Performance Analysis of Various Text Classification Algorithms

2015312904, Dept. of Electronic and Electrical Engineering, Joon Woo Kwon

Abstract

Text Classification is one of the most important research issues in the field of text mining where it categorizes a given text document into a suitable categories. Text classification is used in various fields such as spam mail detection, news classification, automatic response, sentiment analysis, and Chatbot. In general, the text classification uses a machine learning algorithm, and among them algorithms suitable for text data such as Naïve Bayes and Support Vector Machines are known to show good performance. In this paper, we provide a brief review on various text classification algorithms presented through previous studies and analyze performance by implementing each algorithm with the same dataset.

Keyword: Natural Language Processing, Text classification, Machine Learning, KNN, NB, SVM, MLP

1. Introduction

Natural language is the language that we use in our daily lives. Natural language processing is the work of analyzing the meaning of these natural languages so that the computer can process them [3]. Natural language is not only lexically and grammatically unique in each language, but also the forms of expression are so diverse that it is often very difficult to define them as a set of rules to be determined [3, 4]. In addition, natural language has a characteristic that constantly change depending on the environment in which the language is used [3-5].

As deep learning has recently attracted big attention, artificial intelligence has emerged as an important keyword in the IT technologies. Natural language processing is the most important research area for artificial intelligence in that it understands human language to machine [4].

Among many problems of natural language processing, text classification is the most representative and most used problem in natural language processing dealing with natural language data. In this paper, we will provide a brief review on classifiers and compare the performance of various types of classifiers used for text classification by implementing them with the same natural language.

2. Text Classification Algorithms

Text classification refers to the division of a word, sentence or entire text into categories [5]. Like the image classification problem, we will work with supervised learning, which is to create a predictive model by learning machine learning algorithms in the state of being given training data and correct answer labels.

Among various algorithms for text classification, in this paper, we chose 4 classifier and those are KNN (k-nearest neighbor), NB (Naïve Bayes), SVM (Support Vector Machines) and MLP (Multi-Layer Perceptron).

2.1. K – Nearest Neighbor Classifier (KNN)

The K-nearest neighbor is a classification algorithm used in machine learning. It is used under the assumption that data with similar characteristics tends to fall into a similar category [6]. The class distribution in the training set is uneven. Some classes may have more samples than others. Therefore, the system performance is very sensitive to the choice of the parameter k [7]. If the value of k is large, the number of categories used for prediction increases, which results in slowing down classification and affects accuracy.

2.2. Naïve Bayes Classifier (NB)

Naïve Bayes classification is the simplest probabilistic classifier used to classify the text documents [1]. Based on Bayes' theorem, this calculates the posterior probability through the product of prior probability information and likelihood measured through observation. Naïve Bayes requires that features be independent of each other.

Naïve Bayes classifier is a fast, accurate and reliable algorithm. It has high accuracy and is fast for large amounts of data [7].

2.3. Support Vector Machines (SVM)

A Support Vector Machine (SVM) is a supervised classification algorithm that has been extensively and successfully used for text classification tasks [1]. This is an algorithm that defines a decision boundary, that is, a baseline for classification. So, when new unsorted points appear, it is possible to perform classification tasks by determining which side of the boundary belongs to.

2.4. Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron is a neural network consisting of more than two layers. Each layer has several perceptron [9]. Multi-Layer Perceptron adds an intermediate layer called Hidden Layer to obtain the effect of drawing multiple linear classification discrimination lines. A typical way of training MLP is backpropagation, whereby the term weights of a training document are loaded into the input units, and if a misclassification occurs the error is "back propagated" so as to change the parameters of the network and eliminate or minimize the error [1].

3. Experiments and Remarks

3.1 Performance Evaluation Methods

We implemented 4 chosen classifiers with the same dataset and evaluated them with several evaluation methods. Here is a brief review of the evaluation methods used in this paper.

1) Confusion Matrix

Confusion matrix is a matrix of prediction and actual labels [9].

2) Accuracy

Accuracy is ratio of correctly predicted samples to the whole samples [9].

3) Precision

Precision is ratio of actual true samples to the samples predicted as true [9].

4) Recall

Recall is ratio of samples predicted as true to the actually true samples [9].

5) F1-score

F1-score is harmonic mean between precision and recall [9].

6) Macro Averaging

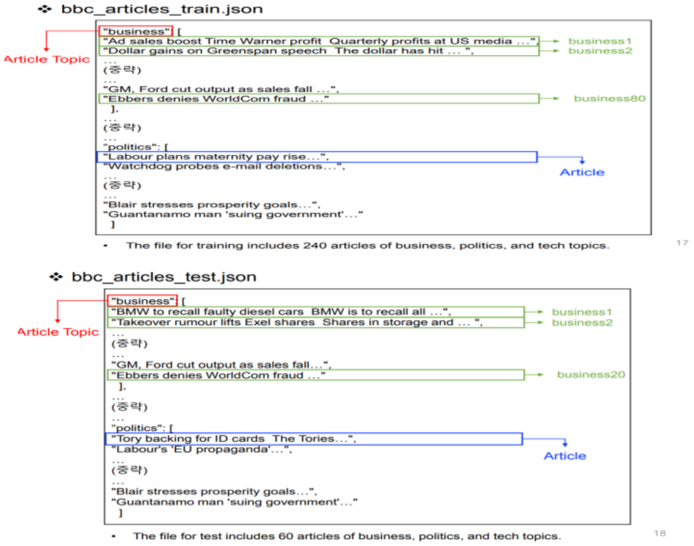
To calculate Macro averaging scores, we first calculate precision, recall and F1-score for each label respectively and then divide each score with the number of labels [9].

7) Micro Averaging

To calculate Micro-averaging scores, we first add up all confusion matrices into one confusion matrix, and then calculate precision, recall and F1-score using the confusion matrix [9].

3.2 Computer Simulation Results and Remarks

Computer simulation on 300 articles on business, politics, and tech topics from BBC news shown in Figure 1. has been performed to show the performance of the algorithm. Among 300 articles 240 articles were used to train the model and the rest 60 articles were used to test and verify the model. All simulations have been performed in Google Colab on a computer with the following configuration: Intel® Core™ i5-5200U CPU @ 2.20GHz and 8GB of memory with Microsoft Windows 8.1 K system. All codes were written in python language.



❖ bbc_articles_train.json

```
"business": [
"Ad sales boost Time Warner profit  Quarterly profits at US media ...",
"Dollar gains on Greenspan speech  The dollar has hit ... ",
...
(중략)
...
"GM, Ford cut output as sales fall ...",
"Ebbers denies WorldCom fraud ..."
 ],
...
(중략)
...
"politics": [
"Labour plans maternity pay rise...",
"Watchdog probes e-mail deletions...",
...
(중략)
...
"Blair stresses prosperity goals...",
"Guantanamo man 'suing government'..."
 ]
```

- The file for training includes 240 articles of business, politics, and tech topics.

❖ bbc_articles_test.json

```
"business": [
"BMW to recall faulty diesel cars  BMW is to recall all ...",
"Takeover rumour lifts Exel shares  Shares in storage and ... ",
...
(중략)
...
"GM, Ford cut output as sales fall...",
"Ebbers denies WorldCom fraud ..."
 ],
...
(중략)
...
"politics": [
"Tory backing for ID cards  The Tories...",
"Labour's 'EU propaganda'...",
...
(중략)
...
"Blair stresses prosperity goals...",
"Guantanamo man 'suing government'..."
 ]
```

- The file for test includes 60 articles of business, politics, and tech topics.

Figure 1 Description of total 300 articles from BBC news.

(Sequence from left to right: 240 articles for training the model, 60 articles for test)

Table 1 shows the parameter used in each classifier. Table 2 shows the confusion matrix for KNN, NB, SVM, and MLP classifier respectively. Table 3 shows the evaluation results of performance for KNN, NB, SVM, and MLP classifier respectively.

Table 1 The parameter setting to implement classifiers.

Classifier	Parameter

KNN	k = 2,
	weights = 'distance',
	metric = 'euclidean'

NB	alpha = 1
SVM	C = 1.0, max_iter = 1000

MLP	Hidden_layer_sizes = (100,0)
	Activation = 'tanh', Solver ='adam',
	batch_size ='auto',
	learning_rage_init=0.001, max_iter=200

Table 2 The confusion matrix for each classifier

Classifier	Confusion Matrix

KNN	20 0 0
	1 18 1
	0 0 20

NB	19 0 1
	1 19 0
	0 0 20

SVM	20 0 0
	2 17 1
	0 0 20

MLP	19 0 1
	2 17 1
	1 0 19

Table 3 The evaluation results for each classifier

	Accuracy	Precision (%)		Recall (%)		F1-score (%)	
		Macro	Micro	Macro	Micro	Macro	Micro
KNN	96.6666	96.8253	96.6666	96.6666	96.6666	96.6195	96.6666
NB	96.6666	96.746	96.6666	96.6666	96.6666	96.6656	96.6666
SVM	95.0	95.3823	95.0	95.0	95.0	94.8969	95.0
MLP	91.6666	92.2799	91.6666	91.6666	91.6666	91.6836	91.6666

From Table 2 and Table 3, we can see that all classifiers were successfully conducted and achieved high performance in that their evaluation results were all over 90% respectively. The classifier that achieved prediction the most precisely was KNN and NB. There were only very few differences between them.
Besides, MLP was the least precise classifier. It conducted 91.67% of accuracy and this figure is approximately 4~5% lower than the other.

4. Conclusions and Future Works

In this paper, we provided a brief review on several text classification algorithms as well as evaluation methods. To show the performances of the classifiers, we have set parameters for each classifier and provided computer simulation results performed on 240 articles for training and 60 articles for test and performance comparison on simulation results for 4 classifiers.
The results show that MLP algorithm showed relatively low performance. Future experiments that adjust parameters of MLP may improve our results further.

References

[1] B. S. Harish, D. S. Guru, and S. Manjunath, "Representation and Classification of Text Documents: A Brief Review", pp. 4-5, RTIPPR, 2010.

[2] P. R. Lim, and H. J. Kim, "A Tensor Space Model based Deep Neural Network for Automated Text Classification", SIGDB, 2018.

[3] Nohanryang, '텍스트 마이닝(Text Mining)이란?',2018. 09., https://bestpractice80.tistory.com/33

[4] ukairia777@gmail.com, '자연어 처리(natural language processing)란?', 2020.05., https://wikidocs.net/21667

[5] cdjs, '(번외편) 자연어 처리 - 텍스트 분류 & 문장 representation', 2019.07., https://cding.tistory.com/70

[6] 머신러닝, 'K-최근접 이웃(K-Nearest Neighbor) 쉽게 이해하기', 2019.12., http://hleecaster.com/ml-knn-concept/#:~:text=K%2D%EC%B5%9C%EA%B7%BC%EC%A0%91%20%EC%9D%B4%EC%9B%83(K%2DNearest%20Neighbor)%EC%9D%80,%EC%9E%88%EB%8B%A4%EB%8A%94%20%EA%B0%80%EC%A0%95%ED%95%98%EC%97%90%20%EC%82%AC%EC%9A%A9%ED%95%9C%EB%8B%A4.

[7] B. Li, S. Yu, and Q. Lu, "An improved k-Nearest Neighbor Algorithm for Text Categorization", Expert Systems with Applications, 2012.

[8] B. K. Shin, '머신러닝-1.나이브 베이즈 분류 (Naïve Bayes Classification)',2019.07.,https://bkshin.tistory.com/entry/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-1%EB%82%98%EC%9D%B4%EB%B8%8C-%EB%B2%A0%EC%9D%B4%EC%A6%88-%EB%B6%84%EB%A5%98-Naive-Bayes-Classification

[9] Y. J. Ko, 'Evaluation', pdf, SKKU, 2020, nlp.skku.edu