International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

© Global Society of Scientific Research and Researchers

http://ijcjournal.org/

Comparative Study for Text Document Classification Using Different Machine Learning Algorithms

Yin Min Tun[a]*, Phyu Hnin Myint[b]

[a,b]*University of Computer Studies, Mandalay (UCSM), Mandalay and 05013, Myanmar*
[a]*Email: yinmintun@ucsm.edu.mm*
[b]*Email: phyuhninmyint@ucsm.edu.mm*

Abstract

Classification is a supervised learning method: the goal is finding the labels of the unknown object. In the real world, the tedious amounts of manual works are required to label the unknown documents. The system is initially trained by labeled documents by using one of the supervise machine learning algorithm and then applied trained model to predict the label of the unknown documents. The framework of text document classification consists of: input text document, pre-processing, feature extraction and classification. The analysis four common classification methods are performed: Naïve Bayes, Decision Tree, Support Vector Machine and K-nearest neighbors for text document classification. The main focus of this paper is to present comparative study of different exiting classification methods for text document classification. The experiment performed different classification methods on the Enron Email Dataset and measure classification accuracy, true positive, true negative, false positive and false negative to compare the performance of different classification methods.

Keywords: Classification; Text Mining; Classification Methods; Enron Email Dataset.

1. Introduction

Text document classification is one of the important tasks in text mining and NLP. Text document classification sometimes borrowed concepts from Natural Language Processing (NLP) and Artificial Intelligent (AI). In this paper, the different types of classification methods are studies to analysis the classifier performance of each classification methods. The TF-IDF feature is extracted from the text documents and trained by different classification methods to perform the comparative study for classification methods. The classifier performance is measured on the structure of 5-fold cross validation according to the value of average classification accuracy (ACA), true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The automatic classification of texts into pre trained label has seen developing interest in the last 10 years, owing to the increased availability of documents in digital form and the subsequent need for their organization [2].

----

* Corresponding author.

Many Approaches have been proposed for text document classification since 80 century. Bayes formula was used to vector a document according to a probability distribution reflecting the probable classes that the document related to. Bayes formula gives a probability range in which the document can be assigned according to a predetermined set of classes for example twenty NewsGroups Dataset. SVM classifier used to classify the label of documents [1]. The fast text classifier fastTextis is proposed for a simple and efficient baseline for text classification [3]. An algorithm for learning from labeled and unlabeled documents based on the combination of Expectation-Maximization (EM) and a naive Bayes classifier are proposed. The algorithm first trains a classifier using the available classes of documents, and probabilistically labels the unknown documents. It then trains a new classifier using the labels for all the documents, and iterates to convergence. This basic EM procedure works well when the data conform to the generative assumptions of the model [4]. Bi-Normal Separation' (BNS) is proposed as the new feature for text classification. A new evaluation methodology is offered to choose one or pair of metrics for the best classifier performance [5]. Text document classification is presented in section II. Different classification methods are presented in section III. The description of Dataset and experiments are presented in section IV. And section V presented the conclusion.

2. Text Document Classification

Text classification assignes documents to one or more pre-defined catégories. Applications of text classification are :

- Organize web pages into hierarchies
- Domain-specific information extraction
- Sort email into different folders
- Find interests of user

The framework of the text document classification is shown in figure 1. The main steps of the text documents classification are:

- Pre processing
- Feature extraction
- Classification

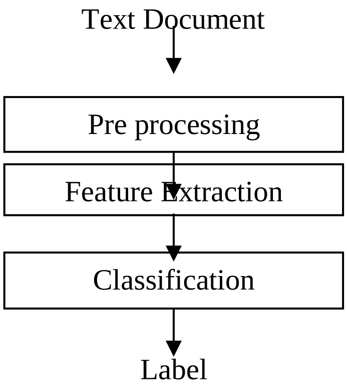

Figure 1: Framework of the text document classification

2.1. Pre processing

In text preprocessing, the documents are processed and formed into vector of term defined by a dictionary. The dictionary is created from the preprocessing of documents. The preprocessing tasks are: tokenization, stop words removing and stemming.

- Tokenization breaks up a sequence of strings into pieces such as words or keywords called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded.

- The useless words from tokens are defined as stop words. The stop word list is a list of commonly used word (such as “the”, “a”, “an”, “in”) that can’t effect the meaning and information of text documents. The stop words removal removes words that occur commonly across all the documents in the corpus.

- Stemming algorithms cuts off the end or the beginning of the word and takes into account a list of common prefixes and suffixes. This indiscriminate cutting can be successful in some occurrences, but remain some issues to solve. The stemming task reduces different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form.

2.2. Feature Extraction

The term frequency is the most common feature to describe the information of document in the form of vector. More frequent terms in a document are more important, i.e. more indicative of the topic.

f_{ij} = frequency of term i in document j

The term frequency is normalized by dividing by the frequency of the most common term in the document:

$tf_{ij} = f_{ij} / \max_i\{f_{ij}\}$

The idf is used to describe the terms that appear in many different documents are less indicative of overall topic.

df_i = document frequency of term i

= number of documents containing term i

idf_i = inverse document frequency of term i,

$$= \log_2 (N/ df_i) \qquad (1)$$

where N is total number of documents and Log used to dampen the effect relative to tf. A typical combined term importance indicator is tf-idf weighting:

$$wij = tfij\ idfi = tfij\ \log2\ (N/ dfi) \quad (2)$$

A term occurring frequently in the document but rarely in the rest of the collection is given high weight. A collection of n documents can be represented in the vector space model by a term-document matrix. An entry in the matrix corresponds to the “weight” of a term in the document; zero means the term has no significance in the document or it simply doesn’t exist in the document. The total number of documents (n) for terms (t) in dictionary is as follow:

Table 1: Structure of document vector

	T1	T2	T3	…..	Tt
D1	W11	W12	W13	…..	W1t
D2	W21	W22	W23	…..	W2t
:					
Dn	Wn1	Wn2	Wn3	…..	Wnt

2.3. Classification Methods

The text document assigns label for the unknown documents using pre trained classifier. The input of the classification is a description of an instance, x∈X, where X is the instance language or instance space. The output is a fixed set of categories C = {c_1, c_2,…, c_n}. Document classification is an example of Machine Learning (ML) in the form of Natural Language Processing (NLP). By classifying text, we are aiming to assign one or more classes or categories to a document, making it easier to manage and sort.

2.3.1. Naïve Bayes Classification

The main idea of Naïve Byes is the probability theory, which assigns to each sentence numerical degree of belief between *0* and *1* It provides a way of summarizing the uncertainty Naïve Bayes is a good strategy is to predict:

$$\arg\max_{Y} P(Y|X_1,\ldots,X_n) \quad (3)$$

The classifier used Bayes Rule:

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)} \quad (4)$$

Where P(X1,…Xn|Y) is the likelihood , P(Y) is Prior probability and P(X1,…,Xn) is the normalized constant.

2.3.2. Decision Tree Classifier

Decision Tree builds an accurate model for each class based on the set of attributes. The trained model is used to classify future data for which the class labels are unknown. The training time of Decision tree is relatively fast compared to other classification models. Its classification accuracy is similar and sometimes better accuracy compared to other classification methods. It is simple and easy to understand. It can be converted into simple and easy to understand classification rules.

A decision tree is created in two phases:

Tree Building Phase: Repeatedly partition the training data until all the examples in each partition belong to one class or the partition is sufficiently small

Tree Pruning Phase: Remove dependency on statistical noise or variation that may be particular only to the training set

2.3.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the well-known classification methods for text classification. SVM has the flexibility in choosing a similarity function SVM also has Sparseness of solution when dealing with large data sets Support vectors are used to specify the separating hyper plane SVM has ability to handle large feature spaces and its complexity does not depend on the dimensionality of the feature space. The problem of over fitting can be controlled by soft margin approach. It has nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution and feature Selection. SVM is related to statistical learning theory and it has kernel function to separate the data points.

2.3.4. K-nearest Neighbor (KNN)

K-nearest Neighbor (KNN) is an eager learning method and use explicit description of target function on the whole training set. KNN is instance-based learning in which learning means storing all training instances and classification means assigning target function to a new instance. KNN has Preserving efficiency and accuracy while introducing provable privacy to the system. Constructing k-nearest neighbour classifier over horizontally partitioned databases has the better classifier performance. In nearest-neighbor learning the target function may be either discrete-valued or real valued. Learning a discrete valued function:

$$f : \Re^d \rightarrow V_{(5)}$$

where V is the finite set {v_1... vn}. For discrete-valued, the k-NN returns the most common value among the k training examples nearest to x.

3. Experimental Results

In this section, the experimental results are presented for classifier performance. The experiment is performed on the Enron Email Dataset in the structure of 5-fold cross validation. The average classification accuracy and classifier performance are measure over 5-fold cross validation. For Decision Tree, the fine tree model is used with maximum number of splits 100 and value of surrogate decision splits is Off.

3.1. DBWorld E-mails Dataset

DBWorld Email dataset was collected 64 e-mails from DBWorld newsletter and used them to train different algorithms. It contains 64 e-mails are collected from DBWorld mailing list. They are classified in: 'announces of conferences' and 'everything else'[6].

3.2. Five-Fold Cross Validation

The dataset is sub grouped into 5 groups. The 4 groups are used as training and remaining one is used as testing. The validation is performed for five times. For each validation time, randomly choose 4 groups as training and remaining one is used as testing. The average classification accuracy (ACA), true positive (TP), true negative (TN), false positive (FP) and false negative (FN) are calculated over these five validations.

- *True Positive (TP):* A true positive test result is one that detects the condition when the condition is present.
- *True Negative (TN):* A true negative test result is one that does not detect the condition when the condition is absent.
- *False Positive (FP):* A false positive test result is one that detects the condition when the condition is absent.
- *False Negative (FN):* A false negative test result is one that does not detect the condition when the condition is present.

Table 2: Classifier performance on DbWorld Email Dataset over 5-Fold Cross ValiDation

Classification Methods	**5-Fold Cross Validation**				
	(%)				
	ACA	***TP***	***TN***	***FP***	***FN***
Naïve Bayes	82.8	85.7	14.3	20.7	79.3
Decision Tree	78.1	79.5	80.5	21.5	19.5
Support Vector Machine	**93.8**	94	93.5	6	6.5
KNN Classifier	85.9	86.5	86	13.5	14

The experimental result for different classificaiton methods on DBWorld E-mails Dataset is shown in table II. In this table, the kernel function for SVM classifier is Quadratic kernel function, the box constraint level is 1 and the kernel scale is auto. For KNN classifier, the preset is Cosine KNN, the number of Neighbors is 10, and the value of distance weight is equal. For Naïve Bayes classifier, kappa statistics is 0.6522 and the mean absolute error is 0.181. For Decision Tree, the fine tree model is used with maximum number of splits 100 and value of surrogate decision splits is Off. Among these four classifiers, the SVM has the highest classification accuracy because it uses quadratic kernel function. Sine quadratic kernel function successfully separates the two labels of

DBWorld E-mails Dataset. The classification time and prediction speed are also measure as shown in table 3.

Table 3: Classificaiton time and Prediciton Speed on DbWorld Email Dataset over 5-Fold Cross ValiDation

Classification Methods	***Training Time(secs)***	***Prediction Speed (obj/sec)***
Naïve Bayes	0.05	800
Decision Tree	5.492	370
Support Vector Machine	0.4195	800
KNN Classifier	0.40161	700

3.3. Results and Discussion

In this comparative study for text document classification, Support Vector Machine (SVM) has highest classification accuracy 93.8%. The classification time of Naïve Bayes classifier is shortest and Decision Tree has longest classification time over 5 seconds. According to our comparative study, the SVM classifier with term frequency is the best match for text document classification over 5-fold cross validation.

4. Conclusion

Text document is the essential and important in text mining and opinion mining. The IF-IDF feature can be applied in text classification methods. Among these classification methods the SVM has the highest classification accuracy over 5-fold cross validation. Other remaining classification methods also have acceptable classification accuracy and classifier performance. In future, convolutional neural network will be applied for text document classification with different types of text feature extraction methods.

References

[1] Isa, D., Lee, L.H., Kallimani, V.P. and Rajkumar, R., 2008. Text document preprocessing with the Bayes formula for classification using the support vector machine. IEEE Transactions on Knowledge and Data engineering, 20(9), pp.1264-1272.

[2] Sebastiani, F., 2002. Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1), pp.1-47.

[3] Joulin, A., Grave, E., Bojanowski, P. and Mikolov, T., 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[4] Nigam, K., McCallum, A.K., Thrun, S. and Mitchell, T., 2000. Text classification from labeled and unlabeled documents using EM. Machine learning, 39(2-3), pp.103-134.

[5] Forman, G., 2003. An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research, 3(Mar), pp.1289-1305.

[6] Filannino, M., 2011. DBWorld e-mail classification using a very small corpus. The University of Manchester.