Department of Mathematics and Computer Science

Thesis

Presented for obtaining the Master diploma in Computer Science
Specialty: Software Engineering and distributed system

By :

Sahnoune Chaouche Farouk
Houachemi Amira

Theme

Automatic Text Classification:a comparative study of three algorithms

Publicly supported on July 04, 2019

COLLEGE YEAR : 2018-2019

Dedication

I dedicate this work to my dear parents Fatma,Ahmad God protect them,
To my Brothers and Sisters (Malika,Rachid,Rodwan,Youcef,Samia,Saad,Zohyer)
To my Friends(Latif,Walid,Hamza,Islam,Rahim,Tarak,Bilal,Abd Rahman,Tahar,Ayoub)
To my Friends of Facebook(Halouma,Youssra,Asma,Mina,Hawa,Samira,Amira)

To my Friends of colleagues from 2018/2019(Younes,Ghilas,Zaki,Walid,Harachi,Abdo,Zino,Amar, Hadia,Hadjira,Radia)

Thank you so much for helping me,
both close or distant

Farouk.

Dedication

First and foremost, I would like to thank Allah Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible
Finally i must express my very profound gratitude to my parents ,my family and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through
the process of researching and writing this thesis. This accomplishment would not have been possible without them.
Thank you

Amira.

List of Figures

List of Tables

Acknowledgements

At first, I thank ***ALLAH*** *who helped me and gave me patience and courage during these years of study.*

I wish to express my sincere thanks to the people who helped me and contributed to the development of this message and to the success of this wonderful academic year.

First of all, I would like to thank the professors and faculty members of the Faculty of Science and Technology for the richness and quality of their education and for the modern training offered to their students.

I wish to express my sincere thanks to ***Dr. Mohamed Godjil*** *who, as Director of Memory, has always been attentive and available throughout the implementation of this work, as well as the inspiration and the assistance that he provided me throughout this memory.*

I do not forget my father and mother for their contribution, their support and their patience throughout my university life. Finally, I would like to thank all my friends and family who have always encouraged me to do so.

Thank you all.

Contents

List of Abbreviations

TC	Text Categorization
IR	Information Retrieval
ML	Machine Learning
TF-IDF	Term Frequency Inverse Document Frequency
TF	Term Frequency
NB	Naïve Bayes
SVM	Support Vector Machine
AI	Artificial Intelligence
NLP	Natural Language Processing
BOW	Bag Of Words
DR	Dimensionality Reduction
FS	Feature Selection
DF	Document Frequency
IG	Information Gain
CHI OR CH2	Chi-square
TS	Term Strength
CMU	Carnegie Mellon University
NLTK	Natural Language Tool kit
API	Application Program Interface
NG	NEWS GROUP'S

0.1 Introduction

With the rapid growth of the Internet, it has become natural that we handle a text not as printed but as online information. Today, we can search books and news electronically. Almost all companies and individuals have their own web pages and dispatch their information. When there is some information to find,it is usual to search for the information on the Internet. In this way, a lot of information has become open to the public.
If you have much information, you would need to classify it. When the scale of the target is small or the target is written by yourself, it is possible to classify it.However, in the current situation that many and ordinary people electronically post a lot of texts to the Internet, it is becoming impossible to classify them by hand. Then, it is necessary to classify text information automatically into various fields.
Manually organizing large document bases is extremely difficult, time consuming, error prone, expensive and is often not feasible. Automated text categorization is a viable option for larger organizations which has got time and money as the main constraints.
Text Categorization is the automatic classification of text documents under predefined categories or classes. Information Retrieval (IR) and Machine Learning (ML) techniques are used to assign keywords to the documents and classify them in to specific categories. Machine learning helps us to categorize the documents automatically. Information Retrieval helps us to represent the text as an attribute. The task of automated text categorization has witnessed a thriving significance since a decade both from the researchers as well as the developers .
In looking back upon the historical development of text classification technology, we find that the rule-base approach, i.e., the method of writing classification rules, was mainly used until the second half of the 1980s. This approach is simple but difficult to create rules by hand for high accuracy, and have to write many rules, as we change the domains. However, in the 1990, the performance of computers improved rapidly and it became possible to handle a great amount of text data. This led to the use of the machine learning approach, which creates classifiers automatically from the text data which were previously labeled with categories by hand. This approach became popular because of its high classification accuracy, reduction of labor, and conservative use of resources.

TC uses tools developed by IR researchers because it is a content-based document management task, sharing many characteristics with other IR tasks, such as text search, where the goal is to fi nd the set of documents (or document passes)most relevant to a particular query. For example, documents used in classification tasks are indexed using the same techniques as in IR; moreover, documents are compared and the similarity between them is measured using techniques originally developed for IR. The evaluation of classification tasks is often done using the same effectiveness measures as in IR
The applications of Text Categorization in the field of science and technology. It is also used in the fields of finance, sports and entertainment and medical sciences essay grading, and categorizing news paper.2 Most text classification and document

categorization systems can be deconstructed into the following four phases: Feature extraction, dimension reductions, classifier selection, and evaluations.so we are going to see

• Text pre-processing: In general, texts and documents are unstructured data sets. However, these unstructured text sequences must be converted into a structured feature space when using mathematical modelling as part of a classifier. First, the data needs to be cleaned to omit unnecessary characters and words. After the data has been cleaned, formal feature extraction methods can be applied. The common techniques of feature extractions are Term Frequency Inverse Document Frequency (TF-IDF), Term Frequency (TF), we categorize these methods as either word embedding or weighted word techniques and discuss the technical implementation details.

• Dimensionality Reduction: As text or document data sets often contain many unique words, data pre-processing steps can be lagged by high time and memory complexity. A common solution to this problem is simply using inexpensive algorithms. However, in some data sets, these kinds of cheap algorithms do not perform as well as expected. In order to avoid the decrease in performance, many researchers prefer to use dimensionality reduction to reduce the time and memory complexity for their applications. Using dimensionality reduction for pre-processing could be more efficient than developing inexpensive classifiers. we outline the most common techniques of dimensionality reduction, including like feature selection The aim of feature-selection methods is the reduction of the dimensionality of the dataset by removing features that are considered irrelevant for the classification and see algorithm methode The Naïve Bayes , k-nearest neighbor (KNN) ,Support Vector Machine (SVM) all of those are techniques which employs a discriminative classifier for document categorization.

• Performance evaluation can be done using training efficiency (how long it takes to learn the classified using the training data), classification efficiency (how long it takes to classify a document using the classier) and classification effectiveness (average correctness of the classier). In order to perform this evaluation, a set of pre-classified documents is split into a training and a test set, that are then used to train and to evaluate the classier, respectively.so on this chapter we are going to talk about our architecture,every step we explain with details and an example with every algorithm

• Also we will explain the experimental setting we presnt the steps to follow for the experiments performed , namely the classification methods, the evaluation metrics and the datasets that were used.and also we had see related work people ho with different algorithm in text classification

• finally we are going to talk about python and we presents the results using different methods and we have criticized the result obtained also we decision about what we found as result.

Chapter 1

Theoretical Back Ground

Introduction

Text classification is of great practical importance today given the massive volume of online text available. In recent years there has been an explosion of electronic text from the World Wide Web, electronic mail, corporate databases, chat rooms, and digital libraries. Moreover, in the case of text classification in particular, the texts must be processed before being used in training due to the particular unstructured format of this data.

1.1 Text Classification

1.1.1 Classification

Classification tasks in machine learning consist of automatically assigning the correct labels to some instances based on the study of previous examples. When applied to text documents, it consists of labeling new text documents with their category. Current approaches involve training a classifier on previous instances and constructing a model that captures the regularities in the examples of each category. This model is able to identify these regularities in new examples. types of classification:
• Binary classification:classify input objects into one of the two classes.
• Multi-class classification:classify input objects into one of the multiple classes.
Unlike a better understood problem of binary classification, which requires discerning between the two given classes, the multiclass classification is a more complex and less researched problem .[1]

1.1.2 Definition Of Text Classification

The text classification methods search for the best set of rules or functions that describe a predetermined set of document categories to predicate the categories of documents with unknown category.[2][3]. can be used to organize, structure, and categorize pretty much anything. The derived classification methods which called classifiers

are constructed based on the analysis of a set of documents whose category labels are previously known, such a set is called the training set[4]. For example, new articles can be organized by topics, support tickets can be organized by urgency, chat conversations can be organized by language, brand mentions can be organized by sentiment..
Some of the popular areas where text classification is applied are as follows [5] :
▶ Classify news as Politics, Sports, World, Business, Lifestyle
▶ Classify email as Spam, Other.
▶ Classify Research papers by conference type.
▶ Classify movie reviews as good, bad and neutral.
▶ Classify jokes as Funny, Not Funny.
For a classifier to learn how to classify the documents, it needs some kind of ground truth. For this purpose, the input objects are divided into training and testing data. Training data sets are those where the documents are already labeled. Testing data sets
are those where the documents are unlabeled. The goal is to learn the knowledge from the already labeled training data and apply this on the testing data and predict the class label for the test data set accurately.
The machine learning approaches to text classification is a general inductive process that is done by teaching the system how to classify new documents into pre defined categories. The learning process is done by using labeled samples (per classified text documents), which is called supervised learning, while unsupervised learning uses unlabeled samples

1.1.3 Single Label vs Multi-Label

Depending on the application, documents can be classified into one or more classes. For instance, a piece of news regarding how the Prime Minister spent his vacations may be classified both as politics and in the social column. This kind of classification is called *multi* label classification. where any number $0 < n_j \leq | C |$ of classes may be assigned to each document $d_j \in D$.

1.1.4 Supervised Vs Unsupervised Learning

Data and Knowledge Mining is learning from data. In this context, data are allowed to speak for themselves and no prior assumptions are made. This learning from data comes in two ways, supervised learning and unsupervised learning.

a.Supervised

Supervised learning, in the context of artificial intelligence (AI) and machine learning, is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing.

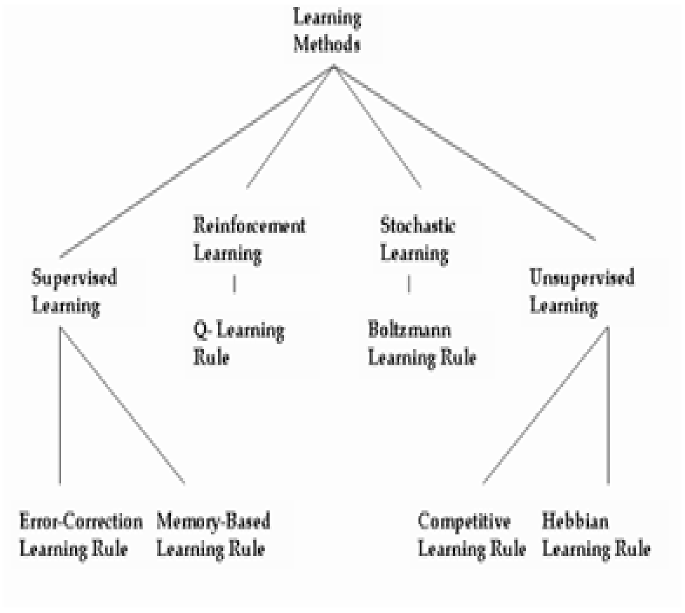


Figure 1.1: learning methods

b.Unsupervised

Self-Organizing neural networks learn using unsupervised learning algorithm to identify hidden patterns in unlabeled input data. This unsupervised refers to the ability to learn and organize information without providing an error signal to evaluate the potential solution. The lack of direction for the learning algorithm in unsupervised learning can sometime be advantageous, since it lets the algorithm to look back for patterns that have not been previously considered[6]

	Supervised learning	Unsupervised learning
Input Data	Uses Known and Labeled Data as input	Uses Unknown Data as input
Computational Complexity	Very Complex	Less Computational Complexity
Real Time	Uses off-line analysis	Uses Real Time Analysis of Data
Number of Classes	Number of Classes are known	Number of Classes are not known
Accuracy of Results	Accurate and Reliable Results	Moderate Accurate and Reliable Results

Table 1.1: comparing between Supervised and Unsupervised learning

1.1.5 Using Unlabeled Data

Traditional classifiers use only labeled data (feature / label pairs) to train. Labeled instances, however, are often difficult, expensive, or time-consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile, unlabeled data may be relatively easy to collect, but there have been few ways to use them. Methods dealing with this issue are divided into two categories: semi-supervised and active learning methods Semi-supervised learning is a task that lies somewhere between supervised and unsupervised learning. We have a few labeled samples, and we try to use the remaining unlabeled samples to get better performance at the learning task when compared to using only the labeled samples for supervised learning or all the samples in unsupervised learning.
Active learning (machine learning) is a specialized version of semi-supervised learning. Here, the author uses the labeled samples, and among the unlabeled samples, we try to find out labeling which small number of them will get much better performance. It is an iterative algorithm where the learner requests the user to label a few samples in each iteration, and this is repeated till convergence. This way, only a small amount of unlabeled data needs to be labeled to get very good performance.

1.2 Application Of Text Classification :

◇ Tagging content or products using categories as a way to improve browsing or to identify related content on your website. Platforms such as E-commerce, news agencies, content curators, blogs, directories, and likes can use automated technologies to classify and tag content and products.
◇ email routing, sending an email sent to a general address to a specific address or mailbox depending on topic[7].
◇ genre classification, automatically determining the genre of a text[8]
◇ sentiment analysis, determining the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document.
◇ As marketing is becoming more targeted everyday, automated classification of users into cohorts can make marketer's life simple. Marketers can monitor and classify users based on how they talk about a product or brand online. The classifier can be trained to identify promoters or detractors. Thus, making brands to serve the cohorts better

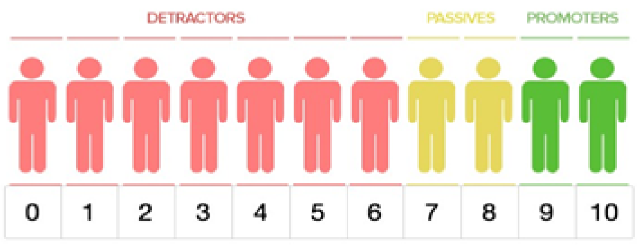
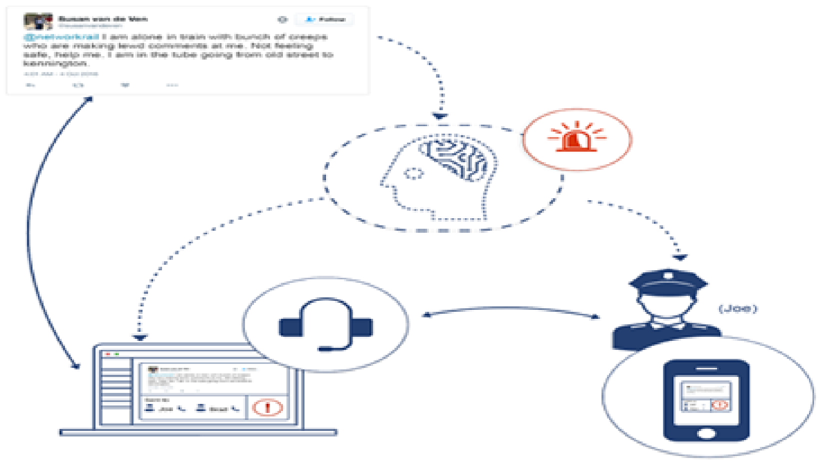


Figure 1.2: automated classification of users into cohorts

◇ A faster emergency response system can be made by classifying panic conversation on social media. Authorities can monitor and classify emergency situation to make a quick response if any such situation arises. This is a case of very selective classification. You can check out this study to read about an elaborated post on one such emergency response system.


Susan van de Ven

@networkrail I am alone in train with bunch of creeps who are making lewd comments at me. Not feeling safe, help me. I am in the tube going from old street to kennington.

Figure 1.3: faster emergency response system

Treatment

1.3 Text Pre-processing

Document $pre - processing$ is an essential step in text classification . The aim of the prepossessing stage is to transform text documents into a suitable format for automatic processing that the way that documents and queries are represented influences the quality of the results that can be achieved. there are several proposals that aim at improving retrieval result

1.3.1 Tokenization

Documents are tokenize by removing punctuation marks, digits, and numbers.The remaining character strings are considered as terms or tokens. The main goal of this step is the investigation of the words in a sentence for example

"After sleeping for four hours, he decided to sleep for another four" In this case, the tokens are as follows:

“After” “sleeping” “for” “four” “hours” “he” “decided” “to” “sleep” “for” “another” “four” .

1.3.2 Stop Word Removal

Text and document classification includes many words which do not contain important significance to be used in classification algorithms Words such as prepositions and articles that occur frequently and do not help in discrimination between classes are language dependent e.g. English stop words [9] such as
“a”, “about”, “above”, “across”, “after”, “afterwards”, “again”,. "the, other, or, in, there, etc. . . "

1.3.3 Stemming

In $NaturalLanguageProcessing(nlp)$, one word could appear in different forms (singular and plural noun form) while the semantic meaning of each form is the same [10].Stemming is the process of reducing words to their stem or root form where morphological information is used to match different variants of words.and One method for consolidating different forms of a word into the same feature space Text stemming modified words to obtain variant word forms using different linguistic processes such as affixçation (addition of affixes)[11][12]. For example, the stem of the word studding is study.

1.3.4 Lemmatization

Lemmatization is the process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errorscite[13]
In simple words, I would explain lemmatization as returning different forms of a single word to its root form. Given that the both examples that I gave are nouns, do not be confused that it is only applicable to nouns. Lemmatization works the same way for adjectives, action verbs, all the same. Such as:citeref14
Constructing - (Lemmatization) $\longrightarrow$ Construct
Extracts – (Lemmatization) $\longrightarrow$ Extract
Singing – (Lemmatization) $\longrightarrow$ Sing

1.3.5 Rare Words Removal

Low-frequency words are not helpful for discrimination between text documents because it is hard for text classification models to learn such terms . and words that are very unique in nature like names, brands, product names, and some of the noise charactersThis is considered as "rare word removal".[15]

1.4 Text Representation

In automatic text classification, it has been proved that the term is the best unit for text representation and classification [16]. Though a text document expresses vast range of information, unfortunately, it lacks the imposed structure of traditional database. Therefore, unstructured data, particularly free running text data has to be transformed into a structured data. To do this, many preprocessing techniques are proposed in lite rature[17][18].
After extracting the distinct words for text documents in the pre-processing stage,the next step is to represent the selected features in a suitable format for automatic processing. The most common for document representation in the TC task is called bag of words "BOW". Also, there is another approach based on phrases that can also be used for representing text documents

1.4.1 Bag Of Words Representation

Bag of words or also known as Vector space Model is the most popular and simplest way of text representation Term or feature in BOW representation is a single word. Each text document is represented as a vector of weights d_j = < w1$_j$, w2$_j$, w3$_j$,. . . ,wn$_j$ > where N denotes the number of distinct features (terms) in the document collection.

Given a collection of documents, its feature vectors are represented by a word by document matrix A, where each entry represents the weight of a word in a document, i.e. A = (a_{ik})
Where a_{ik} is the weight of word i in the document k, since every word does not normally appear in each document, the matrix A is usually sparse. The number of rows, M, of the matrix corresponds to the number of words in the super vector W. M can be very large.
There are several ways of determining the weight aik of word i in document k, but most of the approaches are based on two empirical observations regarding text
• The more times a word occurs in a document, the more relevant it is to the topic of the document.
• The more times the word occurs throughout all documents in the collection,the more poorly it discriminates between documents.
Let f_{ik} be the frequency of word i in document k, N the number of documents in the collection, M the number of words in the collection after stop word removal and word stemming, and n_i the total number of times word i occurs in the whole collection. Traditional methods for term weighting are used to determine the most suitable one for Arabic text categorization task. Here is an example of Bow

Document :
" As the home to $UVA's$ recognized undergraduate and graduate degree programs in systems engineering. In the UVA Department of Systems and Information Engineering, our students are exposed to a wide range of range "
Bag-of-Words(BoW):
" As ", " the ", " home ", " to ", " $UVA's$ ", " recognized ", " undergraduate ", " and ", " graduate ", " degree "," program " " in ", " systems ", " engineering ", " in ", " Department ", " Information "," students ", " "," are ", " exposed ", " wide ", " range "
Bag-of-Feature (BoF):
Feature = 1,1,1,3,2,1,2,1,2,3,1,1,1,2,1,1,1,1,1,1

Boolean Weight

Boolean weighting is the simplest way for weighting the terms. If a term t occurs in a document dj at least once then its weight is 1 other wise 0

Term Frequency Weight

Term weighting is a procedure that takes place during the text indexing process in order to assess the value of each term to the document. Term weighting is the assignment of numerical values to terms that represent their importance in a document in order to improve retrieval effectiveness [19].
Essentially it considers the relative importance of individual words in an information retrieval system, which can improve system effectiveness, since not all the terms in a given document collection are of equal importance. Weighing the terms is the means

that enables the retrieval system to determine the importance of a given term in a certain document or a query.
It is a crucial component of any information retrieval system, a component that has shown great potential for improving the retrieval effectiveness of an information retrieval system [20].

Term Frequency Inverse Document

$TFIDF$ can be considered as a statistical weighting scheme Document frequency of a term $DF(t_j)$ is the number of text documents in the corpus in which t_j occurs at least once and the inverse document frequency IDF of the term t_j
$K.SparckJones$ [21] proposed Inverse Document Frequency (IDF) as a method to be used in conjunction with term frequency in order to lessen the effect of implicitly common words in the corpus

$$\mathrm{IDF}\left(t_i\right) = \log \frac{D}{\mathrm{DF}\left(t_i\right)} \tag{1.1}$$

Where D is the total number of documents in the *dataset*. The weight of term ti in a document d_j using $TF.IDF$ is defined as:

$$\left(t_i, d_j\right) = TF\left(t_i, d_j\right) * IDF\left(t_i\right) \tag{1.2}$$

Document Length Normalization

Naturally, long documents contain more terms than short documents.Considering that the similarity between documents can be measured by how many terms they have in common, long documents will have more terms in common with other documents than short documents, so they will be more similar to other documents than short documents. To contrast this tendency, weights of terms for TC tasks are usually normalized so that document vectors have unitary length.

1.5 Dimensionality Reduction :

In text classification tasks, the documents or examples are represented by thousands of tokens, which make the classification problem very hard for many classifiers. *Dimensionality* reduction is a typical step in text mining, which transform the data representation into a shorter, more compact, and more predictive one [22]

Researchers have developed many dimension reduction techniques. the main objective of these techniques is to reduce the dimension of feature space by eliminating noisy, irrelevant, and non-informative data while retaining relevant and informative ones.Generally, dimension reduction approaches are classified into two classes: feature extraction and feature selection.

There are two distinct ways of viewing DR in terms of the nature of the resulting terms:

⊙ DR by term selection: where the final set of terms representing documents is a subset of the original set of terms, it is the most popular method for DR.
⊙ DR by term extraction: where the final set of terms representing documents is not of the same type of the original set of terms (e.g., if the original terms are words, the terms used to represent documents may not be words at all), but are obtained by combinations or transformations of the original ones

1.5.1 Feature Selection

Feature selection is a way of finding the best possible feature set. The main idea of feature selection is to choose a subset of input variables by eliminating the noisy or redundant features and keeping the quality patterns.This means that processing and classifying the entire feature set takes such a large amount of time that it is hard to work with. Feature selection can significantly improve the effectiveness and robustness of the resulting classifier or regression models. More importantly, feature selection can help to discover the features that are really important in the classification.

Feature selection algorithms can be divided into two types: wrapper and filter, based on whether or not feature selection is done independently of categorization model

1. filter : Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the data set. The methods are often uni-variate and consider the feature independently, or with regard to the dependent variable.

2. Wrapper : Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model us used to evaluate a combination of features and assign a score based on model accuracy.
   The wrapper algorithms consider the performance of a particular learning algorithm while an optimal subset of feature terms is selected. They measure the goodness of feature term set by the performance of the algorithm. Their results are dependent on the algorithm.

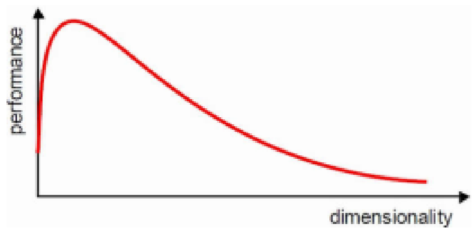


Figure 6 Relation between the dimension of feature space and the performance of the classification model

Feature Selection Process

Feature selection process requires two main steps: search strategy and objective function :

1. Search strategy: The aim of this step is to generate subsets of features for evaluation by an objective (evaluation) function. The behavior of the search process usually depends on the initial start point of the search.

2. Objective function: In *FS* process, an Objective function is used to measure the goodness of subsets of features generated by a search algorithm.

Feature Selection Approaches

There are two quite distinct ways of viewing DR, depending on whether the task is approached locally (i.e. for each individual category, in isolation of the others) or globally. in the local approach, For each category ci , features are chosen in terms of which the classifier for category ci will operate. In the global Approach, features are chosen in terms of which the classifier for all categories C = c1 , ...,cc will operate

1. Document Frequency (DF): Document frequency DF uses the number of documents in which each word appears as the informative score for the term. If the ratio of DF to the total number of documents in the collection of a word is too close to 1 then this word has a uniform distribution over the collection. It means that the term most probably appears in almost all documents and it is not a good feature to discriminate the topic. Moreover, the words with too small values for DF appear in only a few documents and, even if in the same class, they cannot be discrimination features for that class. So we can say Simply measures in how many documents the word appears. Since it can be computed without class labels, it may be computed over the entire test set as well. Selecting frequent words will improve the chances that the features will be present in future test cases. It is defined as

$$DF = \sum_{i=1}^{m}(A) \tag{1.3}$$

1. Information Gain (IG) : IG scores show the contribution ratio of the presence or absence of a term to correct classification of text documents IG assigns a maximum value to a term if it is a good indicator for assigning the document to any class. As it is indicated below, IG is a global feature selection metric as producing only one score for any term t and this score is calculated using :

$$\begin{aligned} IG(t) = &- \sum_{i=1}^{m} P\left(C_i\right) \log P\left(C_i\right) \\ &+ p(t) \sum_{i=1}^{m} P\left(C_i|t\right) \log P\left(C_i|t\right) + p(\neg t) \sum_{i=1}^{m} P\left(C_i|-t\right) \log P\left(C_i|\neg t\right) \end{aligned} \tag{1.4}$$

where M is the number of classes, P(C_i) is the probability of class C_i, P(t) and P(t) an empirically determined number. are the probabilities of presence and absence of term t, P(c_i/t) and P(c/t) are the conditional probabilities of class C_i give presence and absence of term t, respectively.

1. Chi-square Statistic (CHI) : Chi-square method is used to determine the degree of dependency between a term t and a document class c. Base on the two way contingency table, suppose we have a term t and a document class c, A denotes the number of times that t appears in c, B denotes the number of times that t happens in all categories except c. C refers to the number of times c happens without t, D represents the number of times that c and t do not exist, and N is the number of all documents, then Chi-square is determined as

$$\begin{gathered} \text{Chi}(t,c) = \tfrac{N \times (AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \\ N = A + B + C + D \end{gathered} \tag{1.5}$$

If X 2 (t, c) is equal zero that means the term t and the category c are independent. One of the ways to determine the term goodness of t by calculating the chi-square of that term with each of all given categories and then picking the maximum score to be the chi-square value of t. Term Strength (TS) : Term strength is used for feature reduction in the text classification task. TS estimates the importance of a term based on that term appearing in pair related documents. It calculates the probability that a term occurs in a pair of documents. Let d1 and d2 be related documents, and t is a term then the TS of the term t can be estimated

$$TS(t) = P\left(t \in d_1 | t \in d_2\right) \tag{1.6}$$

1.6 Classification Method :

In order to prove the efficacy of the proposed method, it was necessary to employ the classifiers commonly used for text classification research in the literature and proven to be significantly success full The most familiar text categorization algorithms will be presents in the following sections. There are two different ways to build a classifier:
○ Parametric: According to this approach, training data are used to estimate parameters of a distribution or discrimination function on the training set. The main example of this approach is the probabilistic Naive Bayes classifier.
○ Non-parametric: These classifiers base classification on the training set itself. This approach may be further subdivided into two categories: A wide variety of techniques has been designed for text classification. In this section, we will show in a brief some techniques that are used for automatic text classification

1.6.1 Vector Method :

Historically, the Vector method was one of the first methods to be applied in Information retrieval tasks. In the Vector method, documents and queries are represented as a set of weighted index terms,that is, vectors in a p-dimensional space, where p is the total number of index terms.
Based on the weights of its terms, each document can be ranked by decreasing order of similarity to the query. The similarity of each document d to the query q is computed as the cosine of the angle formed by the vectors that represent each of them

$$\operatorname{sim}\left(\vec{d_i}, \vec{q}'\right) = \frac{\vec{d_j} \cdot \vec{q}}{\left\|\vec{d_j}\right\| \times \|\vec{q}\|} \tag{1.7}$$

1.6.2 K-Nearest Neighbors :

The k-nearest Neighbors algorithm (KNN) is a non-parametric technique used for classification. This method is used for text classification applications in many research domains[23]. The initial application of k-Nearest Neighbors ($k - NN$) to text categorization was reported by *Masand* and *colleagues*.The basic idea is to determine the category of a given query based not only on the document that is nearest to it in the document space, but on the categories of the k documents that are nearest to it. Having this in mind, the Vector method can be viewed as an instance on the k-NN method, where $k = 1$.

The best choice of k depends on the data; generally, larger values of k reduce the effect of noise on the classification but make boundaries between classes less distinct. Figure 1.4 shows that the test sample (circle) should be classified either to the first class of squares or to the second class of triangles. If k = 3 it is classified to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 it is classified to first class (3 squares vs. 2 triangles inside the outer circle). A good k can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when $k = 1$) is called the nearest neighbor algorithm

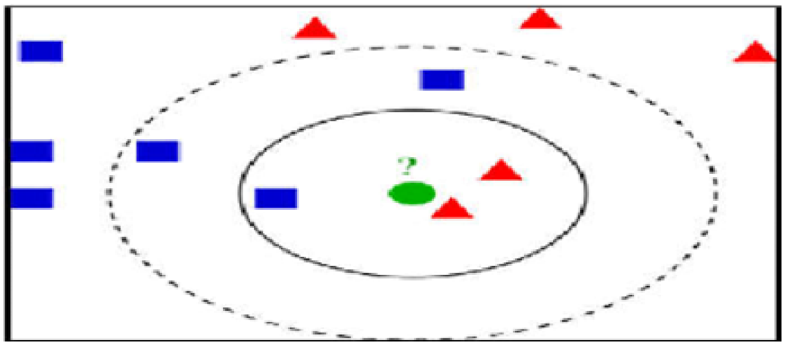


Figure 1.4: Example of KNN classification

1.6.3 Support Vector Machine SVM

Support vector machines (SVM) are a popular class of supervised learning algorithms. SVM was developed by *Vapnik* and *Chervonenkis*[24] in 1963. *B.E.Boser* et al. [25] adapted this version into a nonlinear formulation in the early 1990. SVM was originally designed for binary classification tasks. However, many researchers work on multi-class problems using this dominate technique [26]. The Figure 1.5 indicates the linear and non-linear classifier which is used for 2 Svm is particularly applicable to large and high-dimensional classification problems. It is based on the structural risk minimization principle where input points in N-dimensional space are mapped into a higher dimensional space and then a maximal separating hyper plane is found Consider a training set of labeled instances $x_t \in R^n$ $i = 1 \ldots L$, belong to a set of categories Figure 1.6 is an example of an optimal hyper plane for separating two classes From Figure 1.6, SVM builds the classification model on the training data using a linear separating function to classify unseen instances .For linearly separable vectors, the kernel function is simple. It takes the form:

$$f(x) = W \cdot X + b \qquad (1.8)$$

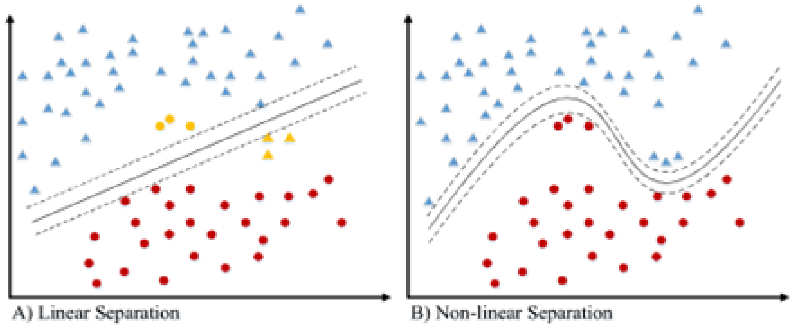


Figure 1.5: figure shows the linear and non-linear Support Vector Machine

W is called weight vector for optimal hyper-plane and b is known as the bias. The class of X (test instance) can be found using the following linear decision function

$$y = sign(f(x)) \tag{1.9}$$

Support Vector Machines have been applied successfully in many text classification

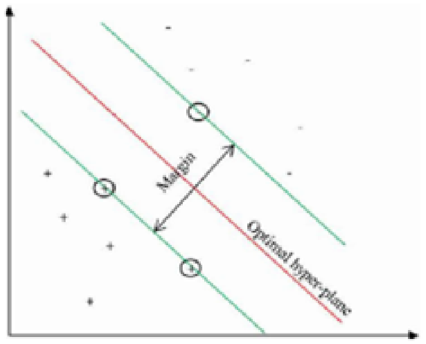


Figure 1.6: the linear separation between two classes (points marked with circles are support vectors)

tasks since most text classification problems are linearly separable and $SVMs$ are robust in high dimensional space and robust with sparse data.

1.6.4 Naive Bayes (NB) :

Bayesian is one of the most well known techniques of categorization. It is used to predict the class membership probabilities i.e. probability of a given record belongs to a particular category which is based on Bayes Theorem. Bayes theorem is a simple mathematical formula used for calculating conditional[33] probabilities Let us study about Bayes Theorem using a small example. X is a sample data record whose category is not known and H is some assumption. Let sample X belongs to a specified category C. If one needs to determine P(H | X) the probability that the assumption H holds given the data sample X Bayes Theorem is given as:

$$P(H|X) = \frac{P(X|H) * P(H)}{P(X)} \tag{1.10}$$

Where P (H | X) is the posterior probability of H on X. Posterior probability is based on information such as background knowledge rather than the prior probability which is independent of data sample X.
In the same way, P (X | H) is the posterior probability of X on H. If the given date is huge data sample, it would be difficult to calculate above probabilities. Conditional in dependency was introduced to overcome this limitation.
so we can say Naive Bayes classification is one of the simplest probabilistic Bayesian classification. It is based on an assumption that the effect of an attribute value on a given category is independent of the values of other attributes which is called as conditional independence. It is used to simplify complex computations [34]. The Naive Bayes classifier is a probabilistic classifier which is based on the Naïve bayes assumption. From Bayes rule, the posterior probability can be given as

$$P(H|X) = \frac{P(c|x) * P(c)}{P(x)} \tag{1.11}$$

Where x is a feature vector and $x = (x_1, \ldots, x_n)$ and c is category.Assume that the category cmax yields to the maximum value for P (c | x).The parameter P(c) is estimated as

$$P(c) = \frac{Numbreofdocumentsinc}{Nombreofducments} \tag{1.12}$$

The categorization results are not affected because parameter p(x) is independent of categories.Assuming that the components of feature vectors are statistically independent of each other, p (x | c) can be calculated as

$$\mathrm{p(x|c)} = \prod_{\mathrm{i}} \mathrm{p\,(x_i|c)} \tag{1.13}$$

If the maximum estimation is used then

$$P(x_i|c) = \frac{x_i, c}{N(c)} \tag{1.14}$$

Where N(x, c) is the joint frequency of x and c,

$$\mathrm{N(c)} = \sum_{\mathrm{x}} \mathrm{N(x, c)} \tag{1.15}$$

If some data x_i disappears in the training data, the probability of any instance containing x_i becomes zero, without considering the other features in the vector. Therefore to avoid zero probability, using Laplacian prior probabilities, p(x_i | c) is estimated as follows

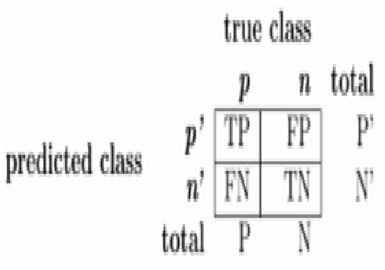
$$\mathrm{p\,(x_i|c)} = \frac{\mathrm{N(x_i,c)}+\lambda}{\mathrm{N(c)}+\lambda|\mathrm{V}|} \tag{1.16}$$

Where λ is a positive constant and is chosen as 1.0 or 0.5, and $|V|$ denotes the number of features The Naive Bayes classifier predicts the category c_{max} with the largest posterior probability [35]: C_{max} = arg_{maxc} P (c | x)
= arg_{maxc} P (c) p (x | c)

1.7 Performance Evaluation :

Evaluating the performance of computational systems is often done in terms of the resources (time and space) they need to operate, assuming that they perform the task that they are supposed to Text Classification systems are supposed to classify a query document, by as associating with it an ordered list of categories to which the query belongs. Obviously, it is not enough to classify a document as belonging to any set of categories in a reasonable amount of time. The categories should also be the " right " ones,that is, the ones that the document in fact belongs to. Measures based on Precision and Recall, which take into account if the predicted categories are the right ones, like F1 , have been widely used to compare the performance of TC methods.

1.7.1 Confusion Matrix:



		true class		
		p	*n*	total
predicted class	*p'*	TP	FP	P'
	n'	FN	TN	N'
	total	P	N	

Figure 1.7: shows a contingency table (or confusion matrix)

figure (1.7) shows a contingency table (or confusion matrix) illustrating the outcomes for a binary classification problem where one class is known as positive and the other as negative. If the outcome of a prediction is positive and the actual label is also positive, then it is called a true positive (TP); however, if the actual label is negative then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when negative examples are correctly labeled as negative, and false negative (FN) is when positive examples are incorrectly labeled as negative.

Precision

is defined as the fraction of the retrieved-documents that are relevant, and can be viewed as a measure of the system's soundness, that is:

$$precision = \frac{\text{\# Relevant Retrieved Documents}}{\text{\# Retrieved Documents}} \tag{1.17}$$

Recall :

Recall is defined as the fraction of the relevant documents that is actually retrieved, and can be viewed as a measure of the system's completeness, that is

$$Recall = \frac{\text{\# Relevant Retrieved Documents}}{\text{\# Relevant Documents}} \tag{1.18}$$

1.7.2 F1 Measure:

A good classifier should balance between precision and recall. It is usually beneficial to have a single measure assessing the system performance. One of the most used measures for the overall performance is $F1$ measure which is defined as the harmonic mean of precision and recall, The F1-measure is the harmonic mean of the two values

$$\mathrm{F1} = \frac{2 \cdot \text{ precision } \cdot \text{ recall}}{\text{precision } + \text{ recall}} \tag{1.19}$$

As Berger Et AL (Berger, Caruana et al. 2000) already pointed out, these measures are not adequate for evaluating tasks where a single answer is required, in particular, because Recall does not make sense when considering only a single answer. The point is that, if a document belongs to a class, it can be correctly classified or not, which means that the contingency tables with False Positives and False Negatives used to calculate Precision and Recall are not adequate. So, to evaluate single label TC tasks, these measures are not adequate

1.7.3 Accuracy

Accuracy, which is defined as the percentage of correctly classifier documents, is generally used to evaluate single-label TC tasks.Usually, Accuracy is represented as a real value between 0 and 1 it can be shown that in single-label classification tasks, Accuracy = microaveraged F1 = microaveraged Precision = microaveraged Recall This is because each document can be correctly classified or not, so the number of False Positives is the same as the number of False Negatives.

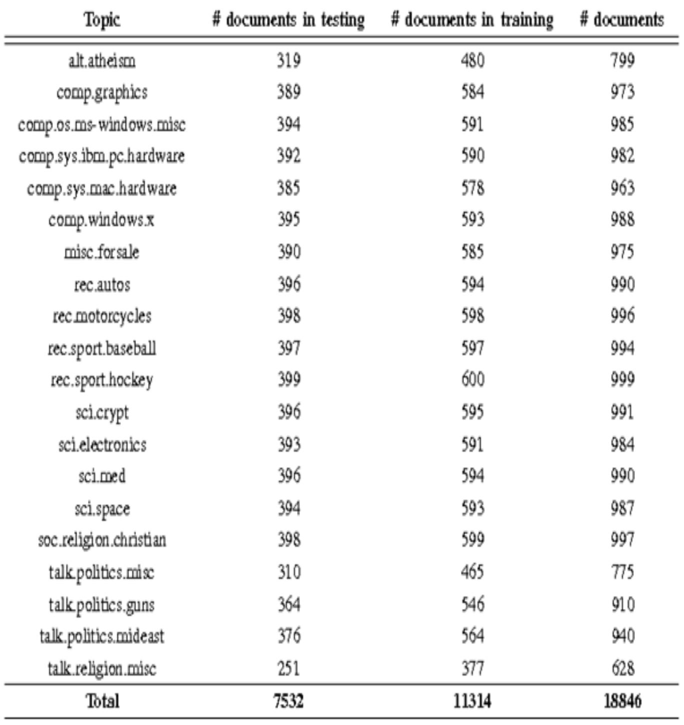
1.8 Data Set :

Data sets are collections of pre-classified documents. They are essential to develop and evaluate a TC system, that is, to train the system and then to test how well it behaves, when given a new document to classify
A data set consists of a set of documents, along with the category or categories that each document belongs to. In a first step, called the training phase, some of these documents (called the training documents) are used to train the TC system, by allowing it to learn a model of the data. Afterwards, in a step called the test phase, the rest of the documents (called the test documents) are used to test the TC system, to see how well the system behaves when classifying previously unseen documents.

Some of the publicly available collections are more used than others. In the TC field, and in the single-label sub-field in particular, the most commonly used collections are the 20-Newsgroups collection, the Reuters-21578 collection, and the Webkb collection

1.8.1 The $20-$Newsgroups Collection

The 20-Newsgroups collection was downloaded from Jason Rennie's page and the "Bydate" version was used, because it already had a standard train/test split. This is a collection of approximately 20,000 newsgroup messages, partitioned (nearly) evenly across the 20 different newsgroups mentioned .Although already cleaned-up, this collection still had several attachments, many PGP keys and approximately 4% of the articles are cross-posted. After removing them and the messages that became empty because of it, the distribution of training and test messages for each newsgroup is presented infigure (1.8). From now on, this particular version of the 20-Newsgroups collection shall be referred to as the 20Ng data set



Topic	# documents in testing	# documents in training	# documents
alt.atheism	319	480	799
comp.graphics	389	584	973
comp.os.ms-windows.misc	394	591	985
comp.sys.ibm.pc.hardware	392	590	982
comp.sys.mac.hardware	385	578	963
comp.windows.x	395	593	988
misc.forsale	390	585	975
rec.autos	396	594	990
rec.motorcycles	398	598	996
rec.sport.baseball	397	597	994
rec.sport.hockey	399	600	999
sci.crypt	396	595	991
sci.electronics	393	591	984
sci.med	396	594	990
sci.space	394	593	987
soc.religion.christian	398	599	997
talk.politics.misc	310	465	775
talk.politics.guns	364	546	910
talk.politics.mideast	376	564	940
talk.religion.misc	251	377	628
Total	**7532**	**11314**	**18846**

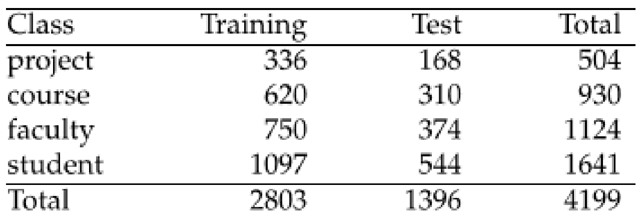
Figure 1.8: Test and Total number of documents for each of the 20 classes of the 20-Newsgroups collection, considering the standard Bydate split— 20Ng dataset

1.8.2 The Reuters-21578 Collection

All the documents contained in the Reuters -21578 collection appeared on the Reuters news wire and were manually classified by personnel from Reuters Ltd. This collection is very skewed, with documents very unevenly distributed among different classes.train test split is generally used for classification tasks.

1.8.3 The Webkb Collection

The Webkb collection contains web pages from computer science departments collected by the Worldwide Knowledge Base(WebKb) project of the CMU text learning group in 1997. For each of the different classes, the collection contains pages from four universities (Cornell, Texas, Washington and Wisconsin), and other miscellaneous pages collected from other universities.



Class	Training	Test	Total
project	336	168	504
course	620	310	930
faculty	750	374	1124
student	1097	544	1641
Total	2803	1396	4199

Figure 1.9: Training, Test and Total number of documents for each of the 4 classes of the Webkb collection, considering my random split — Web4 dataset.

Conclusions

The classification task is one of the most indispensable problems in machine learning. As text and document data sets proliferate. However, the existing text classification algorithms work more efficiently if we have a better understanding of feature extraction methods and how to evaluate them correctly. Currently, text classification algorithms can be chief classified in the following manner: Feature extraction methods, such as Term Frequency-Inverse document frequency ($TF - IDF$), term frequency (TF)... We described the basic methods of text preprocessing step. And we talk about *Dimensionality* reduction methods, such as principal component analysis Existing classification algorithms, such as Naive Bayes Classified (NBC), k-nearest Neighbor (KNN), Support Vector Machine (SVM). Evaluation methods, such as accuracy, F1 measure, With these metrics, the text classification algorithm can be evaluated. Finally, the Data sets are collections of pre classified documents. They are essential to develop and evaluate a TC system like 20−Newsgroups, the Reuters-21578, Webkb collection

Chapter 2

System Design

Introduction

In this chapter we present the experimental setting describe architecture of our work , set experiments was run on data collected(data set) 20 news groups.It missed all stages preprocessing,feature weighting ,feature selection,machine learning using different algorithm of classification (nb,knn,svm) and finally evaluate our system.and also we are going to talk about related work people who work with different algorithm in text classification

2.1 Related Work

In the recent years, the progress of web and social net work technologies have led to a massive interest in the classification of text documents containing links or other meta-information and many studies on classification algorithms have been done by many researches. In this section we will do a review to these works and show the focus points of them. As we will see,
Aggarwal and Zhai (2012) Focused on specific changes which are applicable for the text classification. They used, as text classification algorithms, Decision Trees, Pattern (Rule)- based Classifiers, SVM Classifiers, Neural Network Classifiers, Bayesian (Generative) Classifiers, nearest neighbor classifiers, and genetic algorithm-based classifier. They discussed the methods for features selection in text classification and described these methods for text classification[29]
Korde and Mahender (2012) Gave an introduction to textclassification, process of text classification as well as the overview of the classifiers and tried to compare the some existing classifier on basis of few criteria like time complexity, principal and performance.[30]
Mamoun and Ahmed (2014) highlighted the algorithms that are applied to the text classification and gave a comparative study on different types of approaches to the text categorization. They compared between the accuracy of the utilized algorithms and its results.[31]

2.2 Experimental Setting

this section describes our detailed experimental evaluation we evaluate the performance of standard text classification algorithm in different setting .To evaluate our experiments we computed recall,precision and f1-measure.

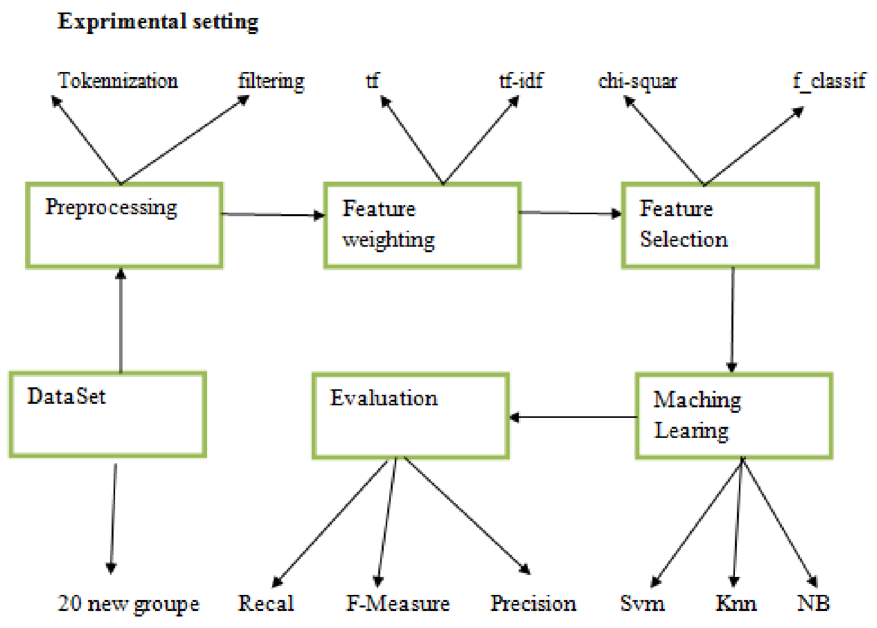


Figure 2.1: Understanding the structure of our study

2.2.1 Data Sets

The first step is the Dataset Preparation step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then splitted into train and validation sets.we work The 20 newsgroups data set comprises around 18846 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation).

2.2.2 Preprocessing

the next step Pre-processing is actually a trial to improve text classification by removing worthless information. It may include removal of numbers, punctuation (such as hyphens), and stop words, which are words that can be found in any text like prepositions and pronouns. In addition,English texts need more consideration in this stage because of their writing style.and the morphological nature of English , some researchers consider root extraction and word stemming as a part of preprocessing).

In our opinion, using the full form of the word, its stem or root, is part of the feature extraction step.

2.2.3 Feature Weighting

The next step is the Feature Engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data
■TF-IDF : an alternative is to calculate word frequencies, and by far the most popular method is called TF-IDF. This is an acronym than stands for "Term Frequency – Inverse Document" Frequency which are the components of the resulting scores assigned to each word
■Term Frequency(TF) : This summarizes how often a given word appears within a document.
■ Inverse Document Frequency(IDF) : This downscales words that appear a lot across documents.
Without going into the math, TF-IDF are word frequency scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents.

2.2.4 Feature Selection

In this step we added a new step in the preprocessing phase which is feature selection. In general, the size of the training data sets are very large. To reduce the high dimensionality of the words by removing irrelevant and redundant information, feature selection was performed. In this case the features are the words to be trained in documents. Feature selection usually used to reduce the size of the training corpus to an acceptable level. The benefit of feature selection also includes a small improvement in predication accuracy in some cases To select the most appropriate words in the document, The most frequently used methods have been Chi Squared and $fclassif$

2.2.5 Machine Learing

In this step, the training matrix that contains the selected features and their corresponding weights in each text of the training data are used to train the classification algorithm. Classical machine learning algorithms have been the most used in text classification, such as Naive Bayes (NB)) k-nearest neighbor (KNN) and support vector machine (SVM)
The training process yields a classification model that will be tested by means of the testing data. The same features that were extracted from the training data and the same weighting methods will be used to test the classification model.

Naive Bayes

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes' Theorem to predict the tag of a text (like a piece of news or a customer review).
They are probabilistic, which means that they calculate the probability of each tag for a given text, and then output the tag with the highest one. The way they get these probabilities is by using Bayes' Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature. We're Let's see how this works in practice with a simple example. Suppose we are building a classifier that says whether a text is about sports or not. Our training data has 5 sentences :

Text	Tag
" A great game "	Sports
" The election was over "	Not-sports
" Very clean match "	Sports
" A clean but forgettable game "	Sports
" It was a close election "	Not-sports

Table 2.1: example show 5 sentences with tag

Now, which tag does the sentence A very close game belong to?
Since Naive Bayes is a probabilistic classifier, we want to calculate the probability that the sentence "A very close game" is Sports and the probability that it's Not Sports. Then, we take the largest one. Written mathematically, what we want is P(Sports | a very close game)— the probability that the tag of a sentence is Sports given that the sentence is "A very close game".

Bayes Theorem is useful when working with conditional probabilities (like we are doing here), because it provides us with a way to reverse them:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{2.1}$$

In our case, we have , so using this theorem we can reverse the conditional probability

$$P(sports|averycloseegam) = \frac{P(averycloseegam)|sports) \times P(sports)}{P(averyclosegame)} \tag{2.2}$$

Since for our classifier we're just trying to find out which tag has a bigger probability, we can discard the divisor which is the same for both tags and just compare :p(a very close game$\setminus$ Sport) x p(Sports)
With : p(a very close game$\setminus$ sport) x p(Not Sports)

This is better, since we could actually calculate these probabilities! Just count how many times the sentence "A very close game" appears in the Sports tag, divide it by the total, and obtain :p(a very close game$\diagdown$Sport)

There's a problem though: "A very close game" doesn't appear in our training data, so this probability is zero.
So here comes the Naive part: we assume that every word in a sentence is independent of the other ones. This means that we're no longer looking at entire sentences, but rather at individual words. So for our purposes, "this was a fun party" is the same as "this party was fun" and "party fun was this". We write this as: p(a very close game)=p(a) x p(very) x p(close) x p(game)
This assumption is very strong but super useful. It's what makes this model work well with little data or data that may be mislabeled. The next step is just applying this to what we had before: p(a very close game$\diagdown$Sports)=p(a$\diagdown$Sports) x p(very$\diagdown$Sports) x p(close$\diagdown$Sports) x p(game$\diagdown$Sports)
And now, all of these individual words actually show up several times in our training data, and we can calculate them.

The final step is just to calculate every probability and see which one turns out to be larger. Calculating a probability is just counting in our training data.First, we calculate the a prior probability of each tag: for a given sentence in our training data, the probability that it is Sports P(Sports) is $\frac{2}{3}$. Then, P(Not Sports) is $\frac{2}{3}$. Then, calculating $P(game \setminus Sports)$ means counting how many times the word " game " appears in Sports texts (2) divided by the total number of words in sports (11). Therefore P(game | Sports) =2/11

However, we run into a problem here: " close " doesn't appear in any Sports text! That means that P(close\ Sports) = 0. This is rather inconvenient since we are going to be multiplying it with the other probabilities, so we'll end up with P(a \ Sports) /times P(very \ Sports) /times 0 /times P(game \ Sports). This equals 0, since in a multiplication if one of the terms is zero, the whole calculation is nullified. Doing things this way simply doesn't give us any information at all, so we have to find a way around.

By using something called Laplace smoothing: we add 1 to every count so it's never zero. To balance this, we add the number of possible words to the divisor, so the division will never be greater than 1. In our case, the possible words are ['a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match']
Since the number of possible words is 14 (I counted them), applying smoothing we get that P(game \ sports) = (2 + 1)/(11 + 14). The full results are:

Now we just multiply all the probabilities, and see who is bigger:
⋇p(a\ Sports) x p(very\ Sports) x p(close\ Sports) x p(game \ Sports) x p(Sports)=$2,76 \star 10^{-5}$=$0,0000276$
▷ p(a\Not Sports) x p(very\ Not Sports) x p(close\ Not Sports) x p(game\ Not Sports) x p(Not Sports)= 0,572$\star 10^{-5}$=0,00000572

Word	P(word/Sports)	P(word /Not Sports)
A	(2+1)/(11+14)	(1+1)/(11+14)
very	(1+1)/(11+14)	(0+1)/(11+14)
close	(0+1)/(11+14)	(1+1)/(9+14)
game	(2+1)/(11+14)	(0+1)/(9+14)

Table 2.2: example Native Bayes calculate

K-Nearest Neighbors

K Nearest Neighbors works by looking at the K closest points to the given ne data point (the one we want to classify) and picking the class that occurs the most to be the predicted value. This is why this algorithm typically works best when we can identify clusters of points in our data set.

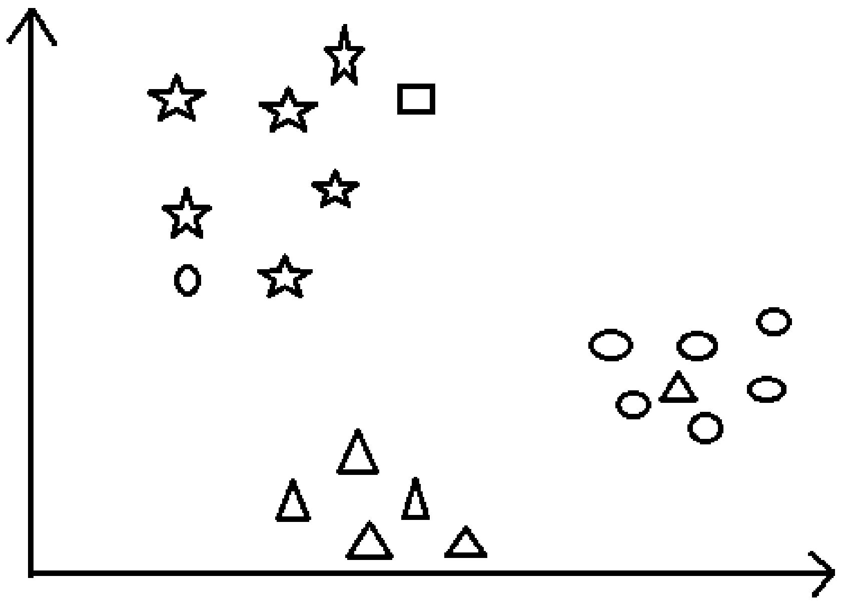


Figure 2.2: example show 3 class different

We can clearly see that there are groups of the same class points clustered together. Because of this correlation we can use the K-Nearest Neighbors algorithm to make accurate classifications. Example Let's have a look at the following example. We want to classify the Square as either ,triangle or Circle. Simply looking at the data we can see that it clearly belongs to the star class, but how do we design an algorithm to do this? First, we need to define a K value. This is going to be how many shapes we should use to make our decision. The K closest points to the black dot. Next, we need to find out the class of these K .Finally, we determine which class appears the most out of all of our K shaps and that is our prediction.

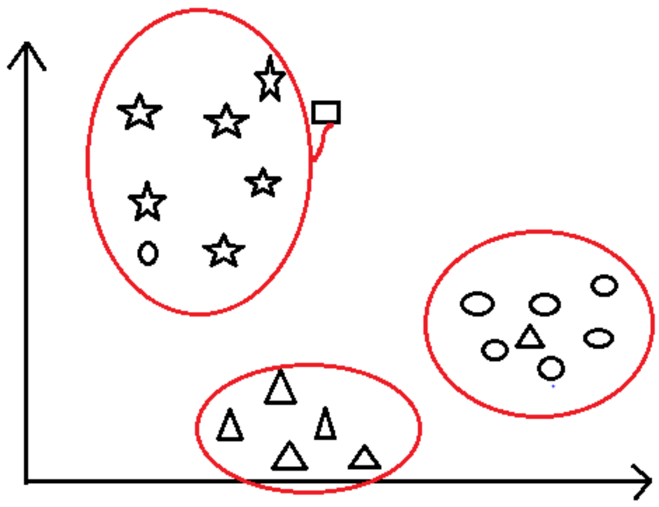


Figure 2.3: example knn classification

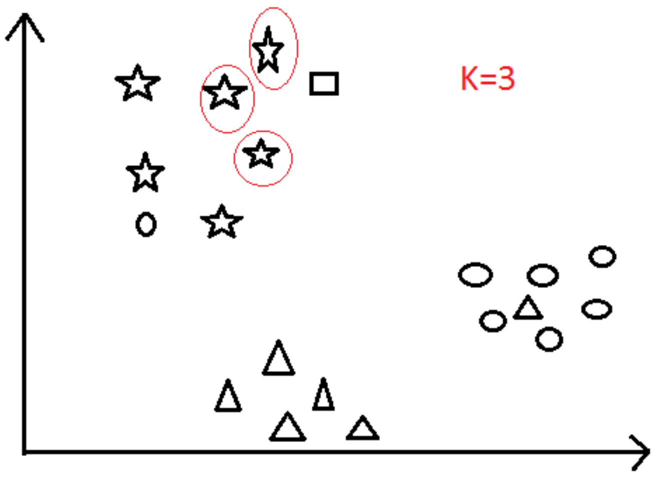
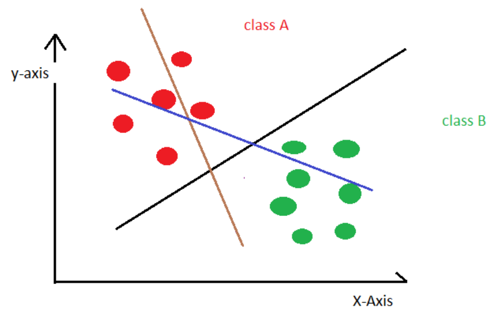


Figure 2.4: knn classification

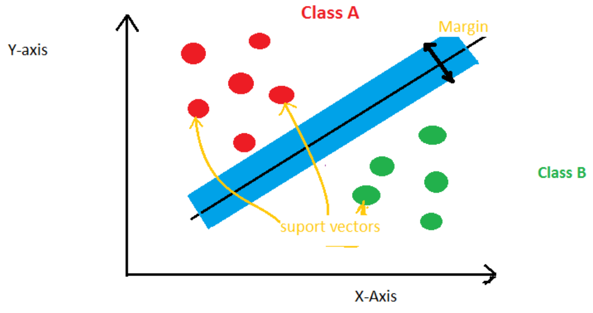
In this example our K value is 3. The 3 closest shapes to our square are the ones that have small square on them. All three of these shapes are stars, therefore red occurs the most. So we classify the square as apart of the stars group.

Support Vector Machine

The main objective is to segregate the given data set in the best possible way. The distance between the either nearest points is known as the margin. The objective is to select a hyper plane with the maximum possible margin between support vectors in the given data set. SVM searches for the maximum marginal hyper plane in the following steps:
• hyper planes which segregates the classes in the best way. Left-hand side figure showing three hyper planes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.
• the right hyper plane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.



(a) figure showing three hyper planes black



(b) hyper plane with the maximum segregation

Figure 2.5: example svm classification

2.2.6 Evaluation

The ability of the classification model to classify texts into the correct classes results from all the previously described steps. A number of methods have been used to assess the performance of the classification model output, such as accuracy, precision and recall, and f-measure From the data summarized, it is difficult to suggest which combination of feature selection method, term weighting, and classification algorithm is the optimal solution for text classification because most of the data sets used are small and are mainly from the news genre. In the following chapter we are going to see our result

conclusion

This chapter we had present the experimental setup for the experiments performed during this work, namely the classification methods, the evaluation metrics and the datasets that were used.and also we had see related work people ho with different algorithm in text classification

Chapter 3

Result and discussion

Introduction

In this chapter the result of the study are presented and discussed for two case the first case raw data and second filtered(stops words and stemming).in every case we compare the result,to do this work we work with python using different library.

3.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed[27].

3.2 Python library

Python library is a collection of functions and methods that allows you to perform many actions without writing your code. Each library in Python contains a huge number of useful modules that you can import for your every day programming. For example Scikit-learn, NLTK , NumPy. . .

3.2.1 Scikit-Learn

Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency. It has minimal dependencies and is distributed under the simplified BSD license, encouraging its use in both academic and commercial settings. Source code, binaries, and documentation can be downloaded from http://scikit-learn.sourceforge.net[28]

For 20 news group we use sklearn datasets fetch 20 news gropes, returnees a list of the raw texts that can be fed to text feature extractions such as sklearn feature extraction text CountVectorizer with custom parameters so as to extract feature vectors. The second one, sklearn datasets fetch 20 news groups vectoriser , returns ready to use features, it is not necessary to use a feature extractor.
The sklearn datasets fetch 20newsgroups function is a data fetching caching functions that downloads the data archive from the original 20 news groups website, extracts the archive contents in the scikit learn data 20news home folder and calls the sklearn datasets load files on either the training or testing set folder, or both of them.
* The sklearn feature extraction text CountVectorizer The provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary. You can use it as follows:
○ create an instance of the Count *Vectorizer* class
○ Call the fit() function in order to learn a vocabulary from one or more documents
○ Call the transform() function on one or more documents as needed to encode each as a vector An encoded vector is returned with a length of the entire vocabulary and an integer count for the number of times each word appeared in the document
Expmle :
We have a text:$''Thequickbrownfoxjumpedoverthelazydog.''$
Assign a unique number to each word as: also known as $''Tokenize''$
$'the' : 7, 'lazy' : 4, 'jumped' : 3, 'brown' : 0, 'over' : 5, 'quick' : 6, 'dog' : 1, 'fox' : 2$
$Featuresare : (Vocabulary)[8features]['brown', 'dog', 'fox', 'jumped', 'lazy', 'over', 'quick', 'the']$
In ML terms: Learn a vocabulary dictionary of all tokens in the raw documents, and it is done by using CountVectorizer fit()
Count the occurrence of each word: basically in ML terms "encoding documents" It is done by using CountVectorizer transform()
$['brown', 'dog', 'fox', 'jumped', 'lazy', 'over', 'quick', 'the'][[11111112]]$
It stores it as an array and its shape is: (1,8) i.e 1 no. of sample and 8 no. of features.
And for tf-id i use TfidfVectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents.
Alternately, if you already have a learned CountVectorizer, you can use it with a TfidfTransformer to just calculate the inverse document frequencies and start encod-

ing documents.
Exapmle :
The same create, fit, and transform process is used as with the CountVectorizer.
We have a text:
$["Thequickbrownfoxjumpedoverthelazydog",$
$"Thedog",$
$"Thefox"]$
Here it learns the IDF from the count matrix obtained from CountVectorizer So for above text, the IDF obtained is:
$[1.69314718, 1.28768207, 1.28768207, 1.69314718, 1.69314718, 1.69314718, 1.69314718, 1]$
The inverse document frequencies are calculated for each word in the vocabulary, assigning the lowest score of 1.0 to the most frequently observed word: 1*the*] .
For Select features according to the k highest scores. It takes as a parameter a score function, which must be applicable to a pair(X, y).
The score function must return an array of scores, one for each feature $X[:, i]$ of X (additionally, it can also return p-values, but these are neither needed nor required).
We use SelectKBest then simply retains the first k features of X with the highest scores.
So, for example, if you pass chi2 as a score function, SelectKBest will compute the chi2 statistic between each feature of X and y (assumed to be class labels).
A small value will mean the feature is independent of y. A large value will mean the feature is non-randomly related to y, and so likely to provide important information.
Only k features will be retained.
Finally, SelectKBest has a default behaviour implemented, so you can write $select = SelectKBest()$ and then call select fit transform(X, y) (in fact I saw people do this).
In this case SelectKBest uses the f classif score function. This interpretes the values of y as class labels and computes, for each feature X[:,i] of X, an F-statistic. The formula used is exactly the one given here:one way ANOVA F-test, with K the number of distinct values of y.
A large score suggests that the means of the K groups are not all equal.
This is not very informative, and is true only when some rather stringent conditions are met: for example, the values $X[:, i]$ must come from normally distributed populations, and the population variance of the K groups must be the same.
I don't see why this should hold in practice, and without this assumption the F-values are meaningless. So using *SelectKBest* carelessly might throw out many features for the wrong reasons.and also we use The SelectPercentile class has two attributes: scores and pvalues.
SelectPercentile is able to select features according to different metrics of quality for the features.
The specific metric used is determined by the score func argument, which according to the documentation must be: Function taking two arrays X and y, and returning a pair of arrays (scores, pvalues) or a single array with scores. Default is f classif (see below "See also").

The default function only works with classification tasks So by default it uses f classif as a measure of features' quality. Looking at the documentation of this

function you can see that it computes "the ANOVA $F - value$", so that is where your p value comes from.finally going to be working with an algorithm called Multinational Naive Bayes.
We'll walk through the algorithm applied to NLP with an example, and will you know how this method works.

3.3 Result

3.3.1 Raw Word

	tf-idf	chi-2	fs($fclassif$)
naive byes	77.38	76.42	79.20
svm	82.38	71.53	74.41
knn	65.91	33.17	34.99

Table 3.1: classification accuracy(%) using tf-idf with different algorithm

table (3.1) shows the classification accuracy with tf-idf using two different term selection methods with three algorithms of classification .The svm classifier shows the highest accuracy among the three classifier 82.38% but when we use feature selection The result decreases 71,53%, the nb classifier gave us a good accuracy 77,38% but when we use feature selection function f-classif the accuracy give better result 79.20% and finally the knn methods recorded the lowest accuracy 65,91% even when we used feature selection give low accuracy 33,17%

	tf	chi-2	fs($fclassif$)
naive byes	70,52	72,18	70,85
svm	77,53	75,06	77,07
knn	40,83	39,92	41,17

Table 3.2: classification accuracy(%) using tf with different algorithme

table (3.2) shows the classification accuracy with tf using two different feature selection methods with three algorithms of classification .the svm classifier shows the highest accuracy among the three classifier 77,53% but when we feature selection The result is not affected much so the accuracy show 77,07%, (75,06% The result is close if we use ch-2),the nb classifier give Results are close accuracy 70,52% even when we use term selection methods Do not raise the score too much 72,18% and finally the knn algorithm recorded the lowest accuracy 40,83% even when we used feature selection give close result 41,17%

3.3.2 Filtred

	tf-idf	chi-2	fs($fclassif$)
naive byes	81,14	77,18	80,09
svm	81,94	75,43	77,62
knn	68,70	44,13	40.28

Table 3.3: classification accuracy(%) using tf-idf with different algorithme

table (3.3) shows the classification accuracy with tf-idf using filtred (stop word and stemming), two different term selection methods with three algorithms of classification .the svm and nb classifier shows the highest accuracy 81,94% (svm) 81,14%(nb) the result are very close but when we feature selection The result of svm decreases 75,43% not so mutch and nb we can say it stay the same 80,09% , finally the knn alorithm recorded the lowest accuracy 68,70% but when we used feature selection the accuracy decreases 40,28%

	tf	chi-2	fs($fclassif$)
naive byes	78,90	79,32	79,00
svm	78,77	77,16	78,75
knn	64,71	34,28	50,87

Table 3.4: classification accuracy(%) using tf with different algorithme

table (3.4) shows the classification accuracy with tf using filtred (stop word and stemming), two different term feature selection methods with three algorithms of classification .The svm and nb classifier shows the highest accuracy 78,90% (nb) 78,77%(svm) the result are very close even when we use feature selection The result stay near the same 79,0% not so match difference between nb and svm , finally the knn alorithm recorded the lowest accuracy 64,71% but when we used feature selection ch-2 the accuracy decreases 34,28%

■ From this section we can noticed that ,feature selection did not effect ,or slightly effected, maximum entropy, naïve bayes, and support vector machines and k-Nearest Neighbor show low result and if you use knn with feature selection they effect your result . But, still naïve bayes is the best classifier and support vector machines follows.

3.3.3 Classification of Class By Category

	precision	recall	f1-score
alt.atheism	0.73	0.72	0.72
comp.graphics	0.80	0.70	0.74
comp.os.ms-windows.misc	0.73	0.76	0.75
comp.sys.mac-hardware	0.71	0.70	0.70
comp.sys.mac-hardware	0.83	0.81	0.82
comp.windows.x	0.83	0.77	0.80
misc.forsale	0.84	0.90	0.87
rec.autos	0.92	0.89	0.91
rec.motorcycle	0.92	0.96	0.94
rec.sport.baseball	0.89	0.90	0.89
rec.sport.hockey	0.88	0.99	0.93
sci.crypt	0.83	0.96	0.89
sci.electronics	0.83	0.60	0.70
sci.med	0.87	0.86	0.86
sci.space	0.84	0.96	0.89
soc.religion.christian	0.76	0.94	0.84
talk.politics.guns	0.70	0.92	0.80
talk.politics.mideast	0.90	0.93	0.92
talk.politics.misc	0.89	0.55	0.68
talk.religion.misc	0.85	0.40	0.55

Table 3.5: Table classification of category(raw word) using Svm with tf-idf

discussion

Table(3.5)show classification of class by category with svm algorithm using tf-idf .From the table we can noticed that ,displays the f1-measure of each categories with svm classifier before filtering We noticed that rec.motorcycle category is recognized very well by svm classifiers f1-score= 94% . on the other hand talk religion.music is the weakest categories

	precision	recall	f1-score
alt.atheism	0.81	0.66	0.73
comp.graphics	0.78	0.72	0.75
comp.os.ms-windows.misc	0.82	0.69	0.75
comp.sys.ibm.pc-hardware	0.69	0.80	0.74
comp.sys.mac-hardware	0.86	0.82	0.84
comp.windows.x	0.86	0.79	0.83
misc.forsale	0.88	0.75	0.81
rec.autos	0.87	0.93	0.90
rec.motorcycles	0.92	0.96	0.94
rec.sport.baseball	0.93	0.92	0.92
rec.sport.hockey	0.90	0.98	0.94
sci.crypt	0.71	0.97	0.82
sci.electronics	0.83	0.66	0.73
sci.med	0.92	0.79	0.85
sci.space	0.82	0.94	0.88
soc.religion.christian	0.60	0.97	0.74
talk.politics.guns	0.65	0.94	0.77
talk.politics.mideast	0.92	0.94	0.93
talk.politics.misc	0.94	0.48	0.64
talk.religion.misc	0.95	0.22	0.35

Table 3.6: Classification of category stemming using Nb and tf-idf

discussion

Table(3.6)show classification of class by category with naive Bayes classifiers filtered we found that the accuracy of rec.motorcycle and rec.sports.hockey is the best the f1-score for each category is =0.94 and religion.music is the worst f1.score= 0.35

	precision	recall	f1-score
alt.atheism	0.58	0.75	0.66
comp.graphics	0.49	0.67	0.57
comp.os.ms-windows.misc	0.54	0.62	0.58
comp.sys.ibm.pc-hardware	0.52	0.60	0.56
comp.sys.mac-hardware	0.60	0.58	0.59
comp.windows.x	0.68	0.56	0.62
misc.forsale	0.53	0.45	0.49
rec.autos	0.73	0.72	0.72
rec.motorcycles	0.82	0.86	0.84
rec.sport.baseball	0.74	0.76	0.75
rec.sport.hockey	0.83	0.87	0.85
sci.crypt	0.77	0.83	0.80
sci.electronics	0.69	0.55	0.61
sci.med	0.82	0.61	0.70
sci.space	0.79	0.81	0.80
soc.religion.christian	0.81	0.79	0.80
talk.politics.guns	0.73	0.74	0.74
talk.politics.mideast	0.81	0.74	0.78
talk.politics.misc	0.76	0.62	0.68
talk.religion.misc	0.61	0.55	0.57

Table 3.7: classification of category stemming using knn with tf-idf

discussion

Table(3.7)show classification of class by category in classifier knn with stemming We can see that the maximum accuracy is rec.sport.hockey category f1-score = 85 and the minimum accuracy is forsale category f1-score= 0.49
• we can say that the rec.motorcycles category accuracy has not changed with svm and naive Bayes and also rec.sport.hockey category accuracy with naive byes Still the best and talk.religion.misc category accuracy Is still weaker for all classifiers
• if you want see more detail chek annex

conclusion

Analysed the text classification using 20 new group with preprocessing ,feature weighing(tf-idf),feature selection(ch2,f-classif),machine learning(nb,svm,knn)than we evaluate our work by Precision ,recall and f1 measure as result we can say that feature selection did not effect ,or slightly effected, maximum entropy, naïve bayes, and support vector machines ,k-Nearest Neighbor show low result ,if you use knn with feature selection they effect your result and naïve bayes is the best classified and support vector machines .

General Conclusion

The text classification problem is an Artificial Intelligence research topic, especially given the vast number of documents available in the form of web pages and other electronic texts like emails, discussion forum postings and other electronic documents. Text Classification is an important application area in text mining why because classifying millions of text document manually is an expensive and time consuming task. in our study we take the different steps of text classification pre-processing and feature extraction than feature selection and machine learning methods and finally confusion matrix. and also we tray to improve the accuracy we compared three different classification knn, svm and naive Bayes and two type of term weighting tf and tf-idf and also two type future selection ch-square and f-classif
Therefore, automatic text classifier is constructed using pre classified sample documents whose accuracy and time efficiency is much better than manual text classification.
If the input to the classifier is having less noisy data, we obtain efficient results. So during mining the text, efficient reprocessing algorithms must be chosen. The test data also should be preprocessed before classifying it. Text can be classified better by identifying patterns . Once patterns are identified we can classify given text or documents efficiently. Identifying efficient patterns also plays major role in text classification. Text classification techniques need to be designed to effectively manage large numbers of elements with varying frequencies. Almost all the known techniques for classification such , Bayes methods, nearest neighbor classifiers, SVM classifiers,
It has observed that even for a specified classification method, classification performances of the classifiers based on different training text corpuses are different; and in some cases such differences are quite substantial. This observation implies that a) classifier performance is relevant to its training corpus in some degree, and b) good or high quality training corpuses may derive classifiers of good performance. Unfortunately, up to now little research work in the literature has been seen on how to exploit training text corpuses to improve classifier's performance
Classification accuracy is very hight some time without feature selection and some time the accuracy we will be rise if we use feature selection related to type of the algorithm used also the accuracy will be rise if you filtering your data set

Appendices

	precision	recall	f1-score
alt.atheism	0.73	0.72	0.72
comp.graphics	0.80	0.70	0.74
comp.os.ms-windows.misc	0.73	0.76	0.75
comp.sys.mac-hardware	0.71	0.70	0.70
comp.sys.mac-hardware	0.83	0.81	0.82
comp.windows.x	0.83	0.77	0.80
misc.forsale	0.84	0.90	0.87
rec.autos	0.92	0.89	0.91
rec.motorcycle	0.92	0.96	0.94
rec.sport.baseball	0.89	0.90	0.89
rec.sport.hockey	0.88	0.99	0.93
sci.crypt	0.83	0.96	0.89
sci.electronics	0.83	0.60	0.70
sci.med	0.87	0.86	0.86
sci.space	0.84	0.96	0.89
soc.religion.christian	0.76	0.94	0.84
talk.politics.guns	0.70	0.92	0.80
talk.politics.mideast	0.90	0.93	0.92
talk.politics.misc	0.89	0.55	0.68
talk.religion.misc	0.85	0.40	0.55

Table 8: Classification of category(raw word) using Svm with tf-idf

	precision	recall	f1-score
alt.atheism	0.80	0.52	0.63
comp.graphics	0.81	0.65	0.72
comp.os.ms-windows.misc	0.82	0.65	0.73
comp.sys.ibm.pc-hardware	0.67	0.78	0.72
comp.sys.mac-hardware	0.86	0.77	0.81
comp.windows.x	0.89	0.75	0.82
misc.forsale	0.93	0.69	0.80
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.94	0.93	0.93
rec.sport.baseball	0.92	0.90	0.91
rec.sport.hockey	0.89	0.97	0.93
sci.crypt	0.59	0.97	0.74
sci.electronics	0.84	0.60	0.70
sci.med	0.92	0.74	0.82
sci.space	0.84	0.89	0.87
soc.religion.christian	0.44	0.98	0.61
talk.politics.guns	0.64	0.94	0.76
talk.politics.mideast	0.93	0.91	0.92
talk.politics.misc	0.96	0.42	0.58
talk.religion.misc	0.97	0.14	0.24

Table 9: Classification of category(raw word) using Native Bayes with tf-idf

	precision	recall	f1-score
alt.atheism	0.85	0.24	0.37
comp.graphics	0.71	0.60	0.65
comp.os.ms-windows.misc	0.69	0.65	0.71
comp.sys.ibm.pc-hardware	0.63	0.75	0.69
comp.sys.mac-hardware	0.86	0.68	0.76
comp.windows.x	0.88	0.68	0.77
misc.forsale	0.90	0.72	0.80
rec.autos	0.71	0.92	0.80
rec.motorcycles	0.84	0.91	0.87
rec.sport.baseball	0.86	0.85	0.86
rec.sport.hockey	0.90	0.93	0.91
sci.crypt	0.52	0.96	0.67
sci.electronics	0.72	0.52	0.63
sci.med	0.82	0.76	0.69
sci.space	0.83	0.81	0.82
soc.religion.christian	0.34	0.98	0.51
talk.politics.guns	0.66	0.80	0.73
talk.politics.mideast	0.96	0.72	0.82
talk.politics.misc	1.00	0.17	0.26
talk.religion.misc	1.00	0.01	0.02

Table 10: Classification of category(raw word) using Bayes with tf

	precision	recall	f1-score
alt.atheism	0.71	0.79	0.75
comp.graphics	0.52	0.77	0.62
comp.os.ms-windows.misc	0.20	0	0.01
comp.sys.ibm.pc-hardware	0.50	0.69	0.58
comp.sys.mac-hardware	0.67	0.81	0.73
comp.windows.x	0.69	0.70	0.74
misc.forsale	0.78	0.84	0.81
rec.autos	0.80	0.89	0.84
rec.motorcycles	0.84	0.92	0.88
rec.sport.baseball	0.90	0.90	0.90
rec.sport.hockey	0.97	0.91	0.94
sci.crypt	0.89	0.87	0.88
sci.electronics	0.71	0.67	0.69
sci.med	0.8	0.77	0.81
sci.space	0.84	0.87	0.86
soc.religion.christian	0.86	0.87	0.87
talk.politics.guns	0.74	0.90	0.81
talk.politics.mideast	0.96	0.82	0.88
talk.politics.misc	0.96	0.60	0.64
talk.religion.misc	0.64	0.53	0.58

Table 11: Classification of category(raw word) using Naive Bayes with TF-IDF and fs(chi-2)

	precision	recall	f1-score
alt.atheism	0.78	0.82	0.80
comp.graphics	0.72	0.80	0.70
comp.os.ms-windows.misc	0.2	0	0.01
comp.sys.ibm.pc-hardware	0.54	0.79	0.64
comp.sys.mac-hardware	0.85	0.78	0.81
comp.windows.x	0.70	0.80	0.75
misc.forsale	0.89	0.76	0.82
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.96	0.94	0.95
rec.sport.baseball	0.95	0.90	0.93
rec.sport.hockey	0.94	0.96	0.95
sci.crypt	0.97	0.94	0.86
sci.electronics	0.77	0.71	0.74
sci.med	0.87	0.82	0.85
sci.space	0.85	0.91	0.88
soc.religion.christian	0.77	0.95	0.85
talk.politics.guns	0.73	0.92	0.82
talk.politics.mideast	0.94	0.89	0.92
talk.politics.misc	0.65	0.61	0.63
talk.religion.misc	0.83	0.45	0.58

Table 12: Classification of category(raw word) using Native Bayes with tf-idf and fs(f-classif)

	precision	recall	f1-score
alt.atheism	0.72	0.62	0.67
comp.graphics	0.72	0.66	0.69
comp.os.ms-windows.misc	0.72	0.68	0.70
comp.sys.ibm.pc-hardware	0.65	0.69	0.67
comp.sys.mac-hardware	0.81	0.73	0.77
comp.windows.x	0.78	0.70	0.74
misc.forsale	0.73	0.91	0.81
rec.autos	0.88	0.84	0.86
rec.motorcycles	0.83	0.94	0.88
rec.sport.baseball	0.79	0.89	0.84
rec.sport.hockey	0.88	0.96	0.92
sci.crypt	0.76	0.95	0.85
sci.electronics	0.80	0.53	0.64
sci.med	0.74	0.79	0.76
sci.space	0.82	0.90	0.86
soc.religion.christian	0.73	0.92	0.81
talk.politics.guns	0.69	0.86	0.76
talk.politics.mideast	0.88	0.87	0.88
talk.politics.misc	0.87	0.50	0.64
talk.religion.misc	0.82	0.34	0.48

Table 13: Classification of category(raw word) using Svm with tf

	precision	recall	f1-score
alt.atheism	0.61	0.65	0.63
comp.graphics	0.61	0.68	0.65
comp.os.ms-windows.misc	0.63	0.64	0.63
comp.sys.ibm.pc-hardware	0.53	0.64	0.58
comp.sys.mac-hardware	0.75	0.66	0.70
comp.windows.x	0.75	0.66	0.70
misc.forsale	0.69	0.77	0.73
rec.autos	0.73	0.79	0.76
rec.motorcycles	0.90	0.78	0.84
rec.sport.baseball	0.65	0.87	0.75
rec.sport.hockey	0.90	0.91	0.91
sci.crypt	0.77	0.91	0.84
sci.electronics	0.77	0.50	0.60
sci.med	0.87	0.59	0.71
sci.space	0.82	0.88	0.85
soc.religion.christian	0.56	0.91	0.69
talk.politics.guns	0.73	0.70	0.72
talk.politics.mideast	0.92	0.74	0.82
talk.politics.misc	0.78	0.45	0.57
talk.religion.misc	0.57	0.36	0.44

Table 14: Classification of category(raw word) using svm with tf-idf and fs (chi-2)

	precision	recall	f1-score
alt.atheism	0.78	0.82	0.80
comp.graphics	0.62	0.80	0.70
comp.os.ms-windows.misc	0.20	0.00	0.01
comp.sys.ibm.pc-hardware	0.54	0.79	0.64
comp.sys.mac-hardware	0.85	0.78	0.81
comp.windows.x	0.70	0.80	0.75
misc.forsale	0.89	0.76	0.82
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.95	0.90	0.93
rec.sport.baseball	0.95	0.90	0.93
rec.sport.hockey	0.94	0.96	0.95
sci.crypt	0.79	0.94	0.86
sci.electronics	0.77	0.71	0.74
sci.med	0.87	0.82	0.85
sci.space	0.85	0.91	0.88
soc.religion.christian	0.77	0.95	0.85
talk.politics.guns	0.73	0.92	0.82
talk.politics.mideast	0.94	0.89	0.92
talk.politics.misc	0.65	0.61	0.63
talk.religion.misc	0.83	0.45	0.58

Table 15: Classification of category(raw word) using svm with tf-idf and fs (f-classif)

	precision	recall	f1-score
alt.atheism	0.43	0.76	0.55
comp.graphics	0.50	0.61	0.55
comp.os.ms-windows.misc	0.56	0.57	0.57
comp.sys.ibm.pc-hardware	0.53	0.58	0.56
comp.sys.mac-hardware	0.59	0.56	0.57
comp.windows.x	0.69	0.60	0.64
misc.forsale	0.58	0.45	0.51
rec.autos	0.75	0.69	0.72
rec.motorcycles	0.84	0.81	0.82
rec.sport.baseball	0.77	0.72	0.74
rec.sport.hockey	0.85	0.84	0.84
sci.crypt	0.76	0.84	0.80
sci.electronics	0.70	0.50	0.58
sci.med	0.82	0.49	0.62
sci.space	0.79	0.76	0.78
soc.religion.christian	0.75	0.76	0.76
talk.politics.guns	0.70	0.73	0.72
talk.politics.mideast	0.62	0.76	0.69
talk.politics.misc	0.55	0.61	0.58
talk.religion.misc	0.56	0.49	0.52

Table 16: Classification of category(raw word) using knn with tf-idf

	precision	recall	f1-score
alt.atheism	0.18	0.55	0.27
comp.graphics	0.36	0.34	0.35
comp.os.ms-windows.misc	0.46	0.31	0.38
comp.sys.ibm.pc-hardware	0.46	0.35	0.40
comp.sys.mac-hardware	0.51	0.24	0.33
comp.windows.x	0.40	0.49	0.44
misc.forsale	0.75	0.43	0.55
rec.autos	0.52	0.36	0.43
rec.motorcycles	0.75	0.53	0.62
rec.sport.baseball	0.63	0.34	0.44
rec.sport.hockey	0.74	0.49	0.59
sci.crypt	0.40	0.61	0.49
sci.electronics	0.62	0.20	0.30
sci.med	0.42	0.25	0.31
sci.space	0.75	0.37	0.49
soc.religion.christian	0.31	0.52	0.39
talk.politics.guns	0.46	0.50	0.48
talk.politics.mideast	0.31	0.56	0.40
talk.politics.misc	0.22	0.45	0.30
talk.religion.misc	0.35	0.27	0.31

Table 17: Classification of category(raw word) using knn with tf

	precision	recall	f1-score
alt.atheism	0.34	0.40	0.37
comp.graphics	0.16	0.41	0.23
comp.os.ms-windows.misc	0.26	0.37	0.30
comp.sys.ibm.pc-hardware	0.29	0.25	0.27
comp.sys.mac-hardware	0.20	0.27	0.23
comp.windows.x	0.37	0.22	0.27
misc.forsale	0.48	0.58	0.53
rec.autos	0.21	0.28	0.24
rec.motorcycles	0.47	0.45	0.46
rec.sport.baseball	0.35	0.29	0.32
rec.sport.hockey	0.50	0.38	0.43
sci.crypt	0.66	0.35	0.46
sci.electronics	0.20	0.17	0.18
sci.med	0.20	0.25	0.23
sci.space	0.53	0.34	0.41
soc.religion.christian	0.37	0.38	0.38
talk.politics.guns	0.55	0.33	0.41
talk.politics.mideast	0.66	0.41	0.50
talk.politics.misc	0.43	0.30	0.35
talk.religion.misc	0.32	0.15	0.21

Table 18: Classification of category(raw word) using knn with and tf-idf fs(chi-2)

	precision	recall	f1-score
alt.atheism	0.34	0.44	0.38
comp.graphics	0.12	0.52	0.20
comp.os.ms-windows.misc	0.28	0.36	0.31
comp.sys.ibm.pc-hardware	0.42	0.29	0.34
comp.sys.mac-hardware	0.32	0.23	0.27
comp.windows.x	0.43	0.23	0.29
misc.forsale	0.58	0.35	0.44
rec.autos	0.25	0.29	0.27
rec.motorcycles	0.67	0.45	0.54
rec.sport.baseball	0.42	0.28	0.34
rec.sport.hockey	0.38	0.46	0.42
sci.crypt	0.51	0.51	0.51
sci.electronics	0.45	0.14	0.21
sci.med	0.27	0.21	0.24
sci.space	0.77	0.29	0.43
soc.religion.christian	0.31	0.56	0.40
talk.politics.guns	0.55	0.35	0.43
talk.politics.mideast	0.56	0.53	0.55
talk.politics.misc	0.55	0.27	0.36
talk.religion.misc	0.44	m0.19	0.27

Table 19: Classification of category(raw word) using knn with tf-idf and fs(f-classif)

	precision	recall	f1-score
alt.atheism	0.78	0.82	0.80
comp.graphics	0.72	0.80	0.70
comp.os.ms-windows.misc	0.2	0	0.01
comp.sys.ibm.pc-hardware	0.54	0.79	0.64
comp.sys.mac-hardware	0.85	0.78	0.81
comp.windows.x	0.70	0.80	0.75
misc.forsale	0.89	0.76	0.82
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.96	0.94	0.95
rec.sport.baseball	0.95	0.90	0.93
rec.sport.hockey	0.94	0.96	0.95
sci.crypt	0.97	0.94	0.86
sci.electronics	0.77	0.71	0.74
sci.med	0.87	0.82	0.85
sci.space	0.85	0.91	0.88
soc.religion.christian	0.77	0.95	0.85
talk.politics.guns	0.73	0.92	0.82
talk.politics.mideast	0.94	0.89	0.92
talk.politics.misc	0.65	0.61	0.63
talk.religion.misc	0.83	0.45	0.58

Table 20: Classification of category(raw word) using Native Bayes with tf-idf and fs(f-classif)

	precision	recall	f1-score
alt.atheism	0.73	0.72	0.72
comp.graphics	0.80	0.70	0.74
comp.os.ms-windows.misc	0.73	0.76	0.75
comp.sys.mac-hardware	0.71	0.70	0.70
comp.sys.mac-hardware	0.83	0.81	0.82
comp.windows.x	0.83	0.77	0.80
misc.forsale	0.84	0.90	0.87
rec.autos	0.92	0.89	0.91
rec.motorcycle	0.92	0.96	0.94
rec.sport.baseball	0.89	0.90	0.89
rec.sport.hockey	0.88	0.99	0.93
sci.crypt	0.83	0.96	0.89
sci.electronics	0.83	0.60	0.70
sci.med	0.87	0.86	0.86
sci.space	0.84	0.96	0.89
soc.religion.christian	0.76	0.94	0.84
talk.politics.guns	0.70	0.92	0.80
talk.politics.mideast	0.90	0.93	0.92
talk.politics.misc	0.89	0.55	0.68
talk.religion.misc	0.85	0.40	0.55

Table 21: Classification of category(raw word) using Svm with tf-idf

	precision	recall	f1-score
alt.atheism	0.72	0.62	0.67
comp.graphics	0.72	0.66	0.69
comp.os.ms-windows.misc	0.72	0.68	0.70
comp.sys.ibm.pc-hardware	0.65	0.69	0.67
comp.sys.mac-hardware	0.81	0.73	0.77
comp.windows.x	0.78	0.70	0.74
misc.forsale	0.73	0.91	0.81
rec.autos	0.88	0.84	0.86
rec.motorcycles	0.83	0.94	0.88
rec.sport.baseball	0.79	0.89	0.84
rec.sport.hockey	0.88	0.96	0.92
sci.crypt	0.76	0.95	0.85
sci.electronics	0.80	0.53	0.64
sci.med	0.74	0.79	0.76
sci.space	0.82	0.90	0.86
soc.religion.christian	0.73	0.92	0.81
talk.politics.guns	0.69	0.86	0.76
talk.politics.mideast	0.88	0.87	0.88
talk.politics.misc	0.87	0.50	0.64
talk.religion.misc	0.82	0.34	0.48

Table 22: Classification of category(raw word) using Svm with tf

	precision	recall	f1-score
alt.atheism	0.80	0.52	0.63
comp.graphics	0.81	0.65	0.72
comp.os.ms-windows.misc	0.82	0.65	0.73
comp.sys.ibm.pc-hardware	0.67	0.78	0.72
comp.sys.mac-hardware	0.86	0.77	0.81
comp.windows.x	0.89	0.75	0.82
misc.forsale	0.93	0.69	0.80
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.94	0.93	0.93
rec.sport.baseball	0.92	0.90	0.91
rec.sport.hockey	0.89	0.97	0.93
sci.crypt	0.59	0.97	0.74
sci.electronics	0.84	0.60	0.70
sci.med	0.92	0.74	0.82
sci.space	0.84	0.89	0.87
soc.religion.christian	0.44	0.98	0.61
talk.politics.guns	0.64	0.94	0.76
talk.politics.mideast	0.93	0.91	0.92
talk.politics.misc	0.96	0.42	0.58
talk.religion.misc	0.97	0.14	0.24

Table 23: Classification of category(raw word) using Native Bayes with tf-idf

	precision	recall	f1-score
alt.atheism	0,81	0,52	0,63
comp.graphics	0,74	0,71	0,72
comp.os.ms-windows.misc	0.77	0.68	0.72
comp.sys.ibm.pc-hardware	0.67	0.74	0.71
comp.sys.mac-hardware	0.83	0.78	0.81
comp.windows.x	0.85	0.77	0.81
misc.forsale	0.87	0.79	0.83
rec.autos	0.86	0.91	0.88
rec.motorcycles	0.90	0.94	0.92
rec.sport.baseball	0.92	0.92	0.92
rec.sport.hockey	0.90	0.97	0.93
sci.crypt	0.71	0.95	0.81
sci.electronics	0.79	0.64	0.71
sci.med	0.90	0.80	0.85
sci.space	0.81	0.93	0.86
soc.religion.christian	0.56	0.97	0.71
talk.politics.guns	0.62	0.93	0.74
talk.politics.mideast	0.90	0.92	0.91
talk.politics.misc	0.98	0.42	0.59
talk.religion.misc	1.00	0.12	0.21

Table 24: Classification of category(raw word) using nb with tf

	precision	recall	f1-score
alt.atheism	0.80	0.52	0.63
comp.graphics	0.81	0.65	0.72
comp.os.ms-windows.misc	0.82	0.65	0.73
comp.sys.ibm.pc-hardware	0.67	0.78	0.72
comp.sys.mac-hardware	0.86	0.77	0.81
comp.windows.x	0.89	0.75	0.82
misc.forsale	0.93	0.69	0.80
rec.autos	0.85	0.92	0.88
rec.motorcycles	0.94	0.93	0.93
rec.sport.baseball	0.92	0.90	0.91
rec.sport.hockey	0.89	0.97	0.93
sci.crypt	0.59	0.97	0.74
sci.electronics	0.84	0.60	0.70
sci.med	0.92	0.74	0.82
sci.space	0.84	0.89	0.87
soc.religion.christian	0.44	0.98	0.61
talk.politics.guns	0.64	0.94	0.76
talk.politics.mideast	0.93	0.91	0.92
talk.politics.misc	0.96	0.42	0.58
talk.religion.misc	0.97	0.14	0.24

Table 25: Classification of category(raw word) using Native Bayes with tf-idf

	precision	recall	f1-score
alt.atheism	0.63	0.66	0.64
comp.graphics	0.73	0.69	0.71
comp.os.ms-windows.misc	0.70	0.70	0.70
comp.sys.ibm.pc-hardware	0.71	0.65	0.68
comp.sys.mac-hardware	0.76	0.77	0.77
comp.windows.x	0.84	0.73	0.78
misc.forsale	0.74	0.86	0.80
rec.autos	0.86	0.85	0.85
rec.motorcycles	0.90	0.93	0.92
rec.sport.baseball	0.90	0.86	0.88
rec.sport.hockey	0.85	0.98	0.91
sci.crypt	0.85	0.94	0.89
sci.electronics	0.79	0.60	0.68
sci.med	0.83	0.82	0.82
sci.space	0.70	0.93	0.80
soc.religion.christian	0.67	0.97	0.77
talk.politics.guns	0.91	0.86	0.87
talk.politics.mideast	0.90	0.92	0.91
talk.politics.misc	0.86	0.54	0.66
talk.religion.misc	0.75	0.31	0.44

Table 26: Classification of category stemming using svm with tf

	precision	recall	f1-score
alt.atheism	0.48	0.71	0.57
comp.graphics	0.43	0.61	0.50
comp.os.ms-windows.misc	0.50	0.62	0.56
comp.sys.ibm.pc-hardware	0.52	0.57	0.55
comp.sys.mac-hardware	0.56	0.52	0.54
comp.windows.x	0.51	0.51	0.51
misc.forsale	0.65	0.66	0.65
rec.autos	0.83	0.82	0.83
rec.motorcycles	0.70	0.71	0.70
rec.sport.baseball	0.70	0.71	0.70
rec.sport.hockey	0.81	0.83	0.82
sci.crypt	0.83	0.82	0.82
sci.electronics	0.66	0.48	0.56
sci.med	0.66	0.49	0.82
sci.space	0.77	0.73	0.75
soc.religion.christian	0.79	0.75	0.77
talk.politics.guns	0.72	0.72	0.72
talk.politics.mideast	0.81	0.70	0.75
talk.politics.misc	0.72	0.59	0.65
talk.religion.misc	0.56	0.48	0.52

Table 27: Classification of category stemming using knn with tf

	precision	recall	f1-score
alt.atheism	0.81	0.53	0.64
comp.graphics	0.73	0.71	0.72
comp.os.ms-windows.misc	0.77	0.68	0.72
comp.sys.ibm.pc-hardware	0.67	0.74	0.70
comp.sys.mac-hardware	0.84	0.79	0.81
comp.windows.x	0.84	0.77	0.80
misc.forsale	0.87	0.80	0.84
rec.autos	0.86	0.91	0.88
rec.motorcycles	0.91	0.94	0.92
rec.sport.baseball	0.91	0.92	0.91
rec.sport.hockey	0.78	0.64	0.70
sci.crypt	0.72	0.95	0.82
sci.electronics	0.78	0.64	0.70
sci.med	0.90	0.80	0.85
sci.space	0.81	0.92	0.86
soc.religion.christian	0.57	0.96	0.72
talk.politics.guns	0.62	0.93	0.74
talk.politics.mideast	0.91	0.92	0.91
talk.politics.misc	0.98	0.42	0.65
talk.religion.misc	1.00	0.13	0.59

Table 28: Classification of category stemming using nb with tf and fs f-claasif

	precision	recall	f1-score
alt.atheism	0.64	0.65	0.65
comp.graphics	0.73	0.69	0.71
comp.os.ms-windows.misc	0.70	0.70	0.70
comp.sys.ibm.pc-hardware	0.72	0.65	0.68
comp.sys.mac-hardware	0.76	0.77	0.76
comp.windows.x	0.83	0.74	0.78
misc.forsale	0.74	0.86	0.86
rec.autos	0.86	0.85	0.85
rec.motorcycles	0.88	0.94	0.91
rec.sport.baseball	0.91	0.85	0.88
rec.sport.hockey	0.84	0.97	0.90
sci.crypt	0.85	0.94	0.89
sci.electronics	0.77	0.61	0.68
sci.med	0.84	0.82	0.83
sci.space	0.82	0.92	30.87
soc.religion.christian	0.70	0.93	0.80
talk.politics.guns	0.67	0.90	0.77
talk.politics.mideast	0.90	0.86	0.88
talk.politics.misc	0.86	0.54	0.66
talk.religion.misc	0.76	0.31	0.45

Table 29: Classification of category stemming using svm with tf and fs f-classif

	precision	recall	f1-score
alt.atheism	0.70	0.58	0.64
comp.graphics	0.33	0.37	0.35
comp.os.ms-windows.misc	0.18	0.79	0.30
comp.sys.ibm.pc-hardware	0.53	0.47	0.50
comp.sys.mac-hardware	0.72	0.26	0.38
comp.windows.x	0.47	0.44	0.45
misc.forsale	0.57	0.33	0.42
rec.autos	0.72	0.50	0.59
rec.motorcycles	0.87	0.67	0.76
rec.sport.baseball	0.81	0.47	0.59
rec.sport.hockey	0.83	0.65	0.73
sci.crypt	0.91	0.70	0.79
sci.electronics	0.76	0.35	0.48
sci.med	0.70	0.32	0.44
sci.space	0.85	0.58	0.69
soc.religion.christian	0.25	0.79	0.38
talk.politics.guns	0.83	0.47	0.60
talk.politics.mideast	0.90	0.55	0.68
talk.politics.misc	0.92	0.45	0.61
talk.religion.misc	0.72	0.36	0.48

Table 30: Classification of category stemming using knn with tf and fs f-claasif

	precision	recall	f1-score
alt.atheism	0.75	0.62	0.68
comp.graphics	0.66	0.75	0.71
comp.os.ms-windows.misc	0.72	0.74	0.73
comp.sys.ibm.pc-hardware	0.66	0.72	0.69
comp.sys.mac-hardware	0.80	0.77	0.78
comp.windows.x	0.84	0.75	0.79
misc.forsale	0.84	0.82	0.83
rec.autos	0.84	0.89	0.87
rec.motorcycles	0.92	0.93	0.92
rec.sport.baseball	0.88	0.91	0.90
rec.sport.hockey	0.91	0.96	0.94
sci.crypt	0.83	0.93	0.88
sci.electronics	0.74	0.63	0.68
sci.med	0.89	0.76	0.82
sci.space	0.66	0.95	0.77
soc.religion.christian	0.66	0.95	0.77
talk.politics.guns	0.63	0.92	0.75
talk.politics.mideast	0.92	0.90	0.91
talk.politics.misc	0.94	0.49	0.65
talk.religion.misc	0.92	0.24	0.37

Table 31: Classification of category stemming using nb with tf and fs ch2

	precision	recall	f1-score
alt.atheism	0.63	0.63	0.63
comp.graphics	0.72	0.69	0.70
comp.os.ms-windows.misc	0.68	0.70	0.69
comp.sys.ibm.pc-hardware	0.76	0.76	0.76
comp.sys.mac-hardware	0.80	0.77	0.78
comp.windows.x	0.82	0.71	0.76
misc.forsale	0.73	0.85	0.79
rec.autos	0.84	0.82	0.83
rec.motorcycles	0.87	0.92	0.90
rec.sport.baseball	0.89	0.84	0.86
rec.sport.hockey	0.82	0.98	0.89
sci.crypt	0.82	0.93	0.87
sci.electronics	0.73	0.52	0.61
sci.med	0.81	0.80	0.81
sci.space	0.79	0.92	0.85
soc.religion.christian	0.69	0.92	0.79
talk.politics.guns	0.67	0.89	0.76
talk.politics.mideast	0.89	0.85	0.87
talk.politics.misc	0.83	0.53	0.64
talk.religion.misc	0.81	0.29	0.43

Table 32: Classification of category stemming using svm with tf and fs ch2

	precision	recall	f1-score
alt.atheism	0.70	0.43	0.53
comp.graphics	0.30	0.33	0.31
comp.os.ms-windows.misc	0.08	0.80	0.15
comp.sys.ibm.pc-hardware	0.61	0.26	0.36
comp.sys.mac-hardware	0.51	0.20	0.29
comp.windows.x	0.48	0.22	0.30
misc.forsale	0.33	0.45	0.38
rec.autos	0.72	0.27	0.39
rec.motorcycles	0.90	0.52	0.65
rec.sport.baseball	0.75	0.22	0.34
rec.sport.hockey	0.84	0.42	0.56
sci.crypt	0.91	0.39	0.55
sci.electronics	0.62	0.17	0.26
sci.med	0.68	0.18	0.29
sci.space	0.86	0.35	0.49
soc.religion.christian	0.61	0.34	0.44
talk.politics.guns	0.86	0.33	0.48
talk.politics.mideast	0.95	0.38	0.55
talk.politics.misc	0.63	0.35	0.35
talk.religion.misc	0.68	0.23	0.35

Table 33: Classification of category stemming using knn with tf and fs ch2

	precision	recall	f1-score
alt.atheism	0.73	0.80	0.76
comp.graphics	0.55	0.80	0.65
comp.os.ms-windows.misc	0.20	0.00	0.01
comp.sys.ibm.pc-hardware	0.52	0.72	0.61
comp.sys.mac-hardware	0.67	0.84	0.75
comp.windows.x	0.77	0.74	0.75
misc.forsale	0.79	0.79	0.79
rec.autos	0.84	0.91	0.87
rec.motorcycles	0.89	0.94	0.91
rec.sport.baseball	0.91	0.93	0.92
rec.sport.hockey	0.97	0.93	0.95
sci.crypt	0.89	0.90	0.89
sci.electronics	0.76	0.67	0.71
sci.med	0.90	0.80	0.84
sci.space	0.85	0.89	0.87
soc.religion.christian	0.85	0.86	0.86
talk.politics.guns	0.73	0.89	0.81
talk.politics.mideast	0.96	0.86	0.91
talk.politics.misc	0.71	0.61	0.66
talk.religion.misc	0.65	0.55	0.59

Table 34: Classification of category stemming using Nb with tf-idf and chi2

	precision	recall	f1-score
alt.atheism	0.80	0.82	0.80
comp.graphics	0.64	0.81	0.71
comp.os.ms-windows.misc	0.33	0.01	0.01
comp.sys.ibm.pc-hardware	0.53	0.80	0.64
comp.sys.mac-hardware	0.80	0.81	0.81
comp.windows.x	0.70	0.82	0.75
misc.forsale	0.85	0.74	0.79
rec.autos	0.87	0.92	0.90
rec.motorcycles	0.94	0.95	0.94
rec.sport.baseball	0.96	0.93	0.94
rec.sport.hockey	0.94	0.97	0.96
sci.crypt	0.82	0.94	0.88
sci.electronics	0.77	0.72	0.74
sci.med	0.88	0.85	0.87
sci.space	0.85	0.92	0.88
soc.religion.christian	0.83	0.95	0.88
talk.politics.guns	0.73	0.91	0.81
talk.politics.mideast	0.96	0.91	0.93
talk.politics.misc	0.73	0.63	0.68
talk.religion.misc	0.81	0.50	0.62

Table 35: Classification of category stemming using Nb with tf-idf and f- classif

	precision	recall	f1-score
alt.atheism	0.70	0.71	0.71
comp.graphics	0.79	0.73	0.76
comp.os.ms-windows.misc	0.75	0.74	0.74
comp.sys.ibm.pc-hardware	0.72	0.67	0.70
comp.sys.mac-hardware	0.80	0.82	0.81
comp.windows.x	0.88	0.78	0.82
misc.forsale	0.81	0.84	0.83
rec.autos	0.89	0.89	0.89
rec.motorcycles	0.91	0.97	0.94
rec.sport.baseball	0.91	0.91	0.91
rec.sport.hockey	0.86	0.99	0.92
sci.crypt	0.83	0.96	0.89
sci.electronics	0.81	0.63	0.71
sci.med	0.88	0.87	0.87
sci.space	0.85	0.95	0.90
soc.religion.christian	0.73	0.94	0.82
talk.politics.guns	0.69	0.93	0.79
talk.politics.mideast	0.92	0.91	0.92
talk.politics.misc	0.87	0.56	0.68
talk.religion.misc	0.80	0.36	0.50

Table 36: Classification of category stemming using svm and tf-idf

	precision	recall	f1-score
alt.atheism	0.66	0.71	0.69
comp.graphics	0.61	0.73	0.66
comp.os.ms-windows.misc	0.64	0.57	0.61
comp.sys.ibm.pc-hardware	0.58	0.61	0.60
comp.sys.mac-hardware	0.66	0.80	0.72
comp.windows.x	0.72	0.66	0.69
misc.forsale	0.75	0.78	0.77
rec.autos	0.86	0.80	0.83
rec.motorcycles	0.84	0.89	0.87
rec.sport.baseball	0.91	0.84	0.88
rec.sport.hockey	0.87	0.94	0.90
sci.crypt	0.87	0.87	0.87
sci.electronics	0.71	0.62	0.66
sci.med	0.87	0.75	0.80
sci.space	0.89	0.87	0.88
soc.religion.christian	0.74	0.86	0.79
talk.politics.guns	0.74	0.80	0.77
talk.politics.mideast	0.92	0.75	0.82
talk.politics.misc	0.71	0.58	0.64
talk.religion.misc	0.52	0.53	0.52

Table 37: Classification of category stemming using svm with tf-idf and ch2

	precision	recall	f1-score
alt.atheism	0.72	0.74	0.73
comp.graphics	0.71	0.70	0.71
comp.os.ms-windows.misc	0.70	0.57	0.63
comp.sys.ibm.pc-hardware	0.61	0.68	0.64
comp.sys.mac-hardware	0.69	0.83	0.75
comp.windows.x	0.83	0.72	0.77
misc.forsale	0.78	0.79	0.79
rec.autos	0.80	0.82	0.81
rec.motorcycles	0.86	0.92	0.89
rec.sport.baseball	0.92	0.86	0.89
rec.sport.hockey	0.88	0.93	0.91
sci.crypt	0.86	0.88	0.87
sci.electronics	0.62	0.72	0.67
sci.med	0.85	0.75	0.80
sci.space	0.84	0.90	0.87
soc.religion.christian	0.81	0.88	0.84
talk.politics.guns	0.74	0.84	0.79
talk.politics.mideast	0.92	0.82	0.86
talk.politics.misc	0.73	0.53	0.61
talk.religion.misc	0.59	0.49	0.53

Table 38: Classification of category stemming using svm with tf-idf and f-classif

	precision	recall	f1-score
alt.atheism	0.55	0.55	0.55
comp.graphics	0.16	0.59	0.25
comp.os.ms-windows.misc	0.25	0.53	0.34
comp.sys.ibm.pc-hardware	0.44	0.37	0.40
comp.sys.mac-hardware	0.37	0.37	0.37
comp.windows.x	0.56	0.28	0.38
misc.forsale	0.48	0.51	0.49
rec.autos	0.47	0.38	0.42
rec.motorcycles	0.75	0.52	0.61
rec.sport.baseball	0.46	0.48	0.47
rec.sport.hockey	0.67	0.50	0.575
sci.crypt	0.90	0.46	0.61
sci.electronics	0.41	0.25	0.31
sci.med	0.41	0.33	0.37
sci.space	0.80	0.51	0.63
soc.religion.christian	0.55	0.62	0.58
talk.politics.guns	0.77	0.35	0.48
talk.politics.mideast	0.50	0.56	0.53
talk.politics.misc	0.80	0.34	0.48
talk.religion.misc	0.54	0.25	0.34

Table 39: Classification of category stemming using knn with tf-idf and ch2

	precision	recall	f1-score
alt.atheism	0.52	0.54	0.53
comp.graphics	0.12	0.64	0.20
comp.os.ms-windows.misc	0.25	0.46	0.32
comp.sys.ibm.pc-hardware	0.44	0.31	0.36
comp.sys.mac-hardware	0.28	0.32	0.30
comp.windows.x	0.56	0.25	0.34
misc.forsale	0.51	0.29	0.37
rec.autos	0.47	0.42	0.44
rec.motorcycles	0.73	0.48	0.58
rec.sport.baseball	0.60	0.45	0.51
rec.sport.hockey	0.69	0.45	0.55
sci.crypt	0.94	0.48	0.63
sci.electronics	0.56	0.17	0.26
sci.med	0.68	0.22	0.33
sci.space	0.83	0.46	0.59
soc.religion.christian	0.33	0.70	0.45
talk.politics.guns	0.80	0.34	0.48
talk.politics.mideast	0.72	0.54	0.61
talk.politics.misc	0.66	0.27	0.39
talk.religion.misc	0.54	0.22	0.31

Table 40: Classification of category stemming using knn with tf-idf and f-classif

Bibliography

[1] Sariel Har-Peled, Dan Roth, Dav Zimak , Constraint Classification for Multiclass Classification and Ranking, Advances in Neural Information Processing Systems (NIPS) 2002.

[2] Melville, P., Gryc, W., and Lawrence, R.D.: 'Sentiment analysis of blogs by combining lexical knowledge with text classification'. Proc. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France 2009.

[3] Witten, I.H., Frank, E., and Hall, M.A.: 'Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques' (Elsevier Science, 2011. 2011)

[4] Chen, J., Huang, H., Tian, S., and Qu, Y.: 'Feature selection for text classification with Naïve Bayes', Expert Systems with Applications, 2009, 36, (3, Part 1), pp. 5432-5435

[5] William Cohen, Text Classification, Tutorial, CMU, CALD Summer Course.

[6] T. Kohonen, O. Simula, "Engineering Applications of the SelfOrganizing Map", Proceeding of the IEEE, Vol. 84, No. 10, 1996, pp.1354 – 1384

[7] Stephan Busemann, Sven Schmeier and Roman G. Arens (2000). Message classification in the call center. In Sergei Nirenburg, Douglas Appelt, Fabio Ciravegna and Robert Dale, eds., Proc. 6th Applied Natural Language Processing Conf. (ANLP'00), pp. 158-165, ACL.

[8] Santini, Marina; Rosso, Mark (2008), Testing a Genre-Enabled Application: A Preliminary Assessment (PDF), BCS IRSG Symposium: Future Directions in Information Access, London, UK, pp. 54–63

[9] $Saif, H.Fernandez, M.He, Alani$ on stop words filtring and data sparsity for sentiment analyses of twitters. In Proceedings of the Ninth International Conference on Language Resources and Evaluation $(LREC2014)$, Reykjavik, Iceland, 26-31 May 2014.

[10] Spirovski, K. Stevanoska, E. Kulakov, A. Popeska, Z. Velinov, G. Comparison of different model's performances in task of document classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018 p. 10

[11] Singh, J.; Gupta, V. Text stemming: Approaches, applications, and challenges. ACM Compu. Surv. (CSUR) 2016, 49, 45. [CrossRef]

[12] Singh, J.; Gupta, V. Text stemming: Approaches, applications, and challenges. ACM Compu. Surv. (CSUR) 2016, 49, 45. [CrossRef]

[13] https://www.machinelearningplus.com/nlp/lemmatization-examples-python/ consult on 15/05/2019 12:03

[14] https://www.twinword.com/blog/what-is-lemmatization/ consult on 15/05/2019 12:10

[15] https://phitchuria.wordpress.com/2018/09/27/python-nltk-stop-word-rare-word-removal/ consulter on 27/05/2019 11 :40

[16] Song and Yang, A comparative study on text representation schemes in text categorization, Journal of Pattern Analysis Application, Vol 8, 2005, pp 199 – 209.

[17] Porter, M.F. 1980. An algorithm for suffix stripping. Program, Vol. 14 (3), pp. 130 –137.

[18] Hotho, A., Nürnberger, A., and Paaß, G. 2005. A Brief Survey of Text Mining. Journal for Computational Linguistics and Language Technology. Vol. 20, pp. 19 – 62.

[19] Salton G. and Buckley C. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage., 24(4):513–523, 1988.

[20] Salton G. and McGill M. Introduction to Modern Information Retrieval. McGraw-Hill Book Company, New York, NY, 1983.

[21] @inproceedingsspirovski2018comparison, title=Comparison of different model's performances in task of document classification, author=Spirovski, Kristijan and Stevanoska, Evgenija and Kulakov, Andrea and Popeska, Zaneta and Velinov, Goran, booktitle=Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, pages=10, year=2018, organization=ACM

[22] @articlegomez2012highly, title=Highly discriminative statistical features for email classification, author=Gomez, Juan Carlos and Boiy, Erik and Moens, Marie-Francine, journal=Knowledge and information systems, volume=31, number=1, pages=23–53, year=2012, publisher=Springer

[23] Jiang, S.; Pang, G.;Wu, M.; Kuang, L. An improved K-nearest-neighbor algorithm for text categorization. Expert Syst. Appl. 2012, 39, 1503–1509

[24] Vapnik, V.; Chervonenkis, A.Y. A class of algorithms for pattern recognition learning. Avtomat. Telemekh 1964, 25, 937–945

[25] @articlekowsari2019text, title=Text classification algorithms: A survey, author=Kowsari, Kamran and Jafari Meimandi, Kiana and Heidarysafa, Mojtaba and Mendu, Sanjana and Barnes, Laura and Brown, Donald, journal=Information, volume=10, number=4, pages=150, year=2019, publisher=Multidisciplinary Digital Publishing Institute

[26] Mohri, M.; Rostamizadeh, A.; Talwalkar, A. Foundations of Machine Learning; MIT Press: Cambridge, MA, USA, 2012

[27] https://www.python.org/doc/essays/blurb/ consult on 11/06/2019 13:27

[28] http://jmlr.org/papers/v12/pedregosa11a.html consult on 11/06/2019 13:59

[29] [29] Aggarwal, C. C., and Zhai, C. (2012). A survey of text classification algorithms. In mining text data (pp. 163-222). Springer US..

[30] Korde, V.,and Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence and Applications, 3(2), 85.

[31] Mamoun, R., and Ahmed, M. A. (2014). A Comparative Study on Different Types of Approaches to the Arabic text classification. In Proceedings of the 1st International Conference of Recent Trends in Information and (Vol. 2, No. 3).

[32] Vala, M., and Gandhi, J (2015). Survey of Text Classification Technique and Compare Classifier

[33] James Joyce,'*BayesTheorem*' Stanford Encyclopedia of Philosophy,June 2003. https://plato.stanford.edu/entries/bayes-theorem/

[34] Wikipedia, the free Encyclopedia, Naive Bayes Classifier, 2008. http://en.wikipedia.org/wiki/Naive Bayesian classification

[35] Thesis on 'Clustering Approaches to Text Categorization' by Hiroya Takamura http://www.lr.pi.titech.ac.jp/ takamura/pubs/dthesis-original.pdf