Text Classification Based on Weighted Extreme Learning Machine

Hayder Mahmood Salman
Al-Turath University College /Department of Computer Science
hayder4all@yahoo.com

Article history: Received in:6October2018, accepted in:1 November2018, Publish January 2019

Abstract

The huge amount of documents in the internet led to the rapid need of text classification (TC). TC is used to organize these text documents. In this research paper, a new model is based on Extreme Machine learning (EML) is used. The proposed model consists of many phases including: preprocessing, feature extraction, Multiple Linear Regression (MLR) and ELM. The basic idea of the proposed model is built upon the calculation of feature weights by using MLR. These feature weights with the extracted features introduced as an input to the ELM that produced weighted Extreme Learning Machine (WELM). The results showed a great competence of the proposed WELM compared to the ELM.

Keywords: Text Classification, Multiple Linear Regression, Extreme Machine Learning.

1.Introduction

Through the growing of social networks, a huge quantity of text data is quickly generated, the need for a well-defined methodology to analyze and classify these voluminous data has drawn many communities' attention to this kind of data which is known as unstructured data [1]. This phenomenon which made the importance of text classification begins to spring up. Text classification (TC) is the way toward assigning a document to a class by assessing its content segments. It has been applied effectively on various occasions and is incorporated in our regular day to day existences. For example, daily paper articles and scholastic papers are frequently composed by subject or field [2]. TC gives a solution in consequently arranging innumerable articles and papers. Another notable application is spam sifting. TC can help to filter out the annoying emails automatically by classifying these text emails as a spam [3].

TC model can be divided into supervised and unsupervised models. Supervised classification consists of two phases training phase and testing phase. During the training phase a set of known labeled data feeds to the machine learning algorithm. The goal of this phase is to reach the desired output by train the algorithm. Through the testing phase, a set of unknown, labeled data feeds to the algorithm to classify them into classes depending on the training phase. Unsupervised classification not required any prior knowledge about the data where the algorithms attempt to learn the input data and discover the relationship between similar data. Based on this similarity the algorithms classify the data into classes. The data classification is performed depending on the classes created. A web search is a good example of unsupervised algorithm. Depending upon the search term the algorithms makes classes that provided to the user [4]. Classification can also be divided into binary classification or multiclass classification. Binary classification attempt to arrange the components of a given input set of data into two classes based on the classification rules. Multiclass is the problem of classifying instances into one of three or more classes [5].

In research this paper a model for TC is proposed. The model consists of many phases that comprise: preprocessing, feature extraction, MLR and ELM is proposed. The main idea of the proposed model based on computing feature weights using MLR and multiplying these feature weighs by the corresponding features to introduce as input to the EML to produced WELM.

2.Related Works

In this section some of TC methods will be investigated.

In 2014, D. Renukadevi and S. Sumathi [9], proposed a method for developing of information technology and cumulative usability of internet. The proposed model consists of many phases. The preprocessing is the first phase, then the Term Frequency- inverse Document Frequency (TF-IDF) for each term is calculated to rank the document. Fuzzy c-mean algorithm applied in the last phase to allow similar data to be grouped into the same cluster. Their proposed improved the clustering accuracy and it was less classification time [6].

In 2017 Conneau et al. Build a text classification model based on the character's level. The model called Very Deep Convolutional Neural Networks (VDCNN) the model start with a look –up table that creates a 2D tensor that contain a number of the embedding character. The model starts by applying one layer of 64 convolutions of size 3, followed by a stack of convolutional blocks. Each layer has the same number of feature map, also the model contains 3 pooling operations to reduce the memory size. The model proves increasing in the performance when using up to 29 convolutional layers [7].

In 2018 Tingyi Zheng et al. Proposed a model that consists of Support Vector Machine (SVM) with Active Self-Paced Learning (ASPL). The unlabeled and very little labeled texts utilized to learn the model. The model starts by extracting features using convolutional neural network and classify few labeled texts using the SVM, then the unlabeled text will be ranked based on their significance weight. The top ranked text would be selected and produced to the ASPL. The process will be repeated and select the next top ranked text. The model shows the TC accuracy can be enhanced by using few labeled and unlabeled text [8].

3.Problem statement

TC involves of two parts: the document collection and the number of document classes. The document collection describes the scope of input document patterns. The training patterns and test patterns are selected from the document collection. The set of document classes describes the probable outputs created by the classifier and is utilized to label document patterns. Assume we have

$D=\{d_1,d_2,\ldots,d_n\}$ is a set of documents collection.

$C=\{c_1,c_2,\ldots.,c_m\}$ is a set of classes for the documents collection (D).

$\Theta_k=\{d_{k,1},d_{k,2},\ldots d_{k,l}\}$ is a set of documents with the same class.

The main goal of TC is to find Θ_k that belong to the same class (C_i).

4.Proposed Method

TC consists of many phases. **Figure 1.** shows the main phases of the proposed method. These phases include: preprocessing, Feature extraction, MLR and ELM. The proposed model based on computing features weight using MLR. The WELM based on multiplying these feature weights by the feature before entering to the ELM, that is made in turn to increase the accuracy of the WELM in compared with ELM.

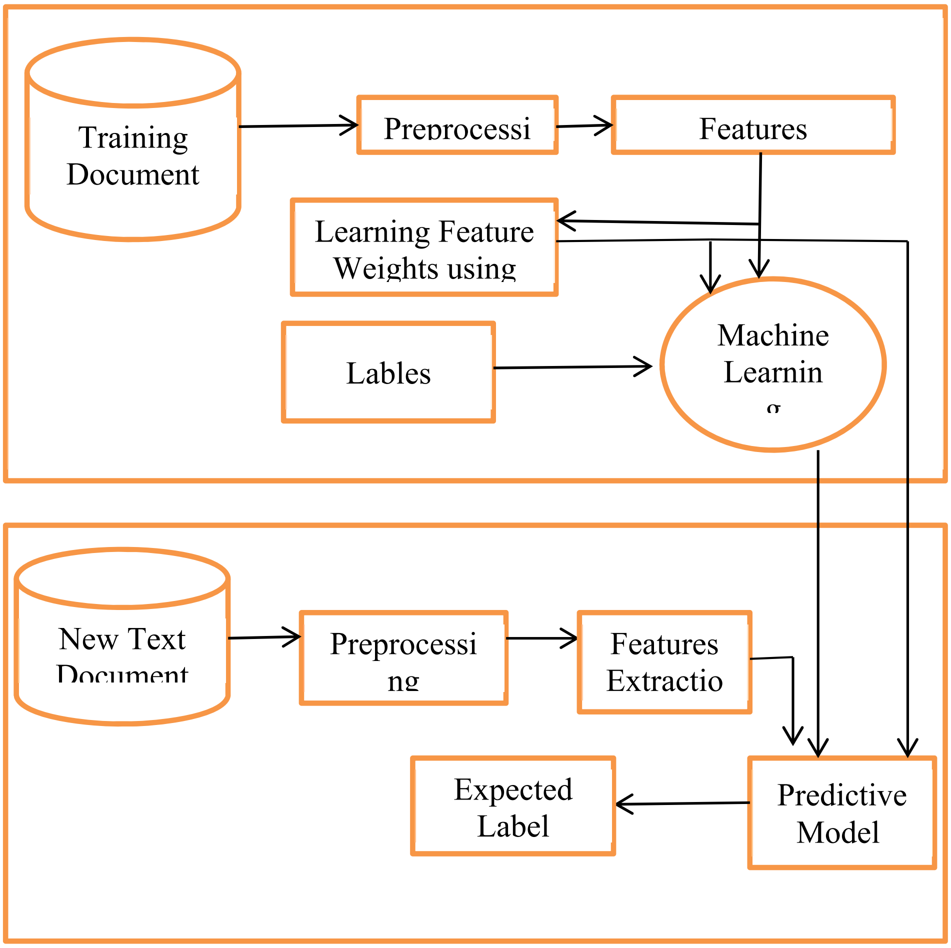

Figure 1. Block Diagram of the Proposed Method

4.1. Preprocessing

TC issues are difficult in nature and are permanently categorized via large dimensionality. To decrease this difficulty investigators, start to apply preprocessing procedures to the original documents in order to output a more simplified text [9]. There are three steps in this stage.

A- **Tokenization**: The main goal of tokenization is separate sentences into words.

B- **Stop Words Removal**: is the manner of removing words that appear many times in the text and don't offer the required information for recognizing an important sense of the document. There are many strategies utilized for indicating such stop words list. Now, various English stop word list is generally utilized to the TC procedure.

C-**Stemming**: is the method of generating origin of the word, in This research paper word stemming is done by using Porter’s stemming algorithm [10].

4.2. Feature Extraction

An essential aspect of TC is computing features score. The features include: Term Frequency and Inverse Document Frequency (TF-IDF), term frequency and thematic words.

1-TF-IDF: - identifying the term significance is very useful in TC system, that can be done by weighting term which can be calculated by multiplying TF by the IDF [2] Which can be calculated as in Equation (1).

$$F1 = TF * \log(\frac{N}{df}) \qquad (1)$$

Where TF= is the term frequency

N=Number of documents in the document collection.

df= is the number of documents in which the term appears

2- Term Frequency (TF): - which can be calculated using Equation (2).

$$F2 = \frac{freq_{term}}{\max\_freq} \qquad (2)$$

Where: Freq$_{term}$= is the number of terms repetitions and Max_freq= is the maximum term frequency.

3- Thematic Words (TW): - Are the most frequently used terms that exist in the document. The calculation of This feature is done by computing the repetition of all terms in the document, then only (K) terms with the maximum repetition is selected, in this paper. This feature is calculated by dividing the number of thematic words by the maximum number of thematic words in the document as expressed in Eq. (3) [11].

$$F3 = \frac{TW_i}{Tw\_max} \qquad (3)$$

Where

Tw$_i$= is the number of thematic words(i)

Tw_max= maximum thematic words in the document.

4.3.Multiple Linear Regression (MLR)

MLR is a statistical technique for formulating the correlation between the independent variables and a dependent variable, where there are two or more independent variables, but only one dependent variable [12]. MLR can be formulated as in Equation (4)

$$Y=W_0+W_1X_1+W_2X_2+\text{---}+X_m\ W_m \qquad (4)$$

Where

[Y] is the output vector (dependent variable).

[W] feature weight vector.

[X] The extracted features (independent variables).

We rewrite Equation (4) to be in the following form as in Equation (5).

$$C=W_0 +W_1F_1+W_2F_2+W_3F_3 \qquad (5)$$

Where

[C] Is the class number.

[W] Feature weight vector.

[F] The extracted features as explained in section 4.2

The regression model can be represented in a matrix form as follows.

$$\begin{bmatrix} C_1 \\ C_2 \\ Ci \\ \\ . \\ Cm \end{bmatrix} = \begin{bmatrix} F_{1,1}F_{1,2}F_{1,3} \\ F_{2,1}F_{2,2}F_{2,3} \\ .. \\ F_{p,1}F_{p,2}F_{p,3} \end{bmatrix} * \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}$$

Where p is the number of documents from the collected document data set. To estimate the weights for the extracted features we must train our model. Reuters 21578 dataset set used to train our model. A subset of the Reuters 21578 used, where six topics only selected for the proposed model. For each topic, 60 files were used for the training phase to learn the feature weights that used later in the classification phase of the proposed model. The three extracted features (F1, F2, F3) that were described in section 4.2 used as input to the model. The desired output C represents the class number for each document in the document set (D).

Our goal is to estimate the values of (W) which represent the weights of the selected features [13]. W will be calculated using Eq. (6).

$$W=(F.F^t)^{-1}.F^t\ C \tag{6}$$

Algorithm 1 shows the main step of MLR.

```
Algorithm (1): MLR
Input: Matrix F of size P*3 Documents features /*Where P number of
          documents, 3 numbers of  features*/
        Desired output C of size P*1
Output: vector Weight of size (3)
Step1: initialize vector weight W to zero.
Step2: For each document in the training set
Step3: $Z = F*F^t$
Step4: $B = Z^{-1}$
Step5: $A = F^t*C$
Step6: Weight=B*A
Step7:End
```

At the end of this algorithm a vector weight obtained which represent the weight for every feature.

4.4. Extreme Learning Machine (ELM)

ELM is suggested by Huang at (2004). The goals of ELM are to avoid time consuming that occurred in the most iterative training algorithms and improving the generalization performance. ELM is a feedforward neural network that consists of an input layer, hidden layer and output layer. The hidden layer may be a single layer or multi layers. ELM is a fully connected neural network, where the node in the input layer connected to every node in the hidden layer which in turn connected to every node in the output layer. The weights between the layers are set initially randomly, also the bias of the hidden layer is generating randomly. ELM could avoid falling into a local minimum. There are two phases for the ELM: training phase and testing phase. In the training phase a set of documents was used as input and the desired class output were set. At this phase the weights are adjusted to make the ELM suitable for the classification purpose [14]. Algorithm 2 shows the main steps of the ELM.

```
Algorithm 2: ELM
Input: feature Weight /* from algorithm 1*/
          Matrix F   /* which represent features that extracted from a set of
                         Documents that used for training*/
       Vector C    /* the desired output for each class*/
Output: vector W /*That represent weight ELM*/
Step1: randomly initialize W, W1 and bias b
Step2: X=weight*F
Repeat
Step3: hidden_output= sigmoid (b* (X*W))
Step3: out_layer=sigmoid (hidden_layer*W1)/* Sigmoid activation function
Step4: W1=$(W1*out\_layer)^{+}$ C /*where $^{+}$ is pseudoinverse */
Step5: W= $(W*hidden\ output)^{+}$
 Until (Max_iteration)
```

1.
2.

5.Reuters 21578 Dataset

Reuters 21578 is a benchmark dataset for document classification. There are 90 categories in the corpus. The archives in the Reuters-21578 gathering showed up on the Reuters newswire in 1987. The collection is divided into 22 files. Every one of the initial 21 files contains 1000 documents, while the last file contains 578 documents. The files are in SGML format. Each of the 22 files begins with a document type declaration line [15].

6.Experimental Results

There are two main purposes of the proposed TC model. The first purpose is to compute the weight of each selected features which indicates the importance of these features. **Figure 2.** shows the weights of each feature in our proposed method. From the results we can see the order of effective features weights as follows TF-IDF, TW and TF.

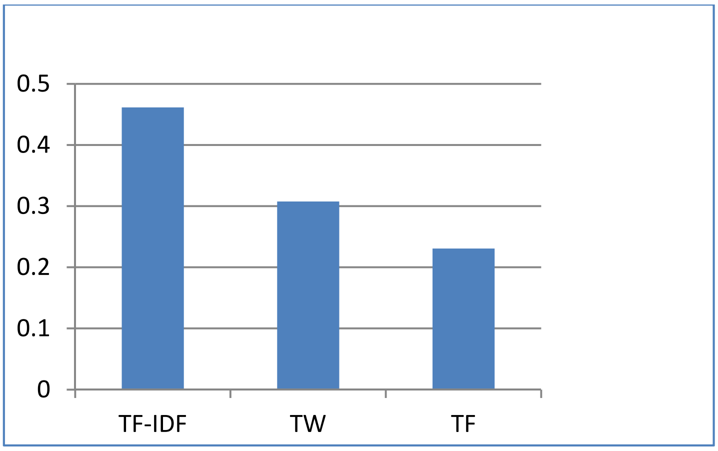

Figure 2. Features weights

The second purpose of the our proposal is to evaluate the TC model. As described in section 5 Reuters 21578 were used. Only six topics were selected with150 files for each topic so the total number would be 900 files, where 90 files for each topic used in the training phase and the remaining 60 files used during the testing phase. The accuracy was used to measure the performance of the proposed model.

$$Accuracy = \frac{N_{coorect}}{N} \qquad (7)$$

Where

$N_{correct}$ is the number of correct classified documents.

N is the total number of documents.

Table 1. shows the accuracy for WELM and ELM for the six selected topics.

Table 1. Classification Accuracy using ELM and WELM.

Topic	ELM Accuracy	WELM Accuracy
Earn	0.76	0.81
Money-fx	0.73	0.78
Grain	0.75	0.83
Crude	0.78	0.85
Trade	0.74	0.79
Interest	0.77	0.83

The results showed the competence of the WELM compared with ELM. As known the most important problem of the TC is the misclassification between classes. This problem occurs because the overlap of text features for one class with another class. The proposed model introduced the feature weights to increase the separation between classes to overcome this problem.

6. Conclusions

In this paper, TC is studied and empirically tested based on the MLR and EML. The MLR was applied to the three features extracted from each document text that produced feature weights that was used in turn with the features in the ELM to produce the WELM. The introducing of feature weights has improved the performance of the TC.

The experiment on the Reuter 21578 dataset has showed that the WELM is effective in TC. The results of a thorough experimental analysis obviously indicate that WEML offers a considerably successful performance in terms of accuracy in compared with ELM.

References

1. Zhang, X.; Wu, B. *Short Text Classification Based on Feature Extension Using The N-Gram Model International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*.**2015.**
2. AL Zaghoul, F.; Al-Dhaheri, S. *Arabic Text Classification Based on Features Reduction Using Artificial Neural Networks*. UKSim 15th International Conference on Computer Modelling and Simulation **.2013**.
3. Özgür, L.; Güngör,T. *Two-Stage Feature Selection for Text Classification Springer International Publishing Switzerland*. **2016**.
4. Jodha, R.; Sanjay, G; Chowdhary, K.; Mishra, A. *Text Classification using KNN with different Features Selection Methods*. International Journal of Research Publications(IJRP). **2018**, *8, 1.*
5. Mowafy, M.; Rezk, A.; El-bakry, H**.** *An Efficient Classification Model for Unstructured Text Document* American *Journal of Computer Science and Information Technology*. **2018**, *6,1.*
6. Renukadevi, D.; Sumathi, S. *Term Based Similarity Measure for text classification and clustering using fuzzy c mean algorithm. International Journal of Science*. Engineering and Technology Research(IJSETR) **.2014**, *3, 4.*
7. Conneau, A.; Holger, S.; Loïc, B; Yann Lecun *Very Deep Convolutional Networks for Text Classification.* In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics.**2017**, *1,* 107–1116.
8. Zheng, T.; Wang, L. *Unlabeled Text Classification optimization Algorithm Based on Active Self-Paced Learning*. IEEE International Conference on Big Data and Smart Computing. **2018**.
9. Mikolov, T.; Karafiat, M.; Burget,L.;Cernocky,J. ; Khudanpur, S.*Recurrent neural network based language Model*. In Proceeding of the Annual Conference of the International Speech Communication Association. **2010**,1045-1048.
10. Porter Stemming Algorithm: http://www.tartarus.org/martin/PorterStemmer.
11. John, A. Muti-Document Summarization System: Using Fuzzy Logic and Genetic Algotihm. *International Journal of Advanced Research in Engineering and Technology (IJARET)*. **2016**, *7, 1*, 30-40.
12. Chatterjee, s.; hadi, A. *Regression analysis by example*. A john Wiley & sons, Inc., PUBLICATION fourth edition. **2006.**
13. Timothy,Z. *Multiple Regression and Beyond: An Introduction to Multiple Regression and Structural Equation Modeling*. by Routledge, Second edition. **2015**.
14. Guang, B.; Qin,Z.; Chee-Kheong,S. *Extreme learning machine: Theory and applications*. ScienceDirect. **2006**.*70, 489-501.*
15. David D. available at: http://www.daviddlewis.com/resources/testcollections/reuters21578.