Fuzzy Evolutionary Self-Rule Generation and Text Summarization

Pradeepika Verma
Department of CSE
IIT (ISM) Dhanbad, India
pradeepikav.verma093@gmail.com

Hari Om
Department of CSE
IIT (ISM) Dhanbad, India
hariom4india@gmail.com

Abstract

Over the past decades, soft computing and statistical techniques has acquired serious attention for solving machine learning problems. A system consisting of human like capabilities can be developed using fuzzy logic. Moreover, evolutionary techniques are often fit for approximating solutions. With these potentials, we propose a supervised text summarization approach based on fuzzy rules and human-engineered features. A data-driven fuzzy rules generation system is modeled as a discrete optimization problem and then used to classify the sentences of document. According to these classifications, the relevant sentences are extracted for summary generation. The experimental results on DUC2006 dataset show the effectiveness of the proposed model.

1 Introduction

The automatic text summarization (ATS) systems are welcomed since the necessity to access large amount of textual data has grown. The main objective of these systems is to fetch significant information from the document while maintaining the user requirement (Mani, 2001) and challenge is to produce high quality summary (Binwahlan et al., 2010). According to Ferreira et al. (2013), an ATS system generates concise form of single or multiple documents. Although, a lot of work has previously been done in the field of extractive summarization, the challenges are still there.

In this paper, the proposed model focuses on extractive summary generation based on fuzzy logic and text features. Most of the existing extractive methods are based on finding the relevant sentences using text features such as sentence position and length (Erkan and Radev, 2004), existence of title words, frequent words, and proper nouns in the sentence (Nenkova et al., 2006; Edmundson, 1969). The scores of these features are typically used to assign the score to sentences for making decision on the relevancy of sentences. However, decision with these scores sometimes become fuzzy and low (Zha, 2002; Lin and Hovy, 2003; Murad and Martin, 2007). For example, suppose there are two sentences s_1 and s_2 evaluated on the basis of two features f_1 and f_2 with their weights 0.9 and 0.2, respectively. The feature's scores of s_1 are 0.1 and 0.9 and s_2 are 0.3 and 0.0. According to these scores, both s_1 and s_2 get 0.27 score and consists of equal priority for selection. But if we observe, the feature f_1, whose score is negligible in s_1, has much higher weight in comparison to f_2. Therefore, it would be appreciable if s_2 gets higher priority than s_1. This condition can be better handled with fuzzy logic which motivates us to explore it for summarization.

Few of research works till date have been devoted to fuzzy based summarization schemes. Binwahlan et al. (2010) proposed a fuzzy swarm based summarization method where every sentence is computed on the basis of their feature scores adjusted by particle swarm optimization generated weights. Their scores are then applied to fuzzy logic for better classification in important and unimportant sentences. Abbasi-ghalehtaki et al. (2016) also applied fuzzy logic in the same manner for summarization task where scores are adjusted by hybrid genetic and particle swarm optimization algorithms. In these schemes, much effort and time has been devoted to create fuzzy rules through human experts. Moreover, the processing with these huge set of rules is also time consuming. It motivates us to generate a data-driven optimal set of fuzzy rules. According

Proc. of ICON-2018, Patiala, India. December 2018 ©*2018 NLPAI*, pages 110–114

to Binwahlan et al. (2009), fuzzy logic and swarm intelligence could perform better in the field of text summarization. Therefore, the optimization power of particle swarm optimization (PSO) algorithm in discrete form has been utilized to generate fuzzy rules in this paper.

2 Problem formulation

In this section, we formally introduce our model as follows. Suppose $D = \{s_1, s_2, \ldots, s_n\}$ is a document which consists of n number of sentences where s_m denotes m^{th} sentence of the document. At first, sentence segmentation, and stop word removal steps are carried out. As a result, each sentence $s_m = \{w_1, w_2, \ldots, w_{|s_m|}\}$ is converted into a set of keywords. Thereafter, we score the sentences according to seven text features as given in Table 1.

Text feature	Description	Formulation
f_1	Title words (tw)	$\frac{\sum_{tw \in s_m} (Count(gram_N)}{\lvert tw \rvert . \lvert s_m \rvert}$
f_2	Proper noun (pn)	$\frac{\sum_{pn \in s_m} Count(gram_N)}{\lvert pn \rvert . \lvert s_m \rvert}$
f_3	Frequent words (fw)	$\frac{\sum_{fw \in s_m} Count(gram_N)}{\lvert fw \rvert . \lvert s_m \rvert}$
f_4	Sentence position	$\frac{\lvert n/2 - m \rvert}{n/2}$
f_5	Sentence length	$1 - \frac{\lvert AL(S) - \lvert s_m \rvert \rvert}{\max(\lvert s_m \rvert)}$
f_6	Sentence similarity	$\frac{\sum_{m'=1, m \neq m'}^{n} (Sim(s_m, s_{m'}))}{\lvert s_m \rvert}$
f_7	Numerical data (nd)	$\frac{\sum_{nd \in s_m} Count(gram_N)}{\lvert nd \rvert . \lvert s_m \rvert}$

Table 1: Description of Text features

As a result, every sentence $s_m = [f_1, f_2, \ldots, f_7]$ is converted into a vector of seven elements where each element denotes the score of j^{th} feature of m^{th} sentence in numerical form. Next, we calculate the score of each sentence s_m using fuzzy inference system. This system requires a set of if-then rules to process the data which has been generated through training corpus. As a result, every sentence has been assigned a score and accordingly extracted the sentence for summary generation.

3 Training data for summarization

Data-driven fuzzy rules generation model requires a large corpus of documents with the labels indicating linguistic informations for every text feature and their respective reference summaries. Therefore, we have created an annotated data for training. In this regard, we have extracted 90% of documents from DUC2002 dataset. The 10% of the dataset has been preserved for testing of summarization model. At first, we pre-process the data

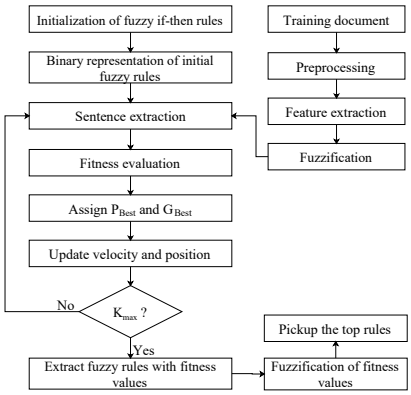

Figure 1: Data-driven fuzzy rules generation

and then compute the scores of text features as given in Table 1. These scores of text features are then used as input for fuzzification process. We use trapezoidal membership function for each input which has been described in Section 5. As a result, every score is represented in the form of linguistic information and every sentence is represented as a set of linguistic informations.

4 Fuzzy evolutionary learning approach

In this work, fuzzy evolutionary learning has been used to find a set of significant fuzzy rules. The fuzzy if-then rules are extracted as linguistic information with the association of fuzzy to discrete PSO algorithm. The flow chart of fuzzy rules generation is illustrated in Figure1.

4.1 Rule encoding

At first, we encode the fuzzy rules. Three linguistic informations (low, medium, and high) have been used to represent each text feature in the rule. These linguistic informations are encoded as 1: Low, 2: $Medium$, and 3: $High$. Therefore, if a rule is, for example, 'If f_1 is Low, f_2 is $Medium$, f_3 is low, f_4 is $High$, f_5 is $High$, f_6 is $High$, f_7 is Low then S_m is $important$ '. This rule can be encoded as 1213331.

4.2 Initialization

The initial encoded form of if-then rules are generated randomly in the range of $[0, 3]$ in the form of population $X = \{x_1, x_2, \ldots, x_p\}$, where $x_q = \{x_{q,1}, x_{q,2}, \ldots, x_{q,dim}\}$ denotes q^{th} rule in the swarm, each element of x_q is denoted by $x_{q,j}$, $q = \{1, 2, \ldots, p\}$ and $j = \{1, 2, \ldots, dim\}$, and population size p. In particular, we generate the

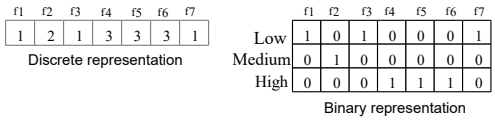

Figure 2: Fuzzy rules representation

rules in the context of text features and seven text features are considered here. Therefore, the dim of every particle is seven. Next, since the elements of every particle is in discrete form as shown in Figure 2, the notion of discrete particle swarm optimization (DPSO) proposed by Izakian et al. (2010) is applied here. In particular, each particle is converted into binary matrix form, as position update with discrete values is an issue in PSO. In the proposed method, every particle is converted into 3×7 matrix as shown in Figure 2. Along with this, the parameters used in DPSO algorithm are also set.

4.3 Sentence extraction

After the initialization of rules, the sentences in the training corpus that follows the rule described in a particle are extracted. There can be any number of sentences that follows a rule. These sentences are then used for evaluation of the rule.

4.4 Fitness function

In this step, we use ROUGE-1 recall function as the fitness function. It calculates the fitness value by matching one to one gram between the system summary and reference summary. It is defined as follows.

$$Fitness(x_q) = \frac{\sum_{s \in Sum_{ref}} \sum_{gram_1 \in s} Count_{match}(gram_1)}{\sum_{s \in Sum_{ref}} \sum_{gram_1 \in s} Count(gram_1)} \quad (1)$$

where Sum_{ref} represents the reference summary. $Count_{match}(gram_1)$ represents the matching grams between the selected set of sentences and the reference summary. The solution having the highest fitness value is marked as the global best solution.

4.5 Particle update

Here, we explain the updation of the particles in the matrix form. Similar to PSO, in DPSO, each element in the matrix of a particle is calculated by the Eq.2 and position of each matrix is updated on the basis of velocity matrix as given in Eq.3.

$$V_q^{k+1}(t,j) = V_q^k(t,j) + c_1 r_1 (pbest_q^k(t,j) - X_q^k(t,j)) + c_2 r_2 (gbest^k(t,j) - X_q^k(t,j)) \quad (2)$$

$$x_q^{k+1}(t,j) = \begin{cases} 1, & \text{if } (V_q^{k+1}(t,j) = \max V_q^{k+1}(t,j)) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

where $V_q^{k+1}(t,j)$ is the element of the t^{th} row and the j^{th} column of the q^{th} velocity matrix at the $(k+1)^{th}$ iteration. $x_q^{k+1}(t,j)$ is the element of the t^{th} row and the j^{th} column of the q^{th} position matrix at the $(k+1)^{th}$ iteration. c_1 and c_2 are positive acceleration constants, and r_1 and r_2 are random numbers belonging to [0,1]. Here, the position updating equation represents that the value 1 is assigned to those element in a column whose corresponding velocity element has maximum value in that column. If two velocity elements belongs to maximum value in a column then any one has been randomly chosen for assigning the value 1.

4.6 Termination of algorithm

If the number of user-defined iterations is over, all the generated rules in the swarm which are evaluated in the optimization process are extracted with their fitness values. These values are again fuzzified in another class of linguistic information (important, average, and unimportant) in the same manner as given in Section 5. Here, we again use trapezoidal membership function for fuzzification. Finally, all the rules which are identified as important rules are extracted for summary generation. If user-defined iterations is not over then new velocity vector is calculated again to update the velocity using Eqs. 2 and so position matrix.

5 Summary generation

Once the fuzzy rules are generated, the summary of test documents has been generated using fuzzy inference system. Similar to training documents, we pre-process the test documents and extract the features of the sentences. Next, the following process has been done to accomplish the summarization task.

5.1 Fuzzification

Three fuzzy set are considered for fuzzification: low, medium, and high. For every input, say

f_j, we use trapezoidal membership function for fuzzifing values of text features. Each membership function consists of four parameters $(\alpha, \beta, \gamma, \delta)$. With these definitions, a trapezoidal membership function $\mu_{ij}(f_j) \rightarrow [0,1]$ for i^{th} fuzzy set on the j^{th} input variable can be defined with the condition $\alpha_{ij} \leq \beta_{ij} \leq \gamma_{ij} \leq \delta_{ij}$ as follows.

$$\mu_{ij}(f_j) = \begin{cases} \frac{f_j - \alpha_{ij}}{\beta_{ij} - \alpha_{ij}}, & \text{if } \alpha_{ij} < f_j < \beta_{ij} \\ 1, & \text{if } \beta_{ij} < f_j < \gamma_{ij} \\ \frac{\delta_{ij} - f_j}{\delta_{ij} - \gamma_{ij}}, & \text{if } \gamma_{ij} < f_j < \delta_{ij} \\ 0, & Otherwise \end{cases} \quad (4)$$

5.2 Inference

In this step, the facts resulted in fuzzification process are exploited with the generated data-driven if-then rules to perform the fuzzy reasoning process. We again use trapezoidal membership function for every output which is defined as $\mu'_i(\alpha_i, \beta_i, \gamma_i, \delta_i)$.

5.3 Defuzzification

It is the process of transforming the fuzzy results obtained from inference process into crisp values which has been accomplished using centroid method $Z = \frac{\sum_{j=1}^{q} Z_j u_c(z_j)}{\sum_{j=1}^{q} u_c(z_j)}$ (Sivanandam et al., 2007), where $u_c(Z_j)$ represents the membership in class c (output fuzzy set denotes to class of s_m) at value Z_j (value obtained from fuzzy rules). Hence, the final score of the sentence Z is obtained by their all fuzzy scores and membership degrees. According to these scores, top l sentences are extracted and reordered according to the original document for summary generation.

6 Evaluations and Results

The experiment for evaluating the proposed method is conducted on the DUC2006 (Document Understanding Conference) dataset. This is a benchmark datasets in the field of text summarization that contain the document collections along with reference (human generated) summaries.

The extensively used evaluation methods for summarization system, are known as ROUGE (R) (R-N, R-L, and R-SU), proposed by Lin (2004), has been used in this study. We have computed recall matrix for every evaluation method to show the performance of the proposed method.

We have compared the performance of our method with four other methods: MsWord, NN-SE (Cheng and Lapata, 2016), FEOM (Song et al., 2011), and UniRank (Wan, 2010). MsWord is a benchmark summarizer which is extensively used for summarization. NN-SE is a deep learning based method. FEOM is a fuzzy evolutionary based method in which sentences of document is clustered and elite sentences from each cluster are extracted for summary generation. UniRank is a graph based method which verify the mutual influence between two tasks for text summarization.

Methods	R-1	R-2	R-L	R-SU
Proposed	0.482	0.239	0.401	0.312
MsWord	0.439	0.201	0.382	0.267
NN-SE	0.473	0.233	0.329	0.291
FEOM	0.477	0.224	0.417	0.303
UniRank	0.463	0.231	0.397	0.307

Table 2: Evaluation results on DUC2006

Table 2 shows the comparison between proposed method and other methods using R-1, R-2, R-L, and R-SU on DUC2006 dataset. We observed that the proposed method performed better in the case of R-1, R-2, and R-SU. However, in the case of R-L, FEOM got better result and was outperformed by 3.8%. The proposed method obtains maximum improved results by 18.2% and 16.8% in case of R-2 and R-SU with Msword. NN-SE got second position for R-1. It also perform slightly better than UniRank method in the same case. By observing these results we found that the performance of FEOM is very close to proposed method. So, paired t-test experiment is performed for these methods to find statistical differences between their performances at 5% significance level. We found p-value=0.042 which shows the significant performance of the proposed method. Overall, our proposed model achieves better performance in comparison to other methods.

7 Conclusion

We explore fuzzy swarm intelligence based an effective model for fuzzy rules generation in the context of text summarization. The effective optimization power of PSO bring a surge of interest in this task. The proposed approach considerably outperforms other methods on DUC dataset. For future work, we are planning to enhance our rule generation model by creating a large corpus for its better training. We are also planning to apply our model for multi-document summarization.

References

Razieh Abbasi-ghalehtaki, Hassan Khotanlou, and Mansour Esmaeilpour. 2016. Fuzzy evolutionary cellular learning automata model for text summarization. *Swarm and Evolutionary Computation*, 30:11–26.

Mohammed Salem Binwahlan, Naomie Salim, and Ladda Suanmali. 2010. Fuzzy swarm diversity hybrid model for text summarization. *Information processing & management*, 46(5):571–588.

MS Binwahlan, N Salim, and L Suanmali. 2009. Intelligent model for automatic text summarization. *Information Technology Journal*, 8(8):1249–1255.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Harold P Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J Simske, and Luciano Favaro. 2013. Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications*, 40(14):5755–5764.

Hesam Izakian, Behrouz Tork Ladani, Ajith Abraham, Vaclav Snasel, et al. 2010. A discrete particle swarm optimization approach for grid job scheduling. *International Journal of Innovative Computing, Information and Control*, 6(9):1–15.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics.

Inderjeet Mani. 2001. *Automatic summarization*, volume 3. John Benjamins Publishing.

Masrah Azrifah Azmi Murad and Trevor Martin. 2007. Similarity-based estimation for document summarization using fuzzy sets. *International Journal of Computer Science and Security*, 1(4):1–12.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM.

SN Sivanandam, Sai Sumathi, SN Deepa, et al. 2007. *Introduction to fuzzy logic using MATLAB*, volume 1. Springer.

Wei Song, Lim Cheon Choi, Soon Cheol Park, and Xiao Feng Ding. 2011. Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. *Expert Systems with Applications*, 38(8):9112–9121.

Xiaojun Wan. 2010. Towards a unified approach to simultaneous single-document and multi-document summarizations. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1137–1145. Association for Computational Linguistics.

Hongyuan Zha. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 113–120. ACM.