iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** XII **Month of publication:** December 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 5 Issue XII December 2017- Available at www.ijraset.com

Condensing Text using Bagging and Boosting

Maya John[1], Jayasudha J. S.[2]
[1]*Department of Computer Science and Engineering, Noorul Islam University, Tamil Nadu, India.*
[2]*Department of Computer Science and Engineering , Sree Chitra Thirunal College of Engineering, Kerala, India.*

Abstract: With the unprecedented growth of internet there has been a rather explosive growth in the quantum of textual information present in the network of networks. In many occasions users are not interested in reading the whole textual content present in web pages. Hence there has been an ever increasing demand in developing effective methods to summarize text. This paper deals with extractive summarization based on bagging and boosting methods. Experiments were conducted using methods such as AdaBoost.M1, Real AdaBoost, Bagging, Gradient Boosting Machine (GBM), Generalized Linear Model Boost (GLM Boost) and eXtreme Gradient Boosting (XGBoost). The performance of summarization methods were evaluated using metrics such as F-Measure, G-Mean and AUC and it is observed that AdaBoost based method outperformed other summarization techniques.

Keywords: Feature Extraction, Boosting, Bagging, Classification, Summarization

I. M. A. FATTAH AND F. REN, "GA, MR, FFNN, INTRODUCTION

Automatic text summarization deals with making use of computers to generate condensed form of given text by retaining the important contents of the text. Text summarization is a challenging area in the field of natural language processing on account of the issues associated with it. Large number of studies are being conducted to devise methods to generate summary with less issues and closest to the way in which humans perceive. Summary generated may be abstractive or extractive in nature. Extractive summary is generated by using phrases and sentences from given text. In the case of abstractive summary, the summary generated contains the main contents of the text presented in a form different from that in the given text. The summary may be generated either giving equal importance to all contents in the text (generic summary) or by focussing on certain parts of the given text (query focussed) [1]. The summarization process is based on either supervised or unsupervised method [2]. In supervised method before summarization the summarizer has to learn from the training data.

Many works have been reported for summarizing the text by extracting the important sentences. Naives Bayes classifier [3], log linear methods [4] etc. can be used to identify important sentences in a piece of text. Shallow linguistic features can be used to generate text summaries [5]. Restricted Boltzmann Machine (RBM) based deep learning technique can be used to generate text summary [6]. A hybrid method employing a trained summarizer and latent semantic analysis is effective in summarizing text [7]. The significant sentences in a piece of text can be identified using statistics based on text features and contextual information [8]. AdaBoost based text summarizer is more effective than J48 based summarizer and multi layer perceptron based summarizer [9].

II. PROPOSED SYSTEM

The proposed system generates extractive summary of the text contained in the web page using supervised method. Here classification technique is used to extract summary sentence from a given piece of text. The three phases involved on the process of text summarization are text preprocessing, feature extraction and classification.

A. Text Preprocessing

The three major steps involved in the preprocessing of the text are segmentation of text, removal of reference information from sentences and removal of url or email address from sentences. Segmentation deals with identifying sentences boundaries and hence divided the given text into a collection of sentences. The references information present within sentences are removed as mentioned in paper [10]. The reference, url and email information are removed from sentences so that they are not tokenised later. In order to facilitate extraction of sentence features after removal of aforesaid contents from sentences, the sentences are divided in to basic units known as tokens. After tokenization the stop words are removed from sentences and the remaining words are lemmatized and stemmed.

B. Feature Extraction

Feature extraction deals with extracting 22 features from each sentence [10]. The details regarding the features extracted are given below:

1) Based on the position of sentence in a given text a feature value is computed. Sentences in the beginning of the text are given more score than later sentences.
2) Sentences which are too short have less chance of being a summary sentence. Taking this into consideration a feature value is computed by comparing length of a sentence with that of longest sentence.
3) The third feature is computed as the fraction of numeric data in sentences.
4) Sentences which contain cue phrases tend to be important [11]. Examples of cue phrases are in short, to sum up, therefore etc. A binary feature is computed to indicate the presence or absence of cue phrase in sentence.
5) The presence of nouns, proper nouns, adjectives in sentences signify that the sentence is important. Three separate features are used to indicate the relative length of sentence in terms of noun/ proper noun/ adjective count.
6) The most frequent ten words in the text are considered as keywords. A feature is computed based on how many words in sentences are keywords.
7) The <title> tag, <h1> tag and <meta> tag contain words which give indication regarding the main content of the text. Hence three separate features are computed based on similarity between the just mentioned tags and the tokens in sentences.
8) The degree of attachment of a sentence with other sentences are computed using cosine similarity.
9) As sentences having url or email id tend to more significant than other sentences, a binary feature is computed to indicate the presence or absence of url/ email in sentence.
10) Contents written within parenthesis in sentences are mere extra information. Hence effective length of sentence ignoring the contents within parenthesis is computed.
11) A sentence is more important if it has information written within quotation marks. A feature is computed on the basis of length of sentence with respect to length of contents written within quotation marks.
12) Binary feature is used to indicate the presence or absence of pronoun in a sentence.
13) Sentences which have information regarding week day or month are considered to be more important. Hence we have used two binary feature to indicate if week day or month is present in a sentence.
14) Sentence may contain words which are indicative of non-essential information. Example of such words are additionally, furthermore etc. A binary feature is assigned value one if just mentioned class of information is not present in sentence and assigned zero otherwise.
15) Mood and modality of a sentence can be found out by employing tools such as Pattern. The system computes two features based on mood and modality of a sentence.
16) Mean term weight of sentences are computed based on the value of term frequency-inverse sentence frequency (TF-ISF).

C. Classification

The proposed system employs bagging and boosting based methods to classify sentences in given text into summary sentences and non-summary sentences. The classification process is pictorially depicted in Fig. 1. The classification

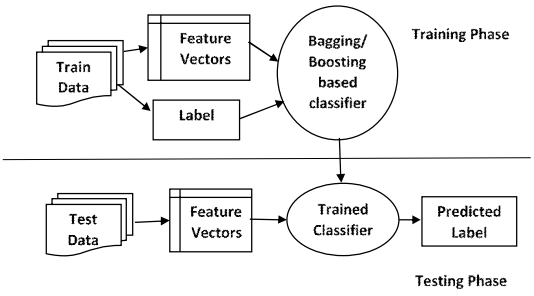


Fig. 1 Diagrammatic representation of classification process

process is divided into two phases namely training phase and testing phase. Sentences in the dataset are labelled as summary sentence or non-summary sentence. During the training phase feature vector and label of each sentence in the train data are given as input to the bagging/boosting based classifier. The classifier learns from the input it receives. In the testing phase feature vectors of the test data are given as input to the trained classifier which predicts whether a sentence belongs to summary or not. The classification is done using methods such as AdaBoost.M1, Real AdaBoost Bagging, GLMBoost, XGBoost and GBM and their performances are analysed.

a) *AdaBoost.M1:* AdaBoost is based on the fact that strong learners can be obtained by combining weak learners[12]. The salient features of AdaBoost include reduction of variance and capability to boost the margins. In this technique the complete data set is fed as input to the classifiers in serial fashion. Each classifier strives to achieve a lower misclassification rate when compared to its predecessor. In this method more attention is given to the sentences which are not classified properly. Here initially equal weight is assigned for all instances. As the boosting proceeds the weight of misclassified instances are increased while weight of rest sentences are reduced. On the basis of classification accuracy a weight is assigned to each classifier. When a piece of text is given to the AdaBoost based summarization system, each classifier assigns a weighed vote for each sentence and the sentence is classified based on majority vote. AdaBoost.M1 is an improved version of AdaBoost where error rate is the basis of computing base classifier weight. A disadvantage of AdaBoost.M1 is that it is applicable only where weighted error does not exceed 0.5.

b) *Real AdaBoost:* Real AdaBoost or SAMME.R is an enhanced version of SAMME [13]. SAMME is considered similar to an additive model which is forward additive in nature and employs multi-class exponential loss function. The main difference between SAMME and SAMME.R is that SAMME considers errors with regard to class label predicted while SAMME.R computes error based on class probabilities predicted. Real AdaBooost is suitable for classification involving two or more classes.

c) *Bagging:* Bagging or Bootstrap Aggregation [14] is a method which strives to enhance classification rate by means of reducing the variance of the prediction. In the case of bagging, bootstrap samples of the data are given to the individual classifiers in training phase while in the case of boosting whole data set is given to the individual classifiers. The result of classification is obtained by majority vote based on prediction results of samples.

d) *Gradient Boosting Machine:* GBM is a boosting technique based on the concept of gradient descend [15]. GBM generates base learners such that the base learners have maximum degree of relationship with the negative gradient of loss function with regards to the complete ensemble. The successive base learners aim at improving the accuracy of predictions. A salient feature of GBM is that the loss function can be chosen per the requirement [16].

e) *GLMBoost:* Generalized Linear Model (GLM) can be considered as an improvisation of ordinary linear regression which supports output variables having non-normal error distribution. The base procedure chosen for boosting aims at optimizing the predictive capacity. In certain cases the base procedure also considers structural features associated with boosting estimates. Component-wise boosting is employed by GLMBoost to fit linear models.

f) *XGBoost:* XGBoost [17] is an end to end tree based boosting technique with high degree of versatility and scalability. It is an extended version of gradient boosting. The features which make XGBoost powerful include use of innovative learning algorithm to manage sparse data, improvised sketch algorithm to deal with weighed data, optimization of cache, parallel learning based on column block and out-of-core computation using blocks. In order to achieve better degree of generalization XGBoost employs regularized learning objective. The generic objective function used has additive solution. Searching over sentences is done by means of structure score.

III. PERFORMANCE EVALUATION

CNN dataset [18] is used for text summarization. Corresponding to each text a summary is manually generated by taking into consideration summaries of evaluators. The sentences in a text are classified as summary sentences and non-summary sentences. Since the number of summary sentences are comparatively less when compared to non-summary sentences, the problem of class imbalance can arise. Due to the aforesaid fact high values of accuracy can be achieved even if the very few summary sentences are classified properly. The performance of the summarizers are compared using metrics such as F-Measure, G-Mean and AUC [19].

True Positive (TP) is the number of positive instances predicted as positive. True Negative (TN) is the number of negative instances predicted correctly. False Positive (FP) is the number of negative instances predicted as positive instance. False Negative (FN) is the number of positive instance predicted as negative. Precision, recall and F-Measure are computed using equation 1, 2 and 3 respectively. Precision also known as Positive Predictive Value (PPV) is the fraction of predicted positives which are actual positive while recall (sensitivity) is the fraction of actual positives which are predicted positive. G-Mean (Geometric Mean) is the square root of product of sensitivity and specificity. Receiver Operative Characteristics (ROC) Curve is a curve plotted with true positive rate (TPR) on the y-axis and False Positive Rate (FPR) on the x-axis. AUC refers to the area under the ROC curve. The value of AUC lies between 0 and 1. AUC value is not affected by relative distribution of class. Specificity is the fraction of actual negatives predicted as negative.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

$$\text{F-Mean} = \frac{2*Precision*Recall}{Precision+Recall} \quad (3)$$

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (4)$$

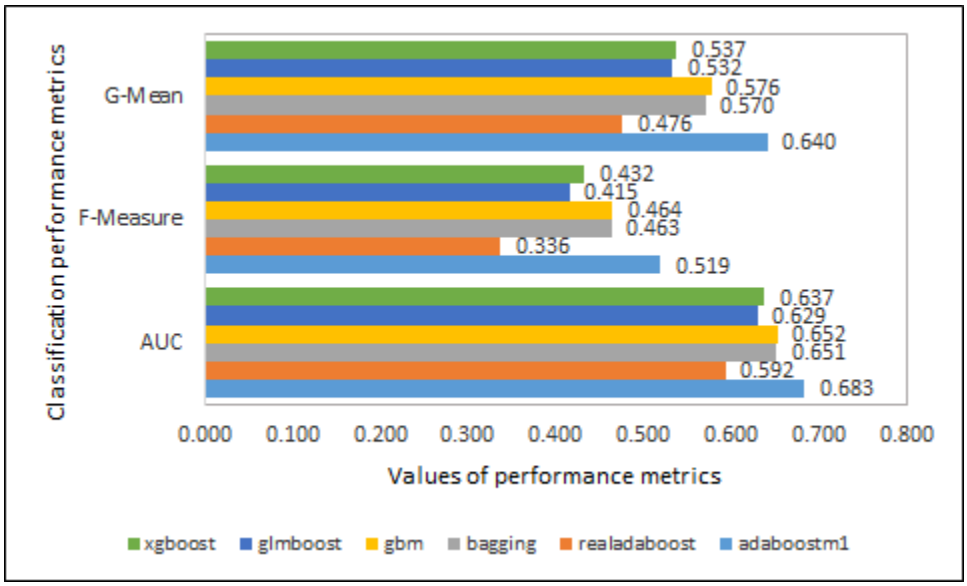


Fig. 2 Comparison of classification performance of different classifiers

The performance of different bagging and boosting methods are depicted in Fig. 2. It is evident from Fig. 2 that AdaBoost.M1 outperformed other methods in terms of AUC, G-Mean and F-Measure.

IV. CONCLUSION

This paper deals with extractive text summarization methods based on bagging and boosting. Bagging and boosting techniques employ more than one classifier to classify data. The proposed system makes use of an extensive feature set to summarize text using classification approach. Use of preprocessing methods prior to feature extraction prevents irrelevant tokens from being generated. Experiments were conducted with different summarization strategies employing the concepts of bagging and boosting. The classification methods used include AdaBoost.M1, GBM, GLMBoost, XGBoost, RealAdaBoost and bagging. The performance of the summarizers were analyzed using performance metrics such as F-Measure, G-Mean and AUC. It i observed that AdaBoost.M1 achieved best classification results. The proposed work can be improved by considering a more effective feature set. Use of various imbalance data handling methods may enhance the classification performance.

REFERENCES

[1] PNN and GMM based models for automatic text summarization," Comput. Speech Lang., vol. 23, no. 1, pp. 126–144, 2009.
[2] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using Latent Semantic Analysis," J. Inf. Sci., vol. 37, no. 4, pp. 405–417, 2011.
[3] J. Kupiec, "A Trainable Document Summarizer," in Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995, pp. 68–73.
[4] M. Osborne, "Using maximum entropy for sentence extraction," in Proceedings of the ACL-02 Workshop on Automatic Summarization, 2002.
[5] P. Gupta and V. S. Pendluru, "Summarizing text by ranking text units according to shallow linguistic features," in Proceedings of 13th International Conference on Advanced Communication Technology (ICACT2011), 2011, pp. 1620–1625.
[6] G. PadmaPriya and K. Duraiswamy, "An approach for text summarization using deep learning algorithm," J. Comput. Sci., vol. 10, no. 1, pp. 1–9, 2014.

[7] J.-Y. Yeh, H.-R. Ke, W.-P. Yang, and I.-H. Meng, "Text summarization using a trainable summarizer and latent semantic analysis," Inf. Process. Manag., vol. 41, no. 1, pp. 75–95, Jan. 2005.

[8] Y. Ko and J. Seo, "An effective sentence-extraction technique using contextual information and statistical approaches for text summarization," Pattern Recognit. Lett., vol. 29, no. 9, pp. 1366–1371, Jul. 2008.

[9] R. Belkebir and A. Guessoum, "A Supervised Approach to Arabic Text Summarization Using AdaBoost," in Advances in Intelligent Systems and Computing, A. Rocha, Ed. Springer International Publishing Switzerland, 2015, pp. 227–236.

[10] M. John and J. S. Jayasudha, "Pre-processing and Feature Extraction for Text Summarization," Fronteiras, vol. 6, pp. 508–515, 2017.

[11] J. Lawler and H. A. Dry, Using Computers in Linguistics: A Practical Guide, Routledge, 2001

[12] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci., vol. 55, pp. 119–139, 1995.

[13] J. Zhu, S. Rosset, H. Zou and T. Hastie,"Multi-class AdaBoost", University of Michigan, 2005.

[14] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140, 1996.

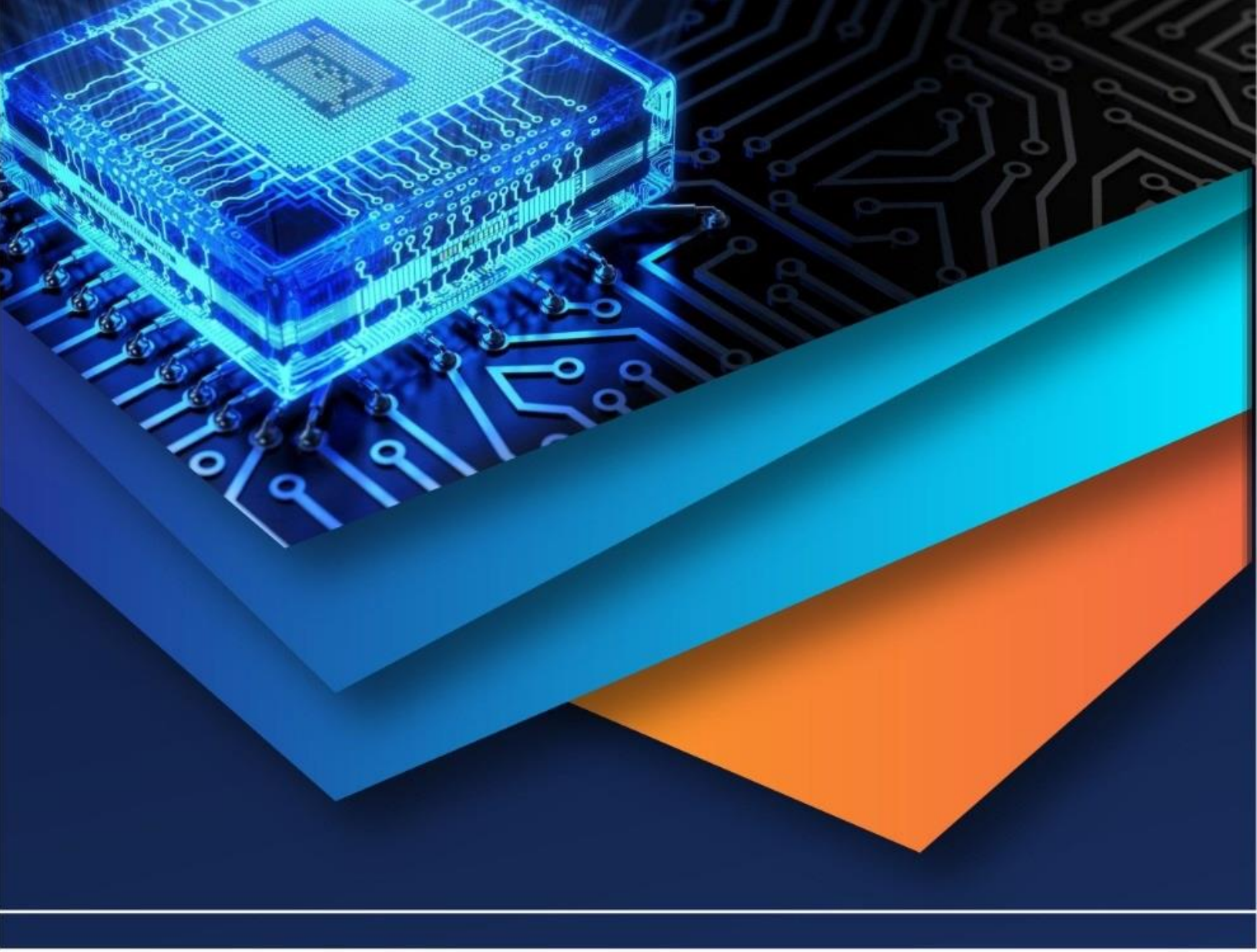
[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," Ann. Stat., vol. 29, no. 5, pp. 1189–1232, 2001.

[16] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Front. Neurorobot., vol. 7, pp. 1–21, 2013.

[17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of KDD'16, 2016.

[18] K. M. Hermann et. al., "Teaching Machines to Read and Comprehend," in Proceedings of 28th International Conference on Neural Information Processing Systems, 2015.

[19] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation Measures for Models Assessment over Imbalanced Data Sets," J. Inf. Eng. Appl., vol. 3, no. 10, pp. 27–39, 2013.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY