Using Gene Expression Programming to Construct Sentence Ranking Functions for Text Summarization

Zhuli Xie, Xin Li, Barbara Di Eugenio, Pete C. Nelson
Artificial Intelligence Laboratory, Department of Computer Science,
University of Illinois at Chicago,
Chicago, IL 60607, U.S.A.
zxie@cs.uic.edu, xli1@cs.uic.edu,
bdieugen@cs.uic.edu, nelson@cs.uic.edu

Weimin Xiao, Thomas M. Tirpak
Motorola Advanced Technology Center,
Motorola Labs
Schaumburg, IL 60196, U.S.A.
awx003@motorola.com,
T.Tirpak@motorola.com

0 Abstract

In this paper, we consider the automatic text summarization as a challenging task of machine learning. We proposed a novel summarization system architecture which employs Gene Expression Programming technique as its learning mechanism. The preliminary experimental results have shown that our prototype system outperforms the baseline systems.

1 Introduction

Automatic text summarization has been studied for decades (Edmundson, 1969) and is still a very active area (Salton et al, 1994; Kupiec et al, 1995; Brandow et al, 1995; Lin, 1999; Aone et al, 1999; Sekine and Nobata, 2001; Mani, 2001; McKeown, 2001; and Radev, 2003). Only a few have tried using machine learning to accomplish this difficult task (Lin, 1999), (Aone, et al, 1999), and (Neto, et al, 2002). Most research falls into combining statistical methods with the linguistic characteristic analysis. We regard the text summarization as a problem of empowering a machine to learn from human-summarized text documents. Though the natural language processing techniques are still used in our work, we employed an evolutionary algorithm, Gene Expression Programming (GEP) (Ferreira, 2001), as the learning mechanism in our Adaptive Text Summarization (ATS) system to learn sentence ranking functions.

This paper outlines our approach, details the system architecture, and discusses preliminary experimental results. Conclusion and future works are outlined at the end.

2 Background

2.1 Gene Expression Programming

Gene Expression Programming, first introduced by (Ferreira, 2001), is an evolutionary algorithm that evolves computer programs and predicts mathematical models from experimental data. When using GEP to solve a problem, usually five components, i.e. the function set, terminal set, fitness function, control parameters and stop condition, must be first determined. GEP uses fixed-length of character strings to represent computer programs, which are afterwards expressed as expression trees when evaluating their fitness. The algorithm begins with the random generation of the initial population, which is composed of candidate solutions in form of chromosomes for the problem. Then the chromosomes are expressed and the fitness for each individual is evaluated. The individuals are then selected according to fitness to reproduce with modification. This process loops for every new generation until the preset number of generations is reached or a perfect solution is found. Upon the invention of GEP, it has been used to solve symbolic regression, sequence induction, and classification problems efficiently (Ferreira, 2002), (Zhou, 2003). We adopted GEP for the purpose of finding explicit form of sentence ranking functions for the automatic text summarization task.

2.2 Sentence Features

In our current Adaptive Text Summarization (ATS) system, every sentence ***s*** is represented by five features, namely location of the paragraph that a sentence belongs to in the document, location of the sentence in this paragraph, length of the sentence, whether or not being a header sentence, and content-word frequencies in this sentence.

- **Location of the Paragraph**

The location of the paragraph that includes ***s*** is normalized to [0, 1] by using the MinMax normalization as follows:

$$Y' = \left(\frac{Y - Min1}{Max1 - Min1}\right) \times (Max2 - Min2) + Min2 \quad (1)$$

where $Max2 = 1$, $Min2 = 0$, $Min1=0$, and $Max1 =$ total number of paragraphs in a document.

- **Location of the Sentence**

We used the same MinMax normalization as (1) for the location of the sentence, but here *Max*1 = total number of sentences in the paragraph.

- **Length of the Sentence**

The length of the sentence is the number of words it contains, and is normalized by using Sigmoidal Normalization as given in (2):

$$\alpha = \frac{L(s) - \mu(L(s))}{std(L(s))}, L(s)' = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}, \quad (2)$$

- **Header Sentence**

If ***s*** is a header, the feature of header sentence is set to 1, 0 otherwise.

- **Content-word Frequencies**

For the content-word frequencies, we define the score for a content word w_i as *log[Freq(w_i)]*, where *Freq(w_i)* is the frequency of w_i in that document, and the content-word score for a sentence ***s*** is the summation of the scores of all individual content-words in this sentense:

$$CW(s) = \sum_{i=1}^{k} \log[Freq(w_i)], w_i \in s. \quad (3)$$

Then the sentence scores are normalized by using Sigmoidal Normalization:

$$\alpha = \frac{CW(s) - \mu(CW(s))}{std(CW(s))}, CW(s)' = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}}, \quad (4)$$

where $\mu(CW(s))$ is the mean of all the sentence scores, and *std(CW(s))* is the standard deviation.

2.3 Sentence ranking function

We assume that there are certain rules, either explicit or implicit, which a human summarizer will follow to perform the task of summarization. Correspondingly, our goal is to induce these rules using a collection of documents and their corresponding summaries produced by the summarizers. We process the text summarization learning task in two stages: training and testing. In the training stage, by given a set of training documents with their summarzations, text features are preprocessed using statistical methods and natural language processing methods, then each sentence in a document is scored based on a sentence ranking function constructed by GEP. Fitness value of the summarization task is the similarity between machine summarizaed text and the summarzation text of traning document. the top ***n*** ranked sentences[1] will be returned as the summary of that document and presented in their nature order. In the testing stage, the document set other than the training set is provided to test the similarity between the machine summarized text and human or other system summarized text.

Specifically, the training stage has three components to be defined: a training document set, sentence features of interest, and a learning method. We have defined the learning method in 2.1 and sentence features in 2.2. The training set for our system is a collection of text documents with corresponding summaries. Even though our system also generates extractive summaries, the sentence ranking function in use differentiates ours from that of (Edmundson, 1969), (Sekine and Nobata, 1999), and (Goldstein, et al, 1999) who specified it to be a linear function of sentence features. We used GEP 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) and (Kupiec, et al, 1995) who used Bayes's rule, and (Neto, et al, 2002) who implemented both Naïve Bayes and decision tree C4.5.

3 System Architecture

Our Adaptive Text Summarization (ATS) system follows the traditional way as in (Edmundson, 1969), (Lin, 1999), (Kupiec, et al, 1995), (Brandow, 1995), and (Zechner, 1996) to extract the highest ranked sentences in a document to compose a summary. However, we embedded a machine learning mechanism in our system, which does not appear in the traditional summarization systems. The system architecture is shown in Figure 1, where a GEP module is highlighted. In the training stage, each of ***r*** training documents is passed to the GEP module after being preprocessed

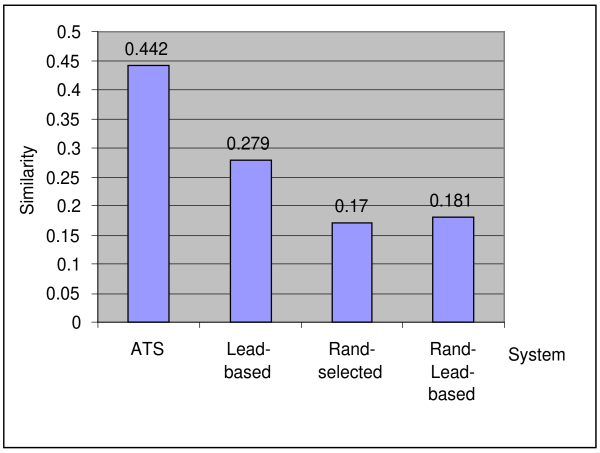
shapeType75fFlipH0fFlipV0fLockRotation1fLockPosition1fLayoutInCell1fPseudoInline1fLayoutInCell1fLockPosition1fLockRotation1b471c923e9be1e770b7a0d49c01bb0a101000900000378000000002001c0000000000040000000301080005000000 0b020000000000500000000c022706430704000000 2e0118001c000000fb021000070000000000bc0200 0000000102022253797374656d00064307000004e0 7dbcd9c5c110004ee8339100536010c0200000400 00002d0100001c000000fb029cff00000000000090 01000000000440001254696d6573204e657720526 f6d616e000000000000000000000000000000000 040000002d010100040000002010100050000000 902000000020d000000320a5900000000100040000 000000400727062 0462d00040000002d010000030 000000000

1 The number of sentences extracted by the GEP module can be a variable, which is decided by the required number of words in a summary. Or it can be a specified percentage of the total number of sentences in the document.

baseline methods, namely lead-based, randomly-selected, and random-lead-based, to generate summaries for performance comparison, which were also adopted by (Brandow, et al, 1995), (Zechner, 1996), and (Radev, 2003). The baseline methods are detailed as follows:

- The lead-based method selects the first sentences from the first five paragraphs as the summary of each of the testing documents.
- The randomly-selected method chooses five sentences from a document at random to compose a summary.
- The random-lead-based method chooses five sentences among the first sentences from all paragraphs in the document at random.

We performed the random selection 1,000 times, and calculated the average similarity of the testing documents for each of the random-based methods. The experimental results are plotted in Figure 2, which have demonstrated that our system outperforms all three baseline methods. However, the value of the average similarity gained by our system is not high enough. The reason most likely lies in the fact that the styles of the objective summaries written by humans vary a lot or even conflict with each other. In other words, they do not possess many common features that are a must for high value of similarity between two texts.



5 Conclusions and Future Work

In this paper, we have presented a prototype summarization system which employs GEP as its learning mechanism for sentence ranking function. In the preliminary experiments for performance testing, our system outperforms the baseline methods by 58% - 160% when generating summaries for 10 documents. Nevertheless, using content-words and the cosine function to measure the similarity is not an ideal evaluation metric, neither is it an ideal fitness measure in the GEP learning mechanism. Our future research will further study what kinds of metric can be obtained from raw texts without involvement of human subjects and how to make them suitable for the GEP fitness measure. Moreover, we plan to collect documents by cluster to train our system and every cluster would only contain articles summarized in a similar style. We will also explore the role of other useful sentence features, such as sentence cohesion, semantic meaning, and rhetorical relations, and include them into the search for an ideal uniform sentence ranking function.

6 Acknowledgements

Our thanks go to Motorola Advanced Technology Center for their support of this research project.

0 References

H. Edmundson. 1969. New methods in automatic abstracting. *Journal of ACM*, 16(2):264-285.

Satoshi Sekine and Chikashi Nobata. 2001. Sentence Extraction with Information Extraction technique. In *Proceedings of ACM SIGIR'01 Workshop on Text Summarization*. New Orleans.

Lin, Chin-Yew. 1999. Training a Selection Function for Extraction. In *the 8th International Conference on Information and Knowledge Management* (CIKM 99), Kansa City, Missouri.

Chinatsu Aone, James Gorlinsky, Bjornar Larsen, and Mary Ellen Okurowski. 1999. A Trainable Summarizer with Knowledge Acquried from Robust NLP Techniques, *Advances in Automatic Text Summarization*, pages 71-80. The MIT Press, Cambridge, Massachusetts.

Inderjeet Mani. 2001. *Automatic Summarization*, John Benjamins Publishing Company, Amsterdam/Philadelphia.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th ACM-SIGIR Conference*, pages 68-73. Seattle, Washington.

Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675-685.

Klaus Zechner. 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *Proceedings of COLING-96*, pages 986-989. Copenhagen, Denmark.

Paul McLellan, et al. 2001. Evaluating summarisation technologies: A task-oriented approach. In *Proceedings of the 1st International Workshop on New Developments in Digital Libraries (NDDL-2001), International Conference on Enterprise Information Systems*

(ICEIS 2001), pages 99-112. Setubal, Portugal.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics, in *Proceedings of SIGIR '99*, pages 121-128. Berkeley, California.

Kathleen R. McKeown, et al. 2001. Columbia Multi-Document Summarization: Approach and Evaluation, in *Proceedings of the Document Understanding Conference (DUC01).* Edmonton, Canada.

Dragomir R. Radev, et al. 2003. Evaluation challenges in large-scale document summarization, in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 375-382. Sapporo, Japan.

Gerard Salton, James Allan, Chris Buckley, and Amit Singhal. 1994. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. *Science*, 264(3):1421-1426.

Cândida Ferreira. 2001. Gene Expression Programming: A New Adaptive Algorithm for solving problems. *Complex Systems*, 13(2):87-129.

Chi Zhou, Weimin Xiao, Thomas M. Tirpak, and Peter C. Nelson. 2003. Evolving Classification Rules with Gene Expression Programming. *IEEE Transactions on Evolutionary Computation*, 7(6):519 – 531.

Joel Larocca Neto, Alex A. Freitas, and Celso A. A. Kaestner. 2002. Automatic Text Summarization using a Machine Learning Approach. In *Proc. 16th Brazilian Symp. on Artificial Intelligence (SBIA-2002). Lecture Notes in Artificial Intelligence 2507*, pp205-215. Springer-Verlag.

Cândida Ferreira. 2002. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence.* Angra do Heroismo, Portugal.