Comparison of Text Categorization Accuracy with Different Document Representation

A Thesis Submitted

in Fulfillment of the Requirements

for the Degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Ishant Sharma 20120001
Karan Khanna 20125016
Monalisa Das 20124102
Vikas Saran 20122044
Oindrila Samanta 20124095

to the

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD

May, 2016

UNDERTAKING

We declare that the work presented in this thesis titled "*Comparison of Text Categorization Accuracy with Different Document Representation*", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

May, 2016
Allahabad

Ishant Sharma 20120001
Karan Khanna 20125016
Monalisa Das 20124102
Vikas Saran 20122044
Oindrila Samanta 20124095

CERTIFICATE

Certified that the work contained in the thesis titled "*Comparison of Text Categorization Accuracy with Different Document Representation*", by

Ishant Sharma 20120001
Karan Khanna 20125016
Monalisa Das 20124102
Vikas Saran 20122044
Oindrila Samanta 20124095

has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Ranvijay)
Computer Science and Engineering Dept.
M.N.N.I.T, Allahabad

May, 2016

PREFACE

With the digitization of text increasing enormously, the need to categorize and classify text has become indispensable. Disorganization and a bit of categorization and classification of text may reduce the response time of text or information retrieval. Therefore the task of organizing, categorizing and classifying texts and digitized documents as per the definitions suggested by text mining experts and computer scientists is of extreme importance. Text Categorization (TC), also known as Text Classification, is the task of classifying a set of text documents into different categories from a predefined set automatically. If a document belongs to exactly one of the categories, it is a single-label classification task; otherwise, it is a multi-label classification task. TC uses several tools from Information Retrieval (IR) and Machine Learning (ML). This paper discussed the different text categorization systems. These systems are using different classification algorithms for the classification of the documents.

ACKNOWLEDGEMENT

It is a great pleasure to thank the giants on whose shoulders we stand. First of all, we would like to thank our supervisor Dr. Ranvijay. This project would not have come into being without his kind guidance. He has been a great mentor and the best advisor we could ever have. His advice, encouragement and critique are sources of innovative ideas, inspiration, and causes behind this successful project. The confidence shown by him was the biggest source of inspiration for us. We thank our panel members Prof. M.M.Gore, Dr. Krishna K. Mishra, Dr. Pragya Dwivedi and Mrs. Rajitha B for their guidance, encouragement and valuable suggestions. We also convey our thanks to our Department Head, Prof. R.S.Yadav for providing excellent facilities and relaxations, without which this work was not easy.

Contents

Chapter 1

INTRODUCTION

The amount of data generated every minute online is huge. How to process tons of such text is a trending research topic. Text classification is one of the major problems among them. Text classification is the process of assigning a new document to an already known category. It is used as an essential task in many areas like information retrieval, email classification, spam detection, language classification etc. Most researchers in the recent past, have worked to find new text categorization algorithms in order to improve classification accuracy and reduce complexity. But very little research has been done in the area of document representation. Traditionally there are 3 models:

- vector space model
- probabilistic model
- inference network model

In this paper, we represent the documents in three forms: Bag of words(BOW) model , Term Frequency(TF) weighted model and Term frequency inverse document frequency(TFIDF) weighted model.

Six types of text categorization methods prevalent now-a-days include: (1) K-nearest neighbors (KNN) [4] [2]; (2) Naive Bayes [2] ; (3) Rocchio's method[5]; (4) SVM ; (5) Regression models; (6)Decision trees. In this paper, we use three major categorization algorithms - Naive Bayes, KNN and SVM on different document representation models and suggest the best combination among all.

1.1 Steps Of Text Categorization

1. **Document Preprocessing** - removing HTML tags, stop words, stemming, tokenization etc.
2. **Document Representation** - vector space model most common.
3. **Dimension Reduction** - Chi-square, Information Gain, mutual information etc. **Training** Generate model from training set.
4. **Testing and Evaluation** Testing set classified and accuracy evaluated.

1.2 Motivation

The rapid growth of World Wide Web has rendered the document classification by humans infeasible which has given impetus to the techniques like Data mining, NLP and Machine Learning for automatic classification of textual documents. With the high availability of information from diverse sources, classification tasks have attained paramount importance. Automated text classification has been considered as a vital method to manage and process a vast amount of documents in digital forms. This project provides an insight into text classification process, its phases and various classifiers.

1.3 Objective

This project aims at comparing and contrasting various available classifiers on the basis of classification accuracy. The ability to classify the text and predict the document it belongs to has been given in various research papers, but, we decided to effectively study the various classification techniques and suggest a technique which could lead to even more accurate results and with a relatively low complexity. We have tested our classifiers on 20 newsgroup dataset which is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. This improvement would surely help in the prediction of documents which would be of immense important in the forthcoming era of digitization.

Chapter 2

DESCRIPTION

In this chapter we describe about all the concepts used in this project.

2.1 Bag-of-words model

The **bag-of-words model** is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. Recently, the bag-of-words model has also been used for computer vision.The bag-of-words model is commonly used in methods of document classification, where the occurrence of each word is used as a feature for training a classifier.[8]

2.2 TF-IDF

TF-IDF, short for Term Frequency Inverse Document Frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.[8][7]

2.3 Naive bayes(NB)

In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. This is a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate preprocessing, it is competitive in this domain with more advanced methods including support vector machines.

2.4 Support Vector Machine (SVM)

In machine learning, support vector machines (SVMs, also support vector networks[2]) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

2.5 K- Nearest Neighbour (KNN)

In pattern recognition, the K-Nearest Neighbor algorithm (or K-NN for short) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether K-NN is used for classification or regression:

- In k-NN classification, the output is a class membership. An object is classified a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbour.

- In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbours

Chapter 3

HYPOTHESIS

3.1 Problem Description

In text categorization problem, we are given a training set D(Train) = (d(1) , 1), . . . , (d(n) , n) of labelled text documents where each document d(i) belongs to a document set D and the label of d(i) is within a predefined set of categories C = c 1 , . . . , c m . The goal in text categorization is to devise a learning algorithm that given the training set D(train) as input will generate a classifier (or a hypothesis) h : D C that will be able to accurately classify unseen documents from D.[3]

3.2 Proposed Solution

To classify a document, we have to first preprocess the document and then represent it in a way so that we can do statistical analysis on it easily. Document representation can be done in three ways: Bag of words (BOW) , Term frequency weighted matrix of document and TFIDF weighted matrix of document. Our aim is to represent documents in each of these forms, classify them using Naive Bayes, KNN and SVM classifiers, and then compare the accuracy of each to find the most suitable method for classification. We have tested our classifiers on 20 newsgroup[1] dataset, which is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.

3.2.1 Approach

The following steps are :

1. Preprocessing

   - **Reorganise Dataset into two categories -** Likes and Dislikes.
     Example: If we have taken 6 of the 20 newsgroups[1] as our dataset.
     Categories = [‘electronics‘, ‘crypt‘, ‘space‘, ‘hockey‘, ‘motorcycles‘ and ‘forsale‘] Then in this step, we will divide them into 2 categories - Likes and Dislikes.
     One such division can be like this.
     Likes = [‘hockey‘, ‘crypt, electronics]
     Dislikes = [‘motorcycles‘, ‘forsale, ‘space‘]
   - **Remove incompatible files -** Some files might be incompatible with utf-8. We will remove them in this step.
   - **Tokenization -** Convert data into list of strings.
   - **Remove stopwords -** Some words like a, an, the does not play any significant role in classification, these words are called stopwords. We remove such words from our data.
   - **Refine emails in data files** -If we closely look at the dataset, the headers of email has some unnecessary information. We delete this information in the header of emails .We deletes only lines in the email that starts with ’Path:’, ’Newsgroups:’, ’Xref:
     Example of an email Header:
     Path:
     cantaloupe.srv.cs.cmu.edu!rochester!udel!gatech!howland.reston.ans.net!usc!cs.utexas.edu!q
     t.cs.utexas.edu!news.Brown.EDU!noc.near.net!bigboote.WPI.EDU!bigwpi.WPI.EDU!kedz
     From: kedz@bigwpi.WPI.EDU (John Kedziora)
     Newsgroups: misc.forsale
     Subject: Motorcycle wanted.
     Date: 22 Feb 1993 14:22:51 GMT

Organization: Worcester Polytechnic Institute
Lines: 11
Expires: 5/1/93
Message-ID: ¡1manjrja0@bigboote.WPI.EDU¿
NNTP-Posting-Host: bigwpi.wpi.edu
Sender:
Followup-To:kedz@wpi.wpi.edu
Distribution: ne
Organization: Worcester Polytechnic Institute

2. Document Representation :

- **Bag of words (BOW) Boolean Representation -** In this representation, attributes are binary variables indicating the presence or absence of a term in the document (ignoring the number of occurrences) and are modeled by a discrete distribution. We form a matrix where rows indicate document and columns indicate features. Then we indicate the presence or absence of feature in a document by 1 or 0.
- **Term Frequency Representation -** In the term-count representation attributes are random variables taking values from the set of natural numbers (0, 1, 2 ...) and are modeled by the multinominal distribution. Term frequency of a term j in document i is the number of occurences of term j in document i divided by the total number of terms in document i.
- **TFIDF Representation -** In the TFIDF representation, document attributes are considered as normally distributed continuous variables. It puts weighting to a term based on its inverse document frequency. It means that if the more documents a term appears, the less important that term will be, and the weighting will be less. IDF for a feature is same for the overall dataset. It can be depicted as:

$$[\ TF - IDF = TF * IDF = Log(tf(i,j) + 1) * log(\frac{N+1}{n(j)}[6]) \qquad (1)$$

Here,

tf(i,j) is term frequency of term j in document i.

n (j) is the the no of ducuments that contain the term j.

N represents the total number of documents in the dataset.

We form a matrix where rows indicate document and columns indicate features. The matrix is filled with TF-IDF weights of features for each document.

3. Splitting dataset into training and testing set

   The dataset is split into training and testing set. We train the classifier with the training set and then predict the class of testing set using the trained classifier. The classifiers used are -

   •**NB**

   •**SVM**

   •**KNN**

4. Plot Results

   For each of the classifiers, we plot the result of classification accuracy where the train-test split is varied from 99% to 1% We have used matplotlib for plotting purpose. Its a python package.

5. Comparing accuracy of different classifiers

3.2.2 Details of Solution

1. **Project Metrics**

   - **Lines of Code(LOC)** : 600
   - **Hours of Work** : 6 Hours/Week

2. **Framework structure**

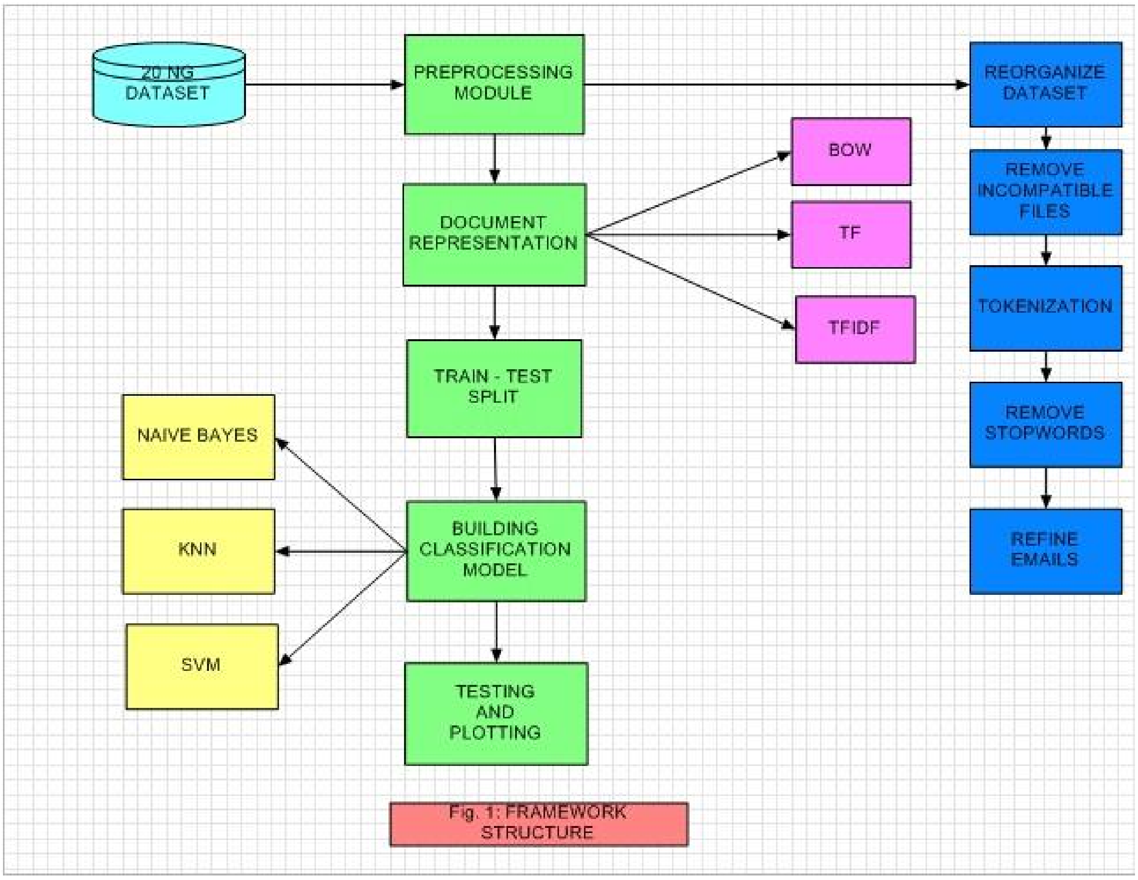

Figure 1: (Framework)

3. **Description**
   The various experiments have been performed by taking the following classification techniques into account, namely -

•**Nave Bayes**

•**Support Vector Machine**

•**K-Nearest Neighbour**

In each of the techniques, we used the following approach:

- The train-test split ratio has been varied from 99%to 1% for the given data.
- At each train-test split, precision is calculated for **Bag of Words, TF, and TFIDF** representation of data and is plotted.
- We have performed the experiments in 2 parts:
  - ∗ Using 6 of the 20 newsgroup(electronics', 'crypt', 'space', 'hockey', 'motorcycles' and 'forsale') categories.
  - ∗ Using the entire 20 newsgroup dataset.

The graphs corresponding to the various techniques, based on the conducted experiments are:

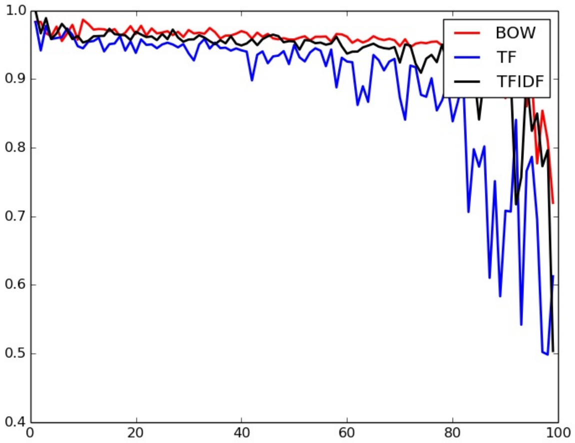

Figure 2: (y : Accuracy, x : Split ratio) [Naive Bayes]

This figure shows the accuracy of Naive Bayes classifier with three different document representations and train-test ratio ranging from 99% to 1%.

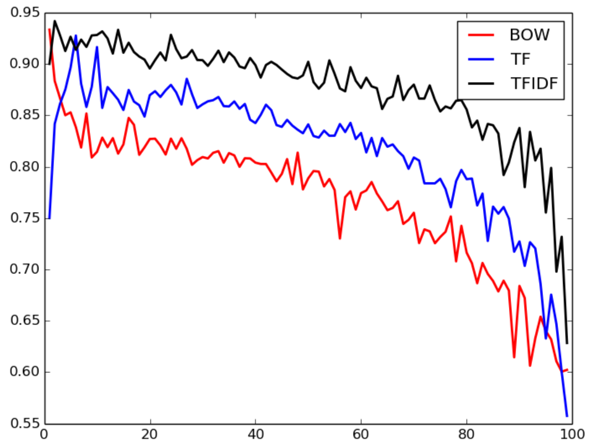

Figure 3: (y : Accuracy, x : Split ratio) [KNN]

This figure shows the accuracy of K-Nearest Neighbour classifier with three different document representations and train-test ratio ranging from 99% to 1%.

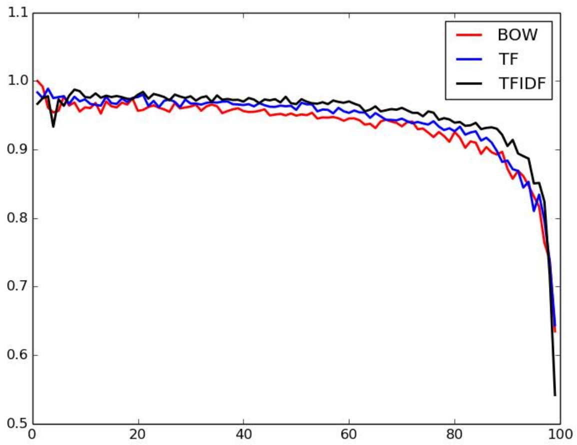

Figure 4: (y : Accuracy, x : Split ratio) [SVM]

This figure shows the accuracy of Support Vector Machine classifier with three different document representations and train-test ratio ranging from 99% to 1%.

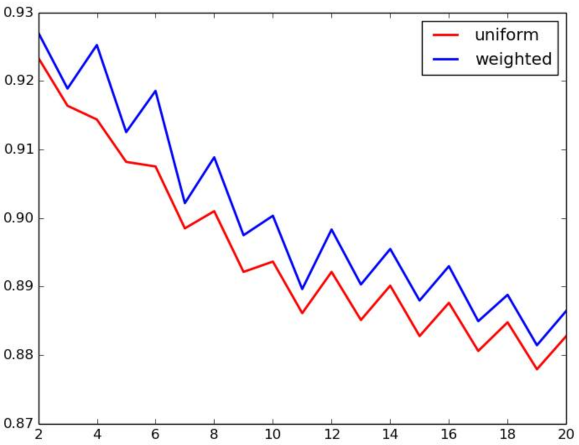

Figure 5: (y : Accuracy, x : Kth Nearest Neighbour) [KNN (Uniform and Weighted)]

This figure shows the accuracy of K-Nearest Neighbour with k ranging from 1 to 20 for uniform weighted and distance weighted nearest neighbour.

3.3 Result and Analysis

Test 1: BOW - NB - 20% test

We use a Bag of Words (BOW) representation of each document. And also a Naive Bayes (NB) classifier. The data is split, so that testing data is 20% of dataset.

	Precision	Recall	F1-score	Support
Dislikes	0.96	0.99	0.97	575
likes	0.99	0.95	0.97	621
Avg / Total	0.98	0.97	0.97	1196

Test 2: TF - NB - 20% test

	Precision	Recall	F1-score	Support
Dislikes	0.97	0.92	0.94	633
likes	0.91	0.97	0.94	563
Avg / Total	0.94	0.94	0.94	1196

Test 3: TFIDF - NB - 20% test

We use a TFIDF representation of each document. And also a Naive Bayes (NB) classifier. The data is split, so that testing data is 20% of dataset.

	Precision	Recall	F1-score	Support
Dislikes	0.96	0.95	0.95	584
likes	0.95	0.96	0.96	612
Avg / Total	0.95	0.95	0.95	1196

Test 4: TFIDF - SVM - 20% test

We use a TFIDF representation of each document. And also a linear Support Vector Machine (SVM) classifier. The data is split, so that testing data is 20% of dataset.

	Precision	Recall	F1-score	Support
Dislikes	0.96	0.97	0.97	587
likes	0.96	0.97	0.97	587
Avg / Total	0.97	0.97	0.97	1196

Test 5: TFIDF - SVM - 90% test

We use a TFIDF representation of each document. And also a linear Support Vector Machine (SVM) classifier.The data is split, so that testing data is 90% of dataset.

	Precision	Recall	F1-score	Support
Dislikes	0.90	0.95	0.93	2689
likes	0.95	0.90	0.92	2693
Avg / Total	0.92	0.92	0.92	5382

Test 6: TFIDF - SVM - KFOLD - 20 classes

We use a TFIDF representation of each document. And also a linear Support Vector Machine (SVM) classifier.

We split the data using Stratified K-Fold algorithm with k = 5.

Mean accuracy: 0.893 (+/- 0.003 std).

Test 7: TFIDF - 5-NN - Distance Weights - 20% test

In this experiment we use a TFIDF representation of each document. And also a K Nearest Neighbors (KNN) classifier with k = 5 and distance weights.

We split the data using Stratified K-Fold algorithm with k = 5.

	Precision	Recall	F1-score	Support
Dislikes	0.93	0.88	0.90	608
likes	0.88	0.93	0.90	588
Avg / Total	0.90	0.90	0.90	1196

Test 8: TFIDF - 5-NN - Uniform Weights - 20% test

We use a TFIDF representation of each document. And also a K Nearest Neighbors (KNN) classifier with k = 5 and uniform weights. The data is split using Stratified K-Fold algorithm with k = 5.

	Precision	Recall	F1-score	Support
Dislikes	0.95	0.90	0.92	581
likes	0.91	0.95	0.93	615
Avg / Total	0.93	0.93	0.93	1196

Test 9: TFIDF - 5-NN - Distance Weights - KFOLD

We use a TFIDF representation of each document. And also a K Nearest Neighbors (KNN) classifier with k = 5 and distance weights.

The data is split using Stratified K-Fold algorithm with k = 5.

Mean accuracy: 0.909 (+/- 0.004 std)

Test 10: TFIDF - 5-NN - Distance Weights - KFOLD - 20 classes

In this experiment we use a TFIDF representation of each document. And also a K Nearest Neighbors (KNN) classifier with k = 5 and distance weights. We split the data using Stratified K-Fold algorithm with k = 5.

We also use the whole ”Twenty Newsgroups” dataset, which has 20 classes.

Mean accuracy: 0.756 (+/- 0.003 std)

Chapter 4

CONCLUSION AND FUTURE WORK

The tests show that text classification can be effectively done by simple tools like **TFIDF** [7] and **SVM**. We have found that **TFIDF** [7] with **SVM** have the best performance. TFIDF with SVM perform well both for 2-class problem and 20-class problem.

As a future work, we need to research on regression models used in text classification. Our future work also includes incorporating background knowledge about words, like using WordNet for their synonyms and finding relationships between words to improve accuracy. There are many words which have more than one synonyms having different meaning. We will work on building a disambiguation strategy to identify proper synonym of any word.

References

[1] 20 newsgroup dataset. `http://qwone.com/jason/20Newsgroups/`.

[2] Text categorization algorithms. `http://answers.google.com/answers/main?cmd=threadviewid=225316`.

[3] BEKKERMAN, R. Using bigrams in text categorization. 01–10.

[4] Bruno Trstenjak, D. D., and Mikac, S. Knn with tf-idf based framework for text categorization.

[5] Joachims, T. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization.

[6] Liu, M., and Yang, J. An improvement of tfidf weighting in text categorization.

[7] Russell, M. A. *Mining the Social Web: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites.* O'Reilly Media; Second Edition, October 22, 2013.

[8] Singhal, A. *Modern Information Retrieval: A Brief Overview. IEEE.* 2001.