Text Summarization Based on Genetic Programming

POOYA KHOSRAVIYAN DEHKORDI & DR. FARSHAD KUMARCI*
Islamic Azad University (Shahrekord Branch)

DR. HAMID KHOSRAVI
International Center for Science & High Technology & Environmental Sciences,
University of Shahid Bahonar Kerman

Abstract:
This work proposes an approach to address the problem of improving content selection in automatic text summarization by using some statistical tools. This approach is a trainable summarizer, which takes into account several features, for each sentence to generate summaries. First, we investigate the effect of each sentence feature on the summarization task. Then we use all features in combination to train genetic programming (GP), vector approach and fuzzy approach in order to construct a text summarizer for each model. Furthermore, we use trained models to test summarization performance. The proposed approach performance is measured at several compression rates on a data corpus composed of 17 English scientific articles.

Categories and Subject Descriptors: H.3.1 [**INFORMATION STORAGE AND RETRIEVAL**]: Content Analysis and Indexing - *Abstracting methods*; J.3 [**Computer Applications**]: LIFE AND MEDICAL SCIENCES - *Biology and genetics*
General Terms: Algorithms, Human Factors, Experimentation
Additional Key Words and Phrases: Automatic Text Summarization, Genetic Programming, Vectorial Model, Fuzzy Model

IJCIR Reference Format:

Pooya Khosraviyan Dehkordi, Hamid Khosravi and Farshad Kumarci. Text Summarization Based on Genetic Programming. International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol.3, No.1, pp 57-64. http://www.ijcir.org/volume3-number1/article7.pdf.

1. INTRODUCTION

Automatic text summarization has been an active research area for many years. Evaluation of summarization is a quite hard problem. Often, a lot of manual labour is required, for instance by having humans read generated summaries and grading the quality of the summaries with regards to different aspects such as information content and text clarity. Manual labour is time consuming and expensive. Summarization is also subjective. The conception of what constitutes a good summary varies a lot between individuals, and of course also depending on the purpose of the summary.

Recently many experiments have been conducted for the text summarization task. Some were about evaluation of summarization using relevance prediction [Hobson et al. 2007], and voted regression model [Hirao et al. 2007]. Others were about single- and multiple-sentence compression using ‘‘parse and trim” approach and a statistical noisy-channel approach [Zajic 2007] and conditional

* Pooya Khosraviyan Dehkordi (PKhosravyan@iaushk.ac.ir), Dr. Farshad Kumarci (FKumarci@iaushk.ac.ir), Islamic Azad University (Shahrekord Branch); Dr. Hamid Khosravi (Hkhosravi@mail.uk.ac.ir) International Center for Science & High Technology & Environmental Sciences, University of Shahid Bahonar Kerman

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol.3, No.1 pp. 57-64, June 2009.

random fields [Nomoto 2007]. Other research includes multi-document summarization [Harabagiu et al. 2007] and summarization for specific domains [Moens 2007].

We employ an evolutionary algorithm, Genetic programming (GP) [Ferreira 2006], as the learning mechanism in our Adaptive Text Summarization (ATS) system to learn sentence ranking functions. Even though our system generates extractive summaries, the sentence ranking function in use differentiates ours from that of [Sekine and Nobata 2001] who specified it to be a linear function of sentence features. We used GP to generate a sentence ranking function from the training data and applied it to the test data, which also differs from [Lin 1999] who used decision tree, [Aone et al. 1999] who used Bayes's rule, and [Neto et al. 2002] who implemented both Naïve Bayes and decision tree.

In this work, sentences of each document are modeled as vectors of features extracted from the text. The summarization task can be seen as a two-class classification problem, where a sentence is labeled as ''correct" if it belongs to the extractive reference summary, or as ''incorrect" otherwise. We may give the ''correct" class a value '1' and the ''incorrect" class a value '0'. In testing mode, each sentence is given a value between '0' and '1' (values between 0 and 1 are continuous). Therefore, we can extract the appropriate number of sentences according to the compression rate. The trainable summarizer is expected to ''learn" the patterns which lead to the summaries, by identifying relevant feature values which are most correlated with the classes ''correct" or ''incorrect". When a new document is given to the system, the ''learned" patterns are used to classify each sentence of that document into either a ''correct" or ''incorrect" sentence by giving it a certain score value between '0' and '1'. A set of highest score sentences are chronologically specified as a document summary based on the compression rate.

2. BACKGROUND

2.1. Text features

We concentrate our presentation in two main points: (1) the set of employed features; and (2) the framework defined for the trainable summarizer, including the employed classifiers.

A large variety of features can be found in the text-summarization literature. In our proposal we employ the following set of features [Ferrier 2001; Luhn 1998]:

(F1) Sentence Length.
(F2) Sentence Position.
(F3) Similarity to Title.
(F4) Similarity to Keywords.
(F5) Occurrence of proper nouns.
(F6) Indicator of main concepts.
(F7) Occurrence of non-essential information.
(F8) Sentence-to-Centroid Cohesion.

2.2. Summarization based on vectorial approach

A frequently employed text model is the vectorial model [Salton and Buckley 1988]. After the preprocessing step each text element – a sentence in the case of text summarization –is considered as a N-dimensional vector. So it is possible to use some metric in this space to measure similarity between text elements. The most employed metric is the cosine measure, defined as: $\cos(\Theta) = (<x.y>) / (|x| \, . \, |y|)$ For vectors *x* and y, where (<,>) indicates the scalar product, and |x| indicates the module of x. Therefore maximum similarity corresponds to $\cos(\Theta)= 1$, whereas $\cos(\Theta)=0$ indicates total discrepancy between the text elements.

2.3. Text summarization based on fuzzy approach

In order to implement text summarization based on fuzzy logic [Khosravi et al. 2008], we used MATLAB since it is possible to simulate fuzzy logic in this software. To do so; first, we consider each characteristic of a text such as sentence length, location in paragraph, similarity to key word and etc, which was mentioned in the previous part, as the input of fuzzy system. Then, we enter all the rules needed for summarization, in the knowledge base of this system (All those rules are formulated by several expends in this field). After ward, a value from zero to one is obtained for each sentence in the output based on sentence characteristics and the available rules in the knowledge base. The obtained value in the output determines the degree of the importance of the sentence in the final summary. To do these steps, we summarize the same text using fuzzy logic.

2.4. Genetic programming

Genetic programming (GP), first introduced by [Ferreira 2006], is an evolutionary algorithm that evolves computer programs and predicts mathematical models from experimental data. The algorithm is similar to Genetic Algorithm (GA), but uses fixed-length character strings (called *chromosomes*) to represent computer programs which are afterwards expressed as expression trees (ETs). GP begins with a random population of candidate solutions in the form of chromosomes. The chromosomes are then mapped into ETs, evaluated based on a fitness function and selected by fitness to reproduce with modification via genetic operations. The new generation of solutions goes through the same process until the stop condition is satisfied. The fittest individual serves as the final solution. GP has been used to solve symbolic regression, sequence induction, and classification problems efficiently [Ferreira 2006]. We utilized GP to find the explicit form of sentence ranking functions for the automatic text summarization.

3. THE PROPOSED AUTOMATIC SUMMARIZATION MODEL

Figure 1 shows the proposed automatic summarization model. We have two modes of operations:

- Training mode where features are extracted from 16 manually summarized English documents and used to train Genetic programming, Fuzzy and Vector models.
- Testing mode, in which features are calculated for sentences from one English document. (These documents are different from those that were used for training.) The sentences are ranked according to the sets of feature weights calculated during the training stage. Summaries consist of the highest-ranking sentences.

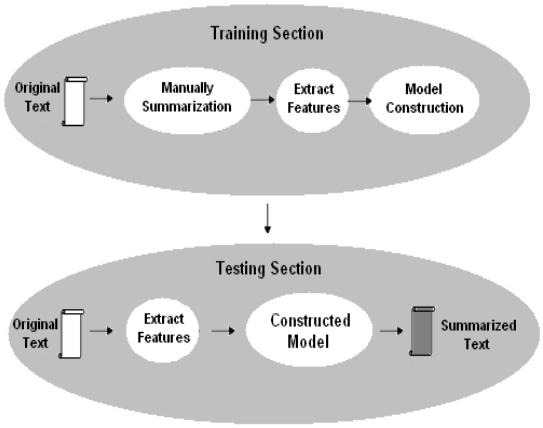


Fig.1: The proposed automatic summarization model

3.1. Genetic programming model

The basic purpose of genetic programming (GP) is optimization. Since optimization problems arise frequently, this makes GP quite useful for a great variety of tasks. As in all optimization problems, we are faced with the problem of maximizing/minimizing an objective function f(x) over a given space X of arbitrary dimension [Yeh 2005]. Therefore, GP can be used to specify the weight of each text feature.

The general view of Genetic Programming like below:

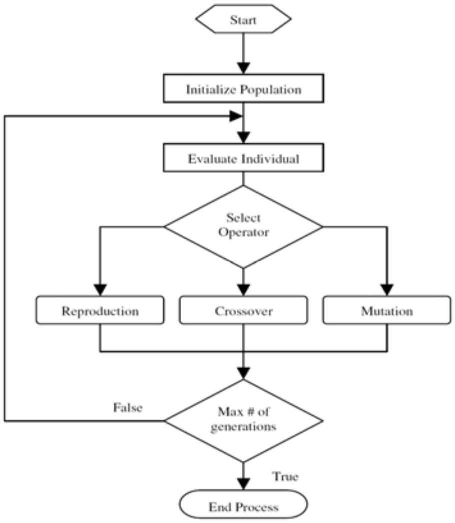


Fig.2: The general view of Genetic Programming

For a sentence s, a weighted score function, is exploited to integrate all the eight feature scores mentioned in Section 2, where w_i indicates the weight of f_i.
The genetic programming (GP) is exploited to obtain an appropriate set of feature weights using the 17 manually summarized English documents. A chromosome is represented as the combination of all feature weights in the form of w_i.

Thousand genomes for each generation were produced. Evaluate fitness of each genome (we define fitness as the average precision obtained with the genome when the summarization process is applied on the training corpus), and retain the fittest 8 genomes to mate for new ones in the next generation. In this experiment, thousand generations are evaluated to obtain steady combinations of feature weights. A suitable combination of feature weights is found by applying GP. All document sentences are ranked in a descending order according to their scores. A set of highest score sentences are chronologically specified as a document summary based on the compression rate.

4. EXPERIMENTAL RESULTS

4.1. The English data

Seventeen English articles in the domain of science were collected from the Reading Book. Seventeen English articles were manually summarized using compression rate of 30%. These manually summarized articles were used to train the previously mentioned three models. The other one English articles were used for testing. The average number of sentences per English articles is 85.8, respectively.

4.2. Genetic programming Configuration

We are going to exploit the GP approach of [Ferrier 2001], for summarization and use it as a baseline approach. For a sentence s, a weighted score function, is exploited to integrate the eight feature scores mentioned in previous section, where w_i indicates the weight of f_i. We use the approach for testing; a set of 17 English documents was used. We apply f_i after using the defined weights from GP execution. All document sentences are ranked in a descending order according to their scores. A set of highest score sentences are chronologically specified as a document summary based on the compression rate.
Related parameters for the training and testing of the GP model like Data, Program Structure, general setting, genetic operators and numerical constants are given in Table I and II.

Table I: GP Data

Independent Variables:	8
Training Samples:	1016
Testing Samples:	105

Table II: GP General Settings

Chromosomes :	30
Genes:	4
Linking Function:	Addition

4.3. The results of genetic programming model

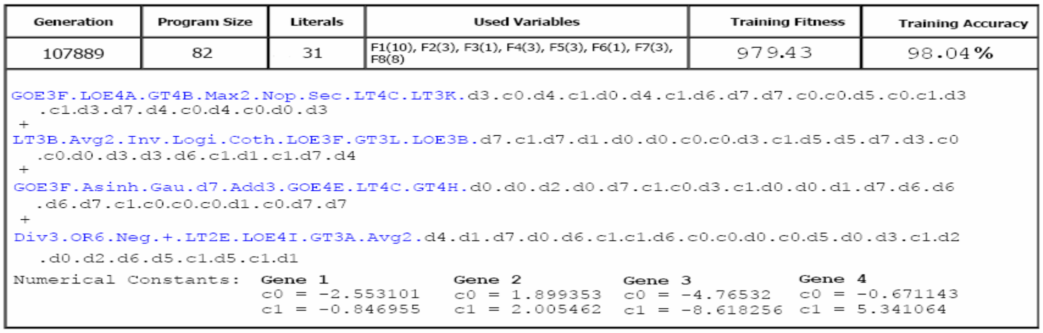
We have exploited the GP approach [Ferreira 2006], for summarization as described above. Therefore, we have exploited the eight features for summarization. The system calculates the feature weights using genetic programming.

All document sentences are ranked in a descending order according to their scores. A set of highest score sentences are chronologically specified as a document summary based on the compression rate. To do GP concepts we using GP Classification model [Ferreira 2006].

4.3.1. GP model explicit formulation

By using GP model and analyzing data we got result given in Table III:

Table III: Specify data was produced by GP concepts for automatic text summarization



Generation	Program Size	Literals	Used Variables	Training Fitness	Training Accuracy
107889	82	31	F1(10), F2(3), F3(1), F4(3), F5(3), F6(1), F7(3), F8(8)	979.43	98.04%

```
GOE3F.LOE4A.GT4B.Max2.Nop.Sec.LT4C.LT3K.d3.c0.d4.c1.d0.d4.c1.d6.d7.d7.c0.c0.d5.c0.c1.d3
   .c1.d3.d7.d4.c0.d4.c0.d0.d3
 +
LT3B.Avg2.Inv.Logi.Coth.LOE3F.GT3L.LOE3B.d7.c1.d7.d1.d0.d0.c0.c0.d3.c1.d5.d5.d7.d3.c0
   .c0.d0.d3.d3.d6.c1.d1.c1.d7.d4
 +
GOE3F.Asinh.Gau.d7.Add3.GOE4E.LT4C.GT4H.d0.d0.d2.d0.d7.c1.c0.d3.c1.d0.d0.d1.d7.d6.d6
   .d6.d7.c1.c0.c0.c0.d1.c0.d7.d7
 +
Div3.OR6.Neg.+.LT2E.LOE4I.GT3A.Avg2.d4.d1.d7.d0.d6.c1.c1.d6.c0.c0.d0.c0.d5.d0.d3.c1.d2
   .d0.d2.d6.d5.c1.d5.c1.d1
Numerical Constants:  Gene 1          Gene 2          Gene 3          Gene 4
                      c0 = -2.553101  c0 = 1.899353   c0 = -4.76532   c0 = -0.671143
                      c1 = -0.846955  c1 = 2.005462   c1 = -8.618256  c1 = 5.341064
```

By using Table III, we can produce Expression Trees (ETs) like below:

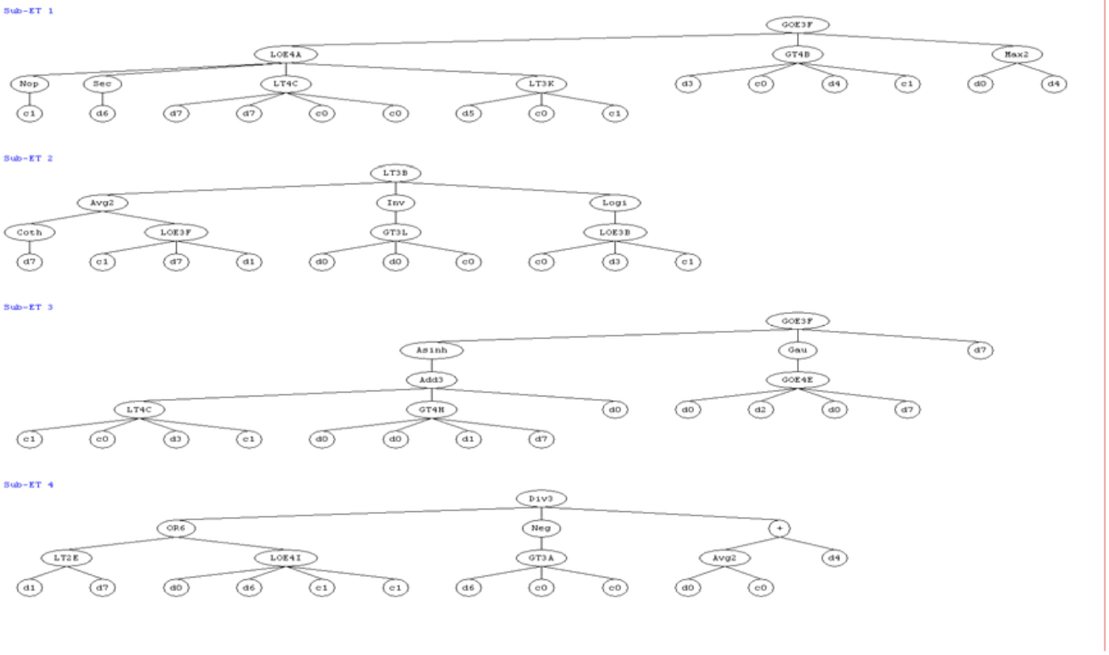


Fig.3: Expression tree of genetic programming model

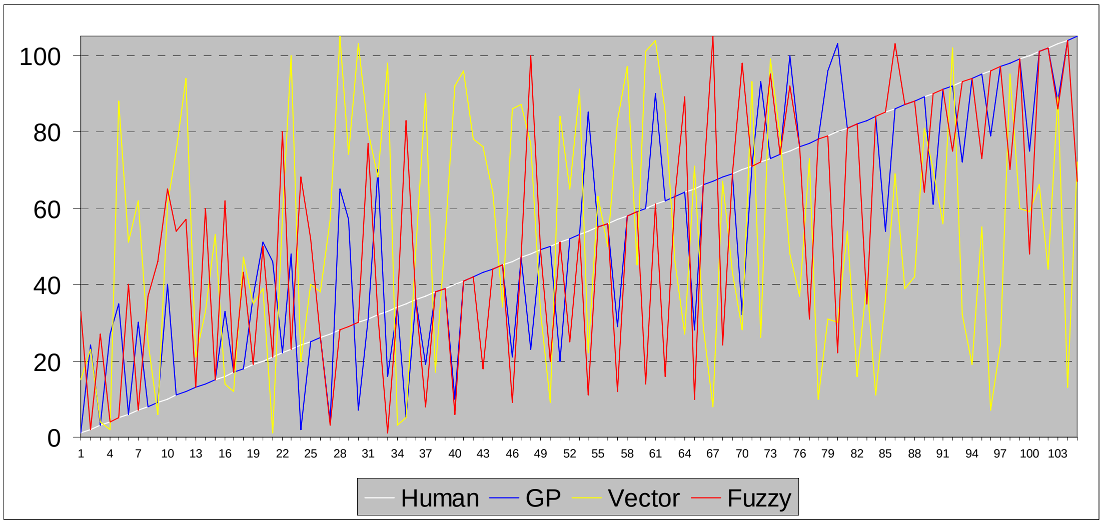


Fig.4: Compare sentence priority of GP, Fuzzy and Vector model with Human priority

Table IV: All models performance evaluation based on Precision

Compression Rate (CR)	10%	20%	30%
	Precision (P)	Precision (P)	Precision (P)
Vector Model	18.18%	19.04%	21.88%
Fuzzy Model	36.36%	42.86%	46.88%
GP Model	54.54%	57.14%	59.38%

4.3.2. Evaluation GP model

4.3.3. We used 16 English text documents for training and one for testing GP model and the results are given in table V and VI:

Table V: Statistics - Training

Best Fitness:	979.43
Max. Fitness:	1000
Accuracy:	98.04%

Table VI: Statistics – Testing

Best Fitness:	625.96
Max. Fitness:	1000
Accuracy:	63.81%

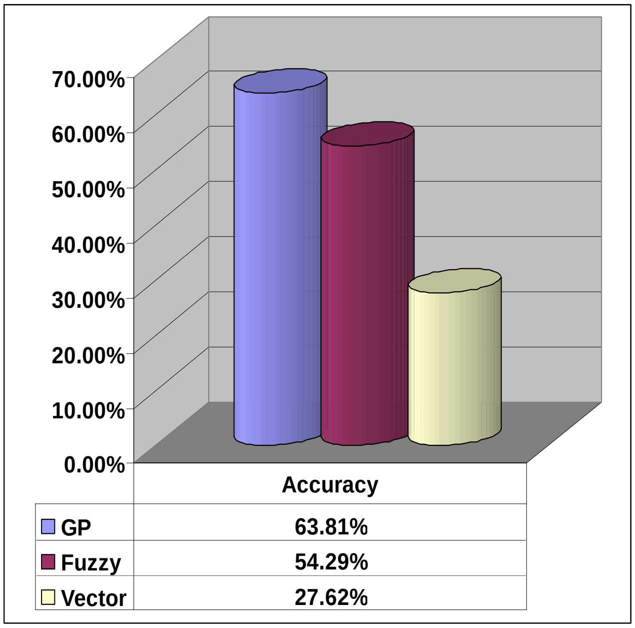


Fig.5: The accuracy for all models

4.3.4. Discussion

It is clear from Table IV and Fig.4 that this approach can be extended to the genre of newswire text. Fig.5 show the total system performance in terms of precision for in case of all models for English articles, respectively. It is clear from the figure that GP approach gives the best results since GP has a good capability to model arbitrary densities. The Fuzzy model has better precision than the Vector model.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the use of genetic programming (GP), vector approach and fuzzy approach for automatic text summarization task. We have applied our new approaches on a sample of 17 English scientific articles. Our approach results outperform the baseline approach results. Our approaches have been used the feature extraction criteria which gives researchers opportunity to use many varieties of these features based on the text type.

In the future work, we will extend this approach to multi-document summarization by addressing some anti-redundancy methods which are needed, since the degree of redundancy is significantly higher in a group of topically related articles than in an individual article as each article tends to describe the main point as well as necessary shared background.

6. REFERENCES

AONE, C., GORLINSKY, J., LARSEN, B., AND OKUROWSKI, M. E. ,"A Trainable Summarizer with Knowledge Acquried from Robust NLP Techniques", *Advances in Automatic Text Summarization*, The MIT Press, Cambridge, Massachusetts, 1999, pages 71-80.

FERREIRA, C., *Genetic programming: Mathematical Modeling by an Artificial Intelligence*, 2nd Edition, Springer-Verlag, Germany, 2006.

FERRIER, L., *A Maximum Entropy Approach to Text Summarization*, School of Artificial Intelligence, Division of Informatics , University of Edinburgh, 2001.

HARABAGIU, S., HICKL, A., LACATUSU, F., "Satisfying information needs with multi-document summaries", *Information Processing & Management*, 43 (6), 2007, 1619–1642.

HIRAO, T., OKUMURA, M., YASUDA, N., ISOZAKI, H., "Supervised automatic evaluation for summarization with voted regression model", *Information Processing & Management*, 43 (6), 2007, 1521–1535.

HOBSON, S., DORR, B., MONZ, C., SCHWARTZ, R., "Task-based evaluation of text summarization using relevance prediction", *Information Processing & Management*, 43 (6), 2007, 1482–1499.

KHOSRAVI, H., ESLAMI, E., KYOOMARSI, K., AND DEHKORDY, P., K., "Optimizing Text Summarization Based on Fuzzy Logic", In: Book Series Studies in Computational Intelligence Publisher Springer, Berlin / Heidelberg, Book *Computer and Information Science*, 2008, Pages 121-130.

LIN, C., "Training a Selection Function for Extraction", *In the 8th International Conference on Information and Knowledge Management (CIKM 99)*, Kansa City, Missouri, 1999, 112-129.

LUHN, H.P., "The automatic creation of literature abstracts", *IBM Journal of Research and Development*, 2 (2), 1998, 159–165.

MOENS, M., "Summarizing court decisions", *Information Processing & Management*, 43 (6), 2007, 1748–1764.

NETO, J. L., FREITAS, A. A., AND KAESTNER, C. A. A., "Automatic Text Summarization using a Machine Learning Approach", *In Proc. 16th Brazilian Symp. on Artificial Intelligence (SBIA-2002).* Lecture Notes in Artificial Intelligence 2507, Springer-Verlag, 2002. pp205-215.

NOMOTO, T., "Discriminative sentence compression with conditional random fields", *Information Processing & Management*, 43 (6), 2007, 1571–1587.

SALTON, G., BUCKLEY, C., "Term-weighting approaches in automatic text retrieval", *Information Processing and Management 24*, Reprinted in: Sparck-Jones, 1988, 513-523.

SEKINE, S. AND NOBATA, C. "Sentence Extraction with Information Extraction technique", *In Proc. Of ACM SIGIR'01 Workshop on Text Summarization*. New Orleans, 2001,1115-1129.

YEH, S.J., KE, T.H., YANG, M.W., "Text summarization using a trainable summarizer and latent semantic analysis", *Information Processing & Management*, 41 (1), 2005, 75–95.

ZAJIC, D., DORR, B., LIN, J., SCHWARTZ, R., "Multi-candidate reduction: sentence compression as a tool for document summarization tasks", *Information Processing & Management*, 43 (6), 2007, 1549–1570.