. and McKeown, K.R., (1999). "The decomposition of human-written summary sentences" in proceedings of Special Interest Group on Information retrieval, SIGIR, ACM, 129 - 136
- Koza, J.R. (1992). "Genetic Programming: On the Programming of Computers by Natural Selection", MIT Press, Cambridge, MA.
- Koza, J.R. (1989). "Hierarchical genetic algorithms operating on populations of computer programs" in proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI, 768-774. Morgan Kaufmann. San Francisco, CA.
- Kupiec, J., Pederson, J. O., Chen, F. (1995). "A Trainable Document Summarizer" in proceedings of Special Interest Group on Information retrieval, SIGIR, ACM, 68-73.
- Luhn, H. (1958). "The automatic Creation of Literature Abstracts", *IBM Journal of Research and Development* 2(92):159-165.
- Marcu, D., (1998). "Improving Summarization through Rhetorical Parsing Tuning", in proceedings of the COLINGACL workshop on Very Large Corpora. Montreal, Canada.
- Maryland, CS Dept. and Inst. for Advanced Computer Studies, College Park, USA. Conference on Intelligent Text Processing and Computational Linguistics, CILing, 568-581.
- Miller, G. (1995). "WordNet: A Lexical Database for English." *Communications of the Association for Computing Machinery (CACM)* 38(11): 39-41.
- Paice, C., (1990). "Constructing literature abstracts by computer: techniques and prospects", *Information processing and management*, 26: 171-186.
- Sobh, I., Darwish, N., Fayek. M. (2007). "An Optimized Dual Classification System for Arabic Extractive Generic Text Summarization" in proceedings of the Seventh Conference on Language Engineering, ESLEC. <http://www.RDI-eg.com/RDI/Technologies/paper.htm>
- Steve, J., Stephen, L., and Gordon, W., (2002). "Interactive Document Summarization Using Automatically Extracted Key phrases", in proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS-35.
- Turney, P.D. (2000). "Learning Algorithms for Keyphrase Extraction", *Information Retrieval*, 2(4), 303-336 (National Research Council 44105, Canada)
- Witten, I.H., Paynter, G.W., Frank E., Gutwin, C., and Nevill-Manning, C.G. (1999). "KEA: Practical Automatic Keyphrase Extraction" in proceedings of ACM Digital Libraries Conference, 254-255.