A Comparative Study of Classifiers for Extractive Text Summarization

Chapter · March 2020

DOI: 10.1007/978-981-15-1884-3_16

3 authors, including:

Anshuman Pattanaik
KIIT University
5 PUBLICATIONS **80** CITATIONS
SEE PROFILE

M.N Das
KIIT University
28 PUBLICATIONS **316** CITATIONS
SEE PROFILE

A Comparative Study of Classifiers for Extractive Text Summarization

Anshuman Pattanaik, Sanjeevani Subhadra Mishra and Madhabananda Das

Abstract Automatic text summarization (ATS) is a widely used approach. Through the years, various techniques have been implemented to produce the summary. An extractive summary is a traditional mechanism for information extraction, where important sentences are selected which refers to the basic concepts of the article. In this paper, extractive summarization has been considered as a classification problem. Machine learning techniques have been implemented for classification problems in various domains. To solve the summarization problem in this paper, machine learning is taken into consideration, and KNN, random forest, support vector machine, multilayer perceptron, decision tree and logistic regression algorithm have been implemented on Newsroom dataset.

Keywords Text summarization · Extractive · Sentence scoring · Machine learning

1 Introduction

A compact version of the original text which produces the same concept as the original document is known as summary. In 1958, Luhn introduces the concept of abstract generation out of text data. That gives boost to the idea of automatic text summarization [1, 2] (ATS). In recent years, high availability of text data helps in growth of natural language processing, especially in the field of text summarization. Summarization can be of two categories, such as abstractive and extractive. Extractive text summarization is one of the oldest and widely used approaches among researchers. The idea behind extractive summarization is to extract the sentences

A. Pattanaik (✉) · S. S. Mishra · M. Das
School of Computer Engineering, Kalinga Institute of Industrial Technology (Deemed-to-be University), Bhubaneswar, India
e-mail: anshumanpattanaik21@gmail.com

S. S. Mishra
e-mail: sanjeevani321@gmail.com

M. Das
e-mail: mndas_prof@kiit.ac.in

D. Swain et al. (eds.), *Machine Learning and Information Processing*,
Advances in Intelligent Systems and Computing 1101,
https://doi.org/10.1007/978-981-15-1884-3_16

from the document in a way that those sentences will represent the core idea of the document [2]. Extractive summary is the basic summarization technique. Sentences are selected in the basis of some scores and ranks. Scoring and ranking of sentences are done by feature mapping and selection. Features can be of different types such as frequency-based and prediction-based. Frequency-based features are more widely used in extractive summarization. Summarized data can be helpful in various fields. A summarized document helps in understanding the whole document in less amount of time. One can find relevant document from a query search more faster by going through summaries. In case of scientific data or medical data, one can easily produce a report through the summaries. Summaries can help in creating proper indexing of multiple documents in much lesser time. Several methods have been implemented over the years for creating improved summaries. Machine learning is one of the best suited methods for resolving classification and clustering-based problem nowadays according to the state-of-the-art approaches on different fields. This work presents a technique for extractive summarization as a classification problem instead of an information extraction problem. Extractive summary has been considered as a two-class problem where each sentences can be considered as either 0 or 1. If the sentence is not selected for the summary, then it is classified as 0 else 1. Different machine learning techniques have been implemented on the same dataset, and comparisons have been made.

Rest of the paper is organized as follows: Sect. 2 represents related works, Sect. 3 explains the proposed method, Sect. 4 presents experimental setup and result analysis. Finally, we have concluded our work in Sect. 5.

2 Related Works

Automatic text summarization (ATS) is a technique to generate summary from a provided text data. Statistical algorithms, graph-based algorithms, etc., are used by ATS to generate the desired summary. These multiple algorithms use specified mathematical models and computational devices. ATS has a wide range of applications, larger diversification and is quite reliable in generating the requisite summary. Hence, it has attracted the attention in the field of research and development. Foreseeing the advantages, the researchers have invested a great effort in modifying and developing ATS techniques.

In extractive summarization, sentence selection is the main criteria. Frequency-based methods and predictive methods are available for sentence selection. In 2014, Meena and Gopalani [3] gave an analysis on different frequency-based features such as term frequency, TF-IDF, sentence location, title similarity, proper noun, word co-occurrence, numerical values in sentences and sentence length. In their work, they have concluded that in most cases TF-IDF, word co-occurrence, sentence length and location give better performance together. Researchers can consider this in case of extractive summarization as a combined parameter. However, more number of different combinations can be taken into consideration.

Pattanaik et al. [4] considered the extractive summarization as an optimization problem. The objectives are to find an optimized solution for the given text considering high coverage of context and lower redundancies between the output sentences. BAT algorithm outperforms the existing model in their experiment. They have taken TF-IDF and sentence similarity as their sentences selection features.

Machine learning algorithms have been amazing when it comes to prediction or classification of any data. Naïve Bayes classifier, support vector machine, kernel support vector machine, decision tree, logistic regression, etc., techniques are widely used in different fields for prediction.

Joachims [5], in 1998, introduced a classification mechanism to text categorization. Text categorization is a standard classification problem where input data is text articles, and output is a set of categories out of which text data will fall into either one or many categories. So, this dataset is not linearly separable. Author explained that the concept of high dimensionality of feature vector is the factor for which general machine learning mechanism fails to achieve the desired output. Support vector machine transcends the state-of-the-art techniques.

3 Proposed Method

In this work, extractive summarization is treated as a two-class classification problem, where each sentence of the document either falls under 0 or 1 class. If a sentence is not selected in the summary, it is considered as class 0 and as class 1 if selected. Each sentence of the document goes through the classifier for prediction. In this work, different machine learning classification models are tested over the same dataset. The proposed model work flow is explained in Fig. 1.

The input document goes through preprocessing. In this phase, highly unstructured text data goes through data cleaning (stemming, tokenizing, stop word removal). After that, the cleaned data is converted to numerical form with different frequency-based scoring. Numerically presented data flows through the machine learning model for training and testing. TF-IDF [6, 7], keywords [8] and sentence length are the parameters taken into consideration for the feature selection. TF-IDF is one of the sentence scoring methodologies where it focuses on the rare terms that appear in any document and have some more weight regarding the query. For single document, it calculates from term frequency inverse sentence frequency (TF-ISF). For multiple document, it uses term frequency inverse document frequency (TF-ID).

$$\mathrm{TF-IDF}(\omega) = \frac{\mathrm{tf}}{(\mathrm{tf}+1)} \log\left(\frac{N}{\mathrm{df}}\right) \tag{1}$$

where tf is term frequency, N is for number of total document, and df stands for document frequency. Higher the tf/idf value rarer the term in the document and has more weight for term occurrence in summary section. In the ML modeling phase, every sentence of the document is first trained with the model. The labeled data,

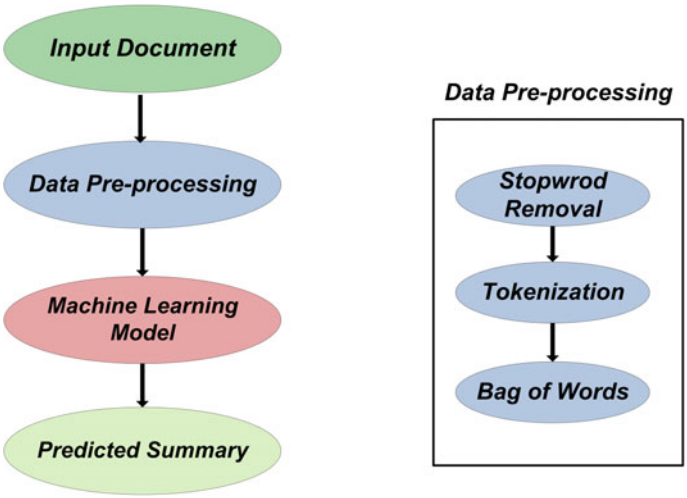

Fig. 1 Workflow of proposed model

i.e., the human generated extractive summaries are the extracted sentences from the original text. It can be considered as two class problem with class value 0 and 1. The value 0 refers to the sentences those are not in the summary where as, value 1 presents sentences available in human generated summary.

In case of machine learning modeling, K-nearest neighbor (KNN), random forest, support vector machine, multilayer perceptron, decision tree and logistic regression models have been implemented. KNN algorithm works on grouping approach by considering the k-nearest majority voting. In case of support vector machines, it handles large dimensions more precisely and also removes irrelevant features easily as explained by Joachims [5] in 1998. Decision tree classifies categorical values more accurately. Multilayer perceptron is a feed forward network which trains through back propagation.

4 Experimental Setup and Result Analysis

Generating extractive summary from the text data is a high computing task when it involves machine learning algorithm. Table 1 elaborates about every specification that author have taken into consideration. Newsroom [9] is one of the widely used dataset for text summarization. This dataset consists 1.3 million news articles and associated summaries by various authors and editors. Extractive summary articles are taken into account, and experiment has been done on these data as explained in Sect. 3. In this dataset, extractive, abstractive and mixed categories of summaries are available in different volumes. In this work, only extractive summaries are taken into account. K-nearest neighbor, random forest, support vector machine, multilayer

Table 1 Experimental setup

Hardware specification	Programming language used	Software specification	Dataset	Sentence selection feature	Machine learning classifier
Intel-core i5 7th generation processor 2.71 GHz clock cycle 32 GB RAM 1 TB HDD	Python 3.7	Windows 10 64 bit Anaconda Distribution 64 bit	NewsRoom [9]	TF-IDF Keyword Sentence length	KNN Random forest SVM with RBF MLP Decision tree Logistic regression

perceptron, decision tree and logistic regression classifiers have been implemented. In Table 1 experimental setup, software used and models used are given for better understanding of the work.

In K-nearest neighbor, ten neighbors are taken into consideration with uniform weights. In random forest 1000, 10,000 n_estimators are taken into consideration with 'gini' criterion for splitting quality of the data is used. In decision tree, "entropy" is used as criterion function. In support vector machine, radial basis function is used as kernel function which focuses on worst case class splitting. Multilayer perceptron classifier is used with 100 hidden layers, having RELU activation function with stochastic gradient-based optimizer for the back propagation and weight updating solver. A classification problem analyzes by its classification report [10] which includes precision, recall and F1-score values. Precision is the score which indicates what percent of your prediction was correct. Recall indicates how much positive prediction is done by the classifier. F1-score can be calculated by taking weighted harmonic mean of recall and precision. Table 2 illustrates classification scores for each class. 0 indicates the sentences are not in the summary, and 1 indicates sentences are in the summary.

True Negative (TN): originally negative and predicted negative
True Positive (TP): originally positive and predicted positive
False Negative (FN): originally positive but predicted negative
False Positive (FP): originally negative but predicted positive

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{3}$$

$$F1\text{-Score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{4}$$

Table 2 Classification report for K-nearest neighbor, random forest, support vector machine, multilayer perceptron, decision tree and logistic regression classifiers

		0	1
K-nearest neighbor (KNN)	Precision	0.78	0.75
	Recall	0.92	0.49
	F1-score	0.84	0.59
Random forest (RF)	Precision	0.80	0.61
	Recall	0.82	0.58
	F1-score	0.81	0.60
Support vector machine (SVM)	Precision	0.78	0.74
	Recall	0.92	0.58
	F1-score	0.84	0.58
Multilayer perceptron (MLP)	Precision	0.80	0.75
	Recall	0.91	0.54
	F1-score	0.85	0.63
Decision tree (DT)	Precision	0.78	0.57
	Recall	0.78	0.57
	F1-score	0.78	0.57
Logistic regression (LR)	Precision	0.78	0.74
	Recall	0.92	0.47
	F1-score	0.84	0.58

In this paper, six different types of classifiers were being examined. Multilayer perceptron classifier seems to perform better in terms of precision, recall and f1-score for class 1. Multilayer perceptron has the advantage of hidden layer which provides some meta-data to the classifier, and the back propagation mechanism keeps the classifier more accurate. In the following tables, i.e., from Tables 3, 4, 5, 6, 7 and 8, confusion matrix values for all the six classifiers. Confusion matrix gives the brief idea about how much data predicted correctly.

Receiver operating characteristic (ROC) [11] curve is a widely used curve to analyze the performance of classifiers. ROC is a curve between true positive rate and false positive rate of any classifier. The curve having more area under the curve

Table 3 Confusion matrix: KNN

	Predicted 0	Predicted 1
Actual 0	402	36
Actual 1	112	107

Table 4 Confusion matrix: RF

	Predicted 0	Predicted 1
Actual 0	358	80
Actual 1	93	126

Table 5 Confusion matrix: SVM

	Predicted 0	Predicted 1
Actual 0	402	36
Actual 1	114	105

Table 6 Confusion matrix: MLP

	Predicted 0	Predicted 1
Actual 0	399	39
Actual 1	101	118

Table 7 Confusion matrix: DT

	Predicted 0	Predicted 1
Actual 0	343	95
Actual 1	94	125

Table 8 Confusion matrix: LR

	Predicted 0	Predicted 1
Actual 0	402	36
Actual 1	115	104

(AUC) is better than others. In Fig. 2, ROC curve of different classifiers is given. Multilayer perceptron classifiers cover more area under the curve as compared to other classifiers. KNN, random forest, logistic regression and support vector machine are having same AUC value.

5 Conclusion and Future Work

In this paper, the authors considered the extractive text summarization problem as a classification problem. A document is classified under a two-class problem. Class 0 indicates that the sentences in the document are not considered as summary sentences. Class 1 is the class of summary sentences. TF-IDF, keywords and sentences length are taken into consideration for evaluation. K-nearest neighbor, random forest, support vector machine, multilayer perceptron, decision tree and logistic regression classifiers have been implemented on Newsroom dataset. The experimental analysis of the algorithms with precision, recall, f1-score and confusion matrix of all the classifier is mentioned in Tables 2, 3, 4, 5, 6, 7 and 8. Confusion matrix is calculated over the test data, which indicates the polarity of data according to the classifiers. The ROC curve is also plotted. All the analyses state that although above classifiers do not satisfy the goal more accurately, and out of them multilayer perceptron classifier gives better result. MLP has hidden layers and back propagation principle which

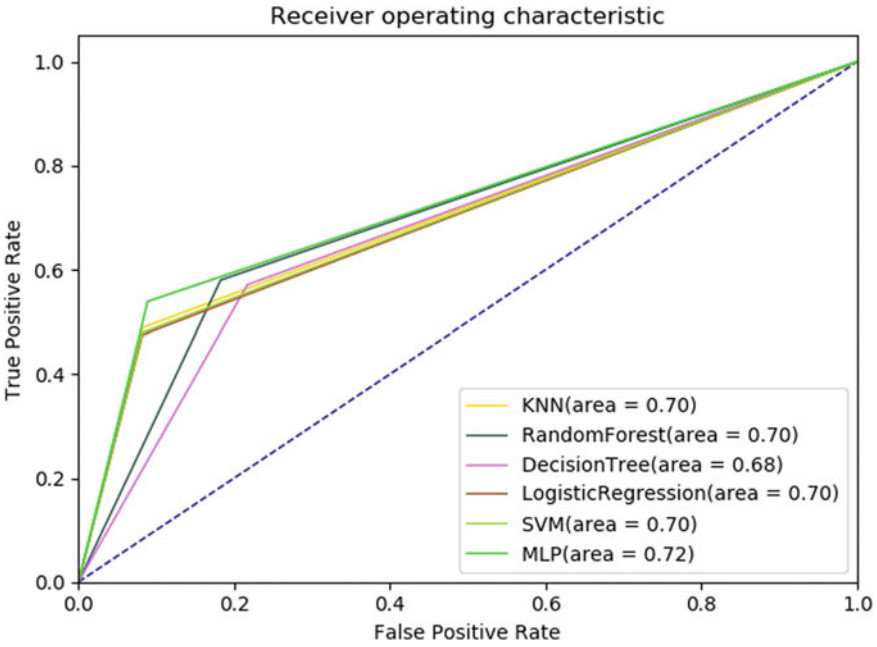

Fig. 2 ROC curve for different classifiers

enhances the quality of the classifier. In this work, MLP provides 72% accuracy. This work can be considered for future work by modifying different parameters and adding more features to the input vectors.

References

1. Luhn, H.P. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2 (2): 159–165.
2. Gambhir, M., and V. Gupta. 2017. Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review* 47 (1): 1–66.
3. Meena, Y.K., and D. Gopalani. 2014. Analysis of sentence scoring methods for extractive automatic text summarization. In *Proceedings of the 2014 international conference on information and communication technology for competitive strategies*, November 2014, 53. ACM.
4. Pattanaik, A., S. Sagnika, M. Das, and B.S.P. Mishra. 2019. Extractive summary: An optimization approach using bat algorithm. *Ambient communications and computer systems*, 175–186. Singapore: Springer.
5. Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, April 1998, 137–142. Springer, Berlin, Heidelberg.
6. Nobata, C., S. Sekine, M. Murata, K. Uchimoto, M. Utiyama, H., and Isahara. 2001. Sentence extraction system assembling multiple evidence. In *NTCIR*.

7. Jafari, M., J. Wang, Y. Qin, M. Gheisari, A.S. Shahabi, and X. Tao. 2016. Automatic text summarization using fuzzy inference. In *22nd International conference on automation and computing (ICAC)*, September 2016, 256–260. IEEE.
8. Matsuo, Y., and M. Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools* 13 (01): 157–169.
9. NewsRoom Dataset Available (2017) Cornell Newsroom. https://summari.es. 2017.
10. Powers, D.M. 2011. *Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation*.
11. Davis, J., and M. Goadrich. 2006. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on machine learning*, June 2006, 233–240. ACM.