# A Comparative Approach of Dimensionality Reduction Techniques in Text Classification

Shaik Rahamat Basha
Department of Computer Science & Technology
Sri Krishnadevaraya University, India

J. Keziya Rani
Department of Computer Science & Technology
Sri Krishnadevaraya University, India

*Abstract*—**This work deals with document classification. It is a supervised learning method (it needs a labeled document set for training and a test set of documents to be classified). The procedure of document categorization includes a sequence of steps consisting of text preprocessing, feature extraction, and classification. In this work, a self-made data set was used to train the classifiers in every experiment. This work compares the accuracy, average precision, precision, and recall with or without combinations of some feature selection techniques and two classifiers (KNN and Naive Bayes). The results concluded that the Naive Bayes classifier performed better in many situations.**

***Keywords-stop word removal; stemming; feature weighting and selection; KNN; Naive Bayes***

## I. INTRODUCTION

In text classification, usually the dimensionality of the feature vector is huge because the input document consists of vast data and many terms [1, 2]. The major approaches for feature reduction are feature selection [2-8] and feature extraction [9, 10]. Feature extraction approaches are computationally more extensive and more effective than feature selection methods [9, 10]. Feature clustering is one effective technique in feature reduction, where similar features are grouped into one cluster and each cluster is treated as a feature [11, 12]. To reduce dimensionality severity in preprocessing, the unnecessary words which do not support the classification task (i.e. articles, verbs, prepositions etc.) are removed. For text categorization (by supervised learning procedure) labels are assigned for some documents from predefined categories (e.g. business, health, movies, etc.). The number of digital documents in the web is increasing, the number of terms (i.e. features) in those documents is quite large but only a few are informative. It is a severe problem which degrades the efficiency of Information Retrieval (IR) procedures.

The current work includes stemming process [13] which reduces the dimensionality of feature space and stochastic dependence between terms. A better feature selection procedure reflects the effectiveness on classification and computational efficiency. In this paper, feature weighting is presented along with the implementation procedure of different feature selection methods and KNN and Naïve Bayes classifiers [14, 15, 26-30]. Experiments are conducted and the results are analyzed.

## II. RELATED WORK

In feature selection approach the redundant and irrelevant features are removed from the corpus, e.g. selecting a subset of features from the training set and using that set as feature set for text classification. Some supervised feature selection approaches (IG, MI, OR, CHI, NGL, GSS etc.) [16] were used in our task. To reduce the noise of data with respect to term frequency (TF) [16], document frequency (DF) [16], is implemented by giving user input threshold and selecting the most probable features (by a threshold value k) and analyzing results with respect to dimensionality size. Rule-based classification is accurate if the rules are written by experts and are easily controlled if their number is small but if it increases or the rules conflict each other, rule maintenance becomes difficult. If the target domain changes the rules must be reconstructed. Machine learning-based approach is domain independent and gives high predictive performance, but training data are required [17].

## III. FEATURE WEIGHTENING

In this process, each feature (a single word or term or token) is assigned with a score based on a score computing function and the higher scored (weighted) terms are selected. Score computing functions include some mathematical definitions and probabilistic approaches which are estimated by some static information in the documents across different categories. Some example notations based on probabilities are:

- P(t): The probability of a document x containing the term t
- P(Ci): The probability of a document x belonging to the category Ci
- P(t, Ci): The probability of a document x containing the term t and belonging to the category Ci
- P(Ci/t): The probability of a document x belonging to the category Ci under the condition that it contains term t
- P(t/Ci): The probability of a document x containing the term t under the condition that it belongs to the category Ci

### A. Document Frequency (DF)

The number of documents in which a word occurs is *DF*:

$$DF = \sum_{i=1}^{m} A_i \qquad (1)$$

Corresponding author: S. Rahamat Basha (basha.ste@gmail.com) where $A_i$= document i where the word is present, *m*= number of documents, and *i* is an integer ranging from 1 to m.

The *DF* was computed for every unique term in the training corpus and the features with less *DF* than the predefined threshold were removed.

### *B. Mutual Information (MI)*

*MI* and *IG* give similar results for binary problems. The implemented multi class problem solving procedure was such that these two techniques give different results.

### *C. Chi Square*

It is a statistical measure used to measure the independence of a feature or a class. In this context, the null hypothesis here is that the particular word and category are completely independent, i.e. that the word is useless for classifying documents.

### *D. GSS Coefficient*

It is a simplified Chi Square function.

### *E. Odds Ratio (OR)*

This measure compares the odds of a word occurring in one class with the odds of occurring in another. *OR* is positive if the feature more often occurs in one document than the other, negative for vice versa and zero if the feature's presence is equal in both:

$$OR(F,C_k) = ln\frac{P(F|C_k)(1-P(F|\overline{C_k}))}{P(F|\overline{C_k})(1-P(F|C_K))} = ln\frac{\left(\frac{NF,C_k}{NC_k}\right)\left(1-\frac{NF,\overline{C_k}}{N\overline{C_k}}\right)}{\left(\frac{NF,\overline{C_k}}{N\overline{C_k}}\right)\left(1-\frac{NF,C_k}{NC_k}\right)} \quad (2)$$

### *F. NGL Coefficient*

It is a variant of the Chi Square metric, also called as correlation coefficient.

### *G. Information Gain (IG)*

This method is implemented on the constraints of class membership function (presence/absence) and by how much information is gained. The *IG* of a term t is given as:

$$IG(t) = H© - H\left(\frac{C}{T}\right) \text{ where}$$

$$T = \{present, absent\}\ and\ C = \{c+, c-\} \quad (3)$$

### *H. Relevancy Score*

$$w(t_{K,}c_i) = \ log[(P\ (t_K|c_i) +\ d)/(\ P\ (t_K|c_i) +\ d)] \ (4)$$

### *I. Multi-Set of Feature (MSF)*

$$m_{tk} = \frac{1}{|C|}\sum_C P(\ t_k|C \quad (5)$$

$$w(\mathrm{t_k}) = \sqrt{\frac{1}{(|\mathrm{C}|-1)\mathrm{m_{tk}}}\sum_{\mathrm{C}}(\mathrm{P}(\ \mathrm{t_k}|\mathrm{C}) - \mathrm{m_{tk}})^2} \quad (6)$$

### *J. KNN and Naive Bayes Classifiers*

The advantage of KNN in this model is that by choosing different constraints in every level of classification task we may compare the results with respect to $N$ ($N$ is variable) most matched values. This classifier is implemented by computing the Euclidean distance. The following is an illustration of the Naive Bayes classifier. Let *D* be a document set with 6 documents $d_1, d_2, \dots d_6$. The documents $d_1, d_2, \dots d_5$ are used to train the classifier and we are predicting the label of document $d_6$. The classifier is trained under the bag of words representation method. As shown in Table I the total unique words in the corpus are 14. According to the Bayes theorem [18, 19], Table II represents the likelihood/impact of each word of the test document to the classes Sports and Politics.

TABLE I. BAG OF WORDS REPRESENTATION OF D

| | **a** | **game** | **very** | **election** | **was** | **clean** | **close** | **over** | **but** | **forgettable** | **match** | **it** | **great** | **the** | **Label** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $d_1$ | 1 | 1 | | | | | | | | | | | 1 | | Sports |
| $d_2$ | | | | 1 | 1 | | | 1 | | | | | | 1 | Politics |
| $d_3$ | | | 1 | | | 1 | | | | | 1 | | | | Sports |
| $d_4$ | 1 | | | | | 1 | | | 1 | 1 | 1 | | | | Sports |
| $d_5$ | 1 | | | 1 | 1 | | 1 | | | | | 1 | | | Politics |
| $d_6$ | 1 | 1 | 1 | | | | 1 | | | | | | | | ? |

TABLE II. TEST DOCUMENT WORDS AND THEIR LIKELIHOOD/IMPACT FOR THE CATEGORIES SPORTS AND POLITICS

| **Words in d6** | P (word \| Sports) | P (word \| Politics) |
|---|---|---|
| **a** | $\frac{2+1}{11+14}$ | $\frac{1+1}{9+14}$ |
| **very** | $\frac{1+1}{11+14}$ | $\frac{0+1}{9+14}$ |
| **Close** | $\frac{0+1}{11+14}$ | $\frac{1+1}{9+14}$ |
| **game** | $\frac{2+1}{11+14}$ | $\frac{0+1}{9+14}$ |

The multiplication of all individual probabilities concludes to the label of $d_6$:

$$P(a|Sports)\ \times P(very|Sports) \times P(close|Sports) \times$$

$$P(game|Sports) \times P(Sports) =$$

$$= 2.76 \times 10^{-5} = 0.0000276 \quad (7)$$

$$P(a|NotSports)\ \times P(very|NotSports) \times$$

$$P(close|NotSports) \times P(game|NotSports) \times$$

$$P(NotSports) = 0.572 \times 10^{-5} = 0.00000572 \ \ (8)$$

According to the above illustration (i.e. 0.00000572 > 0.0000276 ) we can say that the test document *d6* more likely belongs to the class Sports.

## IV. RESULTS AND ANALYSIS

Two different datasets, described below, were used in the experiments.

### *A. Data Set 1: Self Made Data Set*

To train the classifiers, we used a small (of size around 1.5MB) self-made corpus. This allows the needed running time for training to be as short as possible. The documents of the self-made corpus were collected online articles from CNN, Washington Post, and New York Times. We collected 150 documents under the following categories: Business (23), Education (24), Health (30), Movies (10), Science (27), Sports (30), Travel (6), with an average of 702 words per document.

TABLE III. PREPROCESSING RESULTS OF SELF-MADE DATA SET

| Corpus | Stop-words | Stemming | # of processed terms | # of unique terms |
|---|---|---|---|---|
| **Self-made** | No | No | 105443 | 12819 |
| **Self-made** | Yes | No | 105443 | 12660 |
| **Self-made** | No | Porter | 105443 | 8878 |
| **Self-made** | Yes | Porter | 105443 | 8697 |
| **Self-made** | No | Lancaster | 105443 | 7490 |
| **Self-made** | Yes | Lancaster | 105443 | 7288 |

### *B. Data Set 2: The Reuters 21578 corpus*

We have used this corpus as test data. Reuters 21578 corpus consists of a total of 108 categories. The corpus is freely available in many internet sources [20-23]. By using the freely available tool SX, the 22SGML documents were converted to XML documents. Then some single characters were deleted which were rejected by the validating XML parser (e.g. decimal values below 30). The results shown below in the Tables are generated under the constraints where classifier is KNN, $N$=30, Threshold=1, Threshold step size=0.1, method of summation is sum and $DF$=<1, $TF$=<1 for the test documents Reut2-003.xml, Reut2.004.xml, and Reut2.005.xml. Here $TF$ stands for Term Frequency, i.e. the frequency of a term in a document. Often, Chi Square and MSF gave similar results and away from $IG$ and $DF$ values. When feature space size increased the similarity between Chi Square and MSF diminished.

TABLE IV. DIMENSIONALITY OF FEATURE SPACE=250

| | Chi Square | MSF | IG | DF |
|---|---|---|---|---|
| **Break-even-point** | 0.266 | 0.262 | 0.429 | 0.687 |
| **11 Point-precision** | 0.390 | 0.390 | 0.748 | 0.755 |
| **Average precision** | 0.345 | 0.335 | 0.777 | 0.774 |

TABLE V. DIMENSIONALITY OF FEATURE SPACE=500

| | Chi Square | MSF | IG | DF |
|---|---|---|---|---|
| **Break-even-point** | 0 | 0 | 0.429 | 0.687 |
| **11 Point-precision** | 0.143 | 0.143 | 0.831 | 0.837 |
| **Average precision** | 0.159 | 0.159 | 0.869 | 0.869 |

TABLE VI. DIMENSIONALITY OF FEATURE SPACE=750

| | Chi Square | MSF | IG | DF |
|---|---|---|---|---|
| **Break-even-point** | 0.0814 | 0 | 0.804 | 0.804 |
| **11 Point-precision** | 0.225 | 0.154 | 0.857 | 0.857 |
| **Average precision** | 0.1631 | 0.172 | 0.893 | 0.893 |

### *C. Classifiers Performance and Results*

For most results we may conclude that Naïve Bayes classifier performed better even though the performance of the two classifiers is efficient. In Figure 1 it can be seen that the KNN classifier gives different results for each feature selection technique and from Figure 2 that Naive Bayes classifier gives almost similar results for all feature selection techniques except for $IG$ and $MI$. We observe that KNN classifier gave similar results for MSF and Chi Square at feature space size=250 and 750, and for IG and DF at feature space size=500 and 750. The effectiveness of a classifier is not described by precision and recall, it is necessary to compute different evaluation metrics. The F-measure was computed, i.e. the harmonic mean of recall and precision. Micro-average F1 or micro average accuracy of F1 is calculated regardless of topics but macro-average F1 scores on all the topics [17, 24]. Average precision [25] is the average of the precisions at eleven evenly spaced recall levels. Break-even-point is the precision at the point where precision and recall are equal. The averaged value of the precision at the point where the recall equals the 11 values 0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0 is the 11-point precision [7, 25].

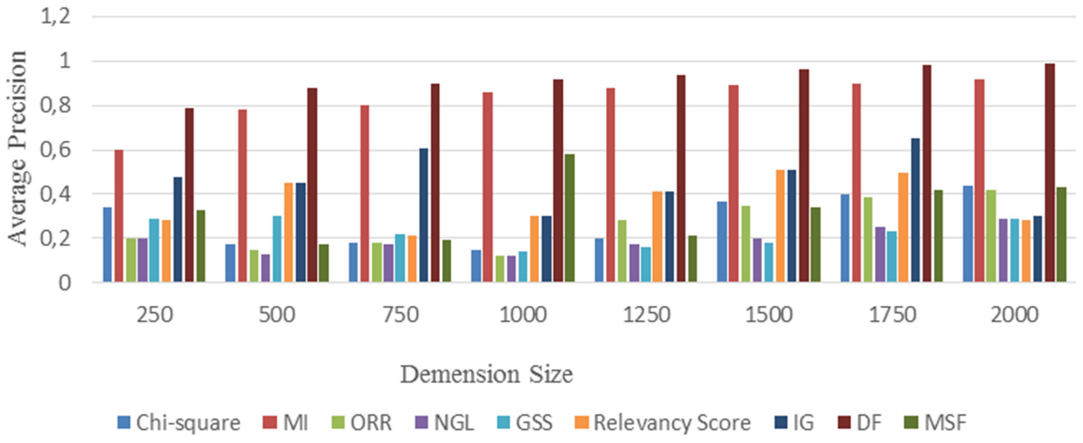

Fig. 1. KNN classifier results for the test document Reut.003.xml at constraints $k$=30, $TF$<1 and $DF$<1

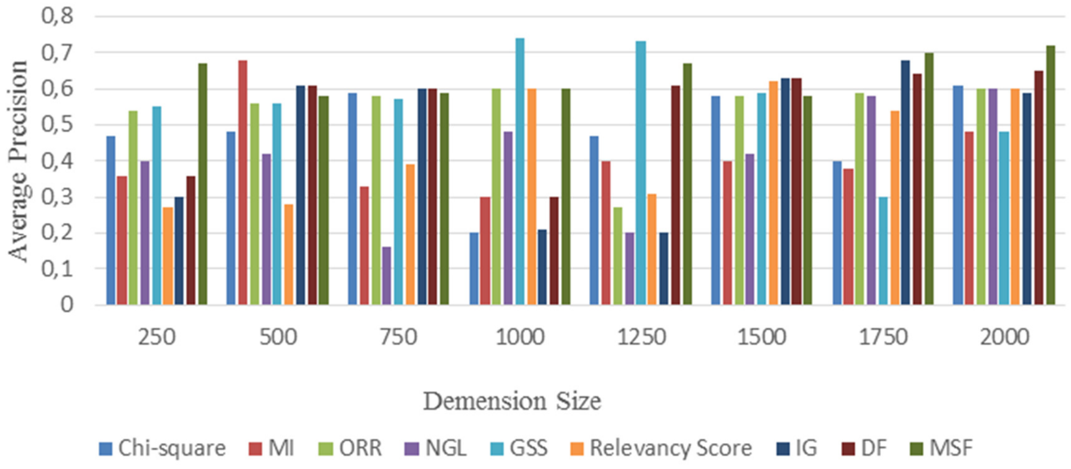

Fig. 2. Naive Bayes classifier results for the test document Reut.003.xml at constraints *k*=30, *TF*<1 and *DF*<1

TABLE VII. KNN CLASSIFIER RESULTS FOR THE TEST DOCUMENT REUT.003.XML AT *K*=30, *TF*<1 AND *DF*<1

| | **Chi Square** | **MI** | **ORR** | **NGL** | **GSS** | **Relevancy Score** | **IG** | **DF** | **MSF** |
|---|---|---|---|---|---|---|---|---|---|
| **BEP** | 0.266 | 0.515 | 0 | 0.170 | 0.170 | 0 | 0.729 | 0.687 | 0.262 |
| **Tolerance** | 0.233 | 0.702 | 0 | 0.162 | 0.162 | 0 | 0.0045 | 0.004 | 0.237 |
| **11 Point-precision** | 0.390 | 0.586 | 0.195 | 0.339 | 0.339 | 0.162 | 0.748 | 0.755 | 0.390 |
| **Avg-precision** | 0.345 | 0.593 | 0.218 | 0.282 | 0.282 | 0.173 | 0.777 | 0.774 | 0.335 |
| **Best category** | **Movies** | **Bus** | **Movies** | **Movies** | **Movies** | **Science** | **Bus** | **Bus** | **Movies** |

TABLE VIII. NAIVE BAYESCLASSIFIER RESULTS FOR THE TEST DOCUMENT REUT.003.XML AT *K*=30, *TF*<1 AND *DF*<1

| | **Chi Square** | **MI** | **ORR** | **NGL** | **GSS** | **Relevancy score** | **IG** | **DF** | **MSF** |
|---|---|---|---|---|---|---|---|---|---|
| **BEP** | 0.5791 | 0.335 | 0.5791 | 0.5791 | 0.5791 | 0.5791 | 0.190 | 0.221 | 0.5791 |
| **Tolerance** | 0.420 | 0.0250 | 0.420 | 0.420 | 0.420 | 0.420 | 0.142 | 0.173 | 0.420 |
| **11 Point-precision** | 0.540 | 0.385 | 0.540 | 0.540 | 0.540 | 0.540 | 0.339 | 0.387 | 0.540 |
| **Avg-precision** | 0.579 | 0.366 | 0.579 | 0.579 | 0.579 | 0.579 | 0.296 | 0.352 | 0.579 |
| **Best category** | **Science** | **Science** | **Science** | **Science** | **Science** | **Science** | **Science** | **Science** | **Science** |

Table VII and Table VIII describe the performance of KNN and Naive Bayes classifiers for the test document Reut.003.xml at dimension size=250. Each feature reduction technique gave a best category result for which the test document belongs.

TABLE IX. KNN RESULT FOR REUT.003.XML AT D=250

| | **Precision** | | **Recall** | |
|---|---|---|---|---|
| | **Micro** | **Macro** | **Micro** | **Macro** |
| **Chi square** | 1 | 1 | 0 | 0 |
| **MI** | 0.469 | 0.513 | 0.720 | 0.688 |
| **ORR** | 1 | 1 | 0 | 0 |
| **NGL** | 1 | 1 | 0 | 0 |
| **GSS** | 1 | 1 | 0 | 0 |
| **Relevancy Score** | 1 | 1 | 0 | 0 |
| **IG** | 0.308 | 0.349 | 0.936 | 0.905 |
| **DF** | 0.309 | 0.310 | 0.936 | 0.905 |
| **MSF** | 1 | 1 | 0 | 0 |

Tables IX and XIV describe the Precision and Recall values (micro and macro) given by the KNN and Naive Bayes classifiers respectively while using different feature reduction techniques. The performance of KNN at *k*=30, Threshold=1, Threshold step size=0.1, method of summation = sum, *DF*=<1, *TF*=<1 and Naive Bayes classifier at Threshold= 0.006666666666666667, Threshold step size= 0.0001, method of summation = sum, *DF*=<1, *TF*=<1for the test document Reut.003.xml at different dimension sizes while using different feature reduction techniques are given by Tables X-XIII, and XV-XVIII respectively.

TABLE X. KNN RESULT FOR REUT.003.XML 11 POINT PRECISION

| **Values** | **250** | **500** | **750** | **1000** | **1250** | **1500** | **1750** | **2000** |
|---|---|---|---|---|---|---|---|---|
| **Chi-square** | 0.4 | 0.15 | 0.14 | 0.12 | 0.18 | 0.38 | 0.4 | 0.45 |
| **MI** | 0.6 | 0.73 | 0.78 | 0.82 | 0.81 | 0.83 | 0.86 | 0.87 |
| **ORR** | 0.19 | 0.15 | 0.16 | 0.12 | 0.31 | 0.38 | 0.4 | 0.41 |
| **NGL** | 0.18 | 0.14 | 0.16 | 0.13 | 0.16 | 0.18 | 0.3 | 0.31 |
| **GSS** | 0.34 | 0.37 | 0.2 | 0.12 | 0.16 | 0.18 | 0.29 | 0.32 |
| **Relevancy score** | 0.18 | 0.14 | 0.28 | 0.35 | 0.4 | 0.51 | 0.5 | 0.54 |
| **IG** | 0.75 | 0.82 | 0.86 | 0.84 | 0.83 | 0.6 | 0.87 | 0.89 |
| **DF** | 0.76 | 0.82 | 0.83 | 0.82 | 0.82 | 0.9 | 0.88 | 0.88 |
| **MSF** | 0.4 | 0.14 | 0.15 | 0.42 | 0.18 | 0.37 | 0.42 | 0.43 |

TABLE XI. KNN RESULT FOR EEUT.003.XML BREAK-EVEN POINT

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **Chi-square** | 0.28 | 0 | 0 | 0.1 | 0.2 | 0.25 | 0.32 | 0.42 |
| **MI** | 0.51 | 0.7 | 0.75 | 0.79 | 0.78 | 0.8 | 0.81 | 0.82 |
| **ORR** | 0 | 0.1 | 0 | 0 | 0.19 | 0.29 | 0.33 | 0.41 |
| **NGL** | 0.18 | 0.2 | 0 | 0 | 0 | 0 | 0.22 | 0.28 |
| **GSS** | 0.18 | 0.28 | 0 | 0 | 0 | 0 | 0.19 | 0.29 |
| **Relevancy score** | 0 | 0 | 0.18 | 0.29 | 0.38 | 0.4 | 0.41 | 0.49 |
| **IG** | 0.72 | 0.78 | 0.8 | 0.79 | 0.8 | 0.61 | 0.63 | 0.84 |
| **DF** | 0.7 | 0.77 | 0.6 | 0.69 | 0.8 | 0.82 | 0.84 | 0.81 |
| **MSF** | 0.28 | 0 | 0 | 0.2 | 0 | 0.39 | 0.4 | 0.42 |

TABLE XII. KNN RESULT FOR REUT.003.XML MICRO PRECISION AND RECALL

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **MI(P)** | 0.42 | 0.62 | 0.65 | 0.7 | 0.8 | 0.82 | 0.83 | 0.84 |
| **IG(P)** | 0.25 | 0.42 | 0.5 | 0.45 | 0.61 | 1 | 0.63 | 0.65 |
| **DF(P)** | 0.24 | 0.43 | 0.55 | 0.49 | 0.65 | 0.71 | 0.82 | 0.95 |
| **MI(R)** | 0.66 | 0.68 | 0.69 | 0.66 | 0.65 | 0.65 | 0.64 | 0.63 |
| **IG(R)** | 0.3 | 0.42 | 0.45 | 0.5 | 0.62 | 0.4 | 0.65 | 0.66 |
| **DF(R)** | 0.87 | 0.89 | 0.87 | 0.88 | 0.86 | 0.85 | 0.84 | 0.84 |

TABLE XIII. KNN RESULT FOR REUT.003.XML MACRO PRECISION AND RECALL

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **MI(P)** | 0.51 | 0.72 | 0.8 | 0.84 | 0.9 | 0.91 | 0.92 | 0.97 |
| **IG(P)** | 0.28 | 0.45 | 0.58 | 0.6 | 0.65 | 1 | 0.65 | 0.66 |
| **DF(P)** | 0.3 | 0.24 | 0.25 | 0.6 | 0.7 | 0.52 | 0.68 | 0.78 |
| **MI(R)** | 0.7 | 0.72 | 0.73 | 0.72 | 0.73 | 0.7 | 0.71 | 0.69 |
| **IG(R)** | 0.76 | 0.78 | 0.87 | 0.89 | 0.84 | 0.6 | 0.82 | 0.91 |
| **DF(R)** | 0.86 | 0.9 | 0.96 | 0.87 | 0.83 | 0.93 | 0.84 | 0.81 |

TABLE XIV. NAIVE BAYES FOR REUT.003.XML AT D=250

| | Precision | | Recall | |
|---|---|---|---|---|
| | **Micro** | **Macro** | **Micro** | **Macro** |
| **Chi square** | 1 | 1 | 0 | 0 |
| **MI** | 0.345 | 0.62 | 0.048 | 0.042 |
| **ORR** | 1 | 1 | 0 | 0 |
| **NGL** | 1 | 1 | 0 | 0 |
| **GSS** | 1 | 1 | 0 | 0 |
| **Relevancy Score** | 1 | 1 | 0 | 0 |
| **IG** | 0.487 | 0.866 | 0.48 | 0.424 |
| **DF** | 0.487 | 0.866 | 0.048 | 0.042 |
| **MSF** | 1 | 1 | 0 | 0 |

TABLE XV. NAIVE BAYES FOR REUT.003.XML 11 POINT PRECISION

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **Chi-square** | 0.2 | 0.25 | 0.3 | 0.35 | 0.28 | 0.29 | 0.15 | 0.1 |
| **MI** | 0.39 | 0.38 | 0.32 | 0.33 | 0.42 | 0.43 | 0.5 | 0.5 |
| **ORR** | 0.53 | 0.54 | 0.55 | 0.45 | 0.4 | 0.61 | 0.32 | 0.63 |
| **NGL** | 0.1 | 0.2 | 0.25 | 0.25 | 0.3 | 0.29 | 0.4 | 0.2 |
| **GSS** | 0.51 | 0.52 | 0.53 | 0.43 | 0.45 | 0.47 | 0.52 | 0.53 |
| **Relevancy score** | 0.54 | 0.55 | 0.6 | 0.61 | 0.61 | 0.62 | 0.63 | 0.48 |
| **IG** | 0.35 | 0.61 | 0.61 | 0.61 | 0.62 | 0.63 | 0.64 | 0.65 |
| **DF** | 0.39 | 0.61 | 0.62 | 0.23 | 0.24 | 0.4 | 0.65 | 0.66 |
| **MSF** | 0.54 | 0.55 | 0.35 | 0.31 | 0.25 | 0.61 | 0.5 | 0.6 |

TABLE XVI. NAIVE BAYES FOR REUT.003.XML BREAK-EVEN POINT

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **Chi-square** | 0 | 0.1 | 0.18 | 0.2 | 0.29 | 0.3 | 0.37 | 0.4 |
| **MI** | 0.32 | 0.31 | 0.25 | 0.22 | 0.32 | 0.33 | 0.4 | 0.41 |
| **ORR** | 0.35 | 0.34 | 0.35 | 0.36 | 0.57 | 0.58 | 0.68 | 0.68 |
| **NGL** | 0.2 | 0.28 | 0.3 | 0.32 | 0.36 | 0.37 | 0.35 | 0.35 |
| **GSS** | 0.36 | 0.57 | 0.57 | 0.48 | 0.48 | 0.39 | 0.6 | 0.61 |
| **Relevancy score** | 0.54 | 0.55 | 0.66 | 0.68 | 0.69 | 0.69 | 0.6 | 0.54 |
| **IG** | 0.28 | 0.45 | 0.46 | 0.48 | 0.47 | 0.48 | 0.49 | 0.62 |
| **DF** | 0.22 | 0.56 | 0.57 | 0.58 | 0.59 | 0.6 | 0.61 | 0.63 |
| **MSF** | 0.46 | 0.45 | 0.48 | 0.49 | 0.6 | 0.71 | 0.72 | 0.74 |

TABLE XVII. NAIVE BAYES FOR REUT.003.XML MICRO PRECISION AND RECALL VALUES

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **MI(P)** | 0.3 | 0.62 | 0.63 | 0.38 | 0.6 | 0.51 | 1 | 0.98 |
| **IG(P)** | 0.85 | 0.87 | 0.85 | 0.87 | 0.87 | 0.98 | 1 | 1 |
| **DF(P)** | 0.45 | 1 | 1 | 0.6 | 1.1 | 1.1 | 1.1 | 1.1 |
| **MI(R)** | 0.09 | 0.1 | 0.09 | 0.07 | 0.2 | 0.1 | 0.12 | 0.14 |
| **IG(R)** | 0.45 | 0.1 | 0.1 | 0.14 | 0.12 | 0.3 | 0.19 | 0.19 |
| **DF(R)** | 0.2 | 0.4 | 0.3 | 0.3 | 0.4 | 0.2 | 0.2 | 0.1 |

TABLE XVIII. NAIVE BAYES FOR REUT.003.XML MACRO PRECISION AND RECALL

| Values | 250 | 500 | 750 | 1000 | 1250 | 1500 | 1750 | 2000 |
|---|---|---|---|---|---|---|---|---|
| **MI(P)** | 0.61 | 0.81 | 0.89 | 0.82 | 0.86 | 0.83 | 1 | 1.1 |
| **IG(P)** | 0.82 | 0.95 | 0.95 | 1 | 0.95 | 0.99 | 1 | 1.2 |
| **DF(P)** | 0.82 | 0.9 | 0.9 | 0.95 | 0.99 | 0.99 | 1 | 1.1 |
| **MI(R)** | 0.05 | 0.1 | 0.14 | 0.16 | 0.17 | 0.4 | 0.15 | 0.19 |
| **IG(R)** | 0.04 | 0.02 | 0.01 | 0.03 | 0.02 | 0.03 | 0.04 | 0.04 |
| **DF(R)** | 0.3 | 0.32 | 0.34 | 0.4 | 0.45 | 0.02 | 0.02 | 0.03 |

## V. CONCLUSION

The results shown in Figures 1-2 clearly show that KNN classifier performed well in classification with MI and DF but when other feature reduction techniques were used, the average precision was low. The Naive Bayes classifier performed well with the feature reduction techniques except MI. The higher average precision reported by KNN was with DF. The results reported in Tables X, XV, XI, and XVI reveal that Naive Bayes classifier worked well with respect to 11 Point Precision and Breakeven point. The results reported in Tables XII, XVII, XIII, and XVIII reveal that KNN classifier worked better than Naive Bayes with respect to micro and macro precision and recall. Figure 3 shows that Naive Bayes classifier works better than KNN with the measures Micro F1 and Macro F1.

The number of categories, the size of the class/corpus, the used feature selection techniques etc. were the key factors of the experimental results. Time complexity of the experiments was not considered/reported in this study. Generally, KNN classifier is simple to use and takes less time when compared with Naive Bayes but it is proved that Naive Bayes can work better in many cases.

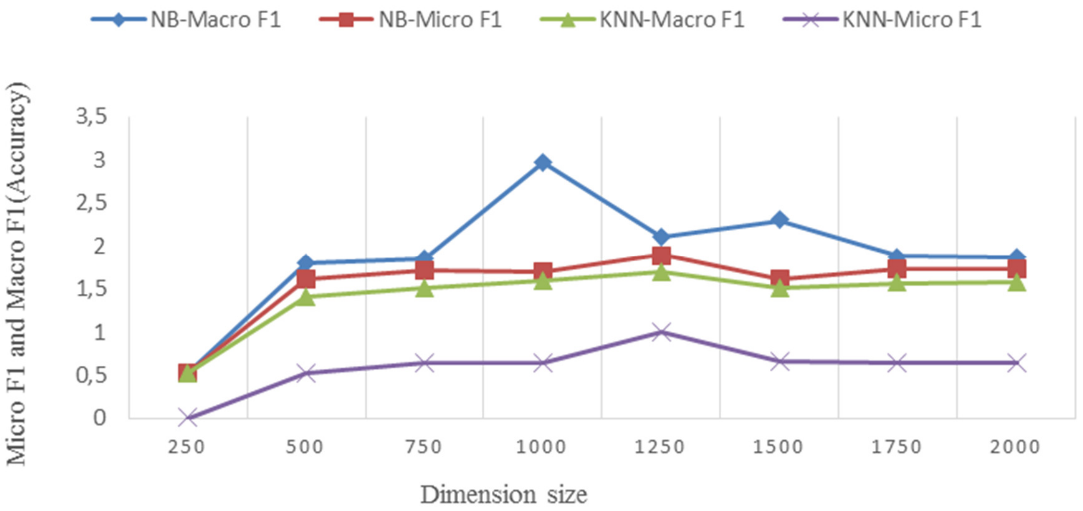

Fig. 3. Micro F1 and macro F1 values of KNN and Naïve Bayes classifiers For Reut.003.xml

## REFERENCES

[1] J. Y. Jiang, R. J. Liou, S. J. Lee, “A Fuzzy self-constructing feature clustering algorithm for text classification”, IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No. 3, pp. 335–349, 2011

[2] H. Kim, P. Howland, H. Park, “Dimension reduction in text classification with Support Vector Machines”, Journal of Machine Learning Research, Vol. 6, pp. 37-53, 2005

[3] A. L. Blum, P. Langley, “Selection of relevant features and examples in machine learning”, Artificial Intelligence, Vol. 97, No. 1-2, pp. 245-271, 1997

[4] E. F. Cambarro, E. Montanes, I. Diaz, J. Ranilla, R. Mones, “Introducing a family of linear measures for feature selection in text categorization”, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 9, pp. 1223-1232, 2005

[5] D. Koller, M. Sahami, “Toward optimal feature selection”, 13th International Conference on Machine Learning, Bari, Italy, July 3-6, 1996

[6] R. Kohavi, G. H. John, “Wrappers for feature subset selection”, Artificial Intelligence, Vol. 97, No. 1-2, pp. 273-324, 1997

[7] Y. Yang, J. O. Pederson, “A comparative study on Feature Selection in Text Categorization”, 14th International conference on Machine Learning, San Francisco, USA, July 8-12, 1997

[8] N. Slonim, N. Tishby, “The power of word clusters for Text Classification”, 23rd European Colloquium on Information Retrieval Research, 2001

[9] D. D. Lewis, “Feature selection and feature extraction for Text Categorization”, Workshop on Speech and Natural Language, New York, USA, February 23-26, 1992

[10] Y. Jan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi, Z. Chen, “Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing”, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 3, pp. 320-331, 2006

[11] M. C. Dalmau, O. W. Marquez Florez, “Experimental results of the signal processing approach to distributional clustering of terms on Reuters-21578 collection”, European Conference on Information Retrieval, Rome, Italy, April 2-5, 2007

[12] F. Sebastani, “Machine learning in automated text categorization”, ACM Computing Surveys, Vol. 34, No. 1, pp. 1-47, 2002

[13] M. F. Porter, “An algorithm for suffix stripping”, in: Readings in Information Retrieval, Morgan Kaufmann, 1997

[14] M. Alghobiri, “A comparative analysis of classification algorithms on diverse datasets”, Engineering, Technology & Applied Science Research, Vol. 8, No. 2, pp. 2790-2795, 2018

[15] E. Jamalian, R. Foukerdi, “A hybrid data mining method for customer churn prediction”, Engineering, Technology & Applied Science Research, Vol. 8, No. 3, pp. 2991-2997, 2018

[16] R. Neumayer, R. Mayer, K. Norvag, “Combination of Feature Selection Methods for Text Categorisation”, in: Lecture notes in computer science, Vol. 6611, Springer, 2009

[17] Y. Sasaki, Automatic Text Classification, Lecture notes, University of Manchester, available at: http://www.nactem.ac.uk/dtc/DTC-Sasaki.pdf 2008

[18] https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/

[19] https://en.wikipedia.org/wiki/Naive_Bayes_classifier

[20] https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[21] https://martin-thoma.com/nlp-reuters/

[22] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[23] http://disi.unitn.it/moschitti/corpora.htm

[24] A. Ozgur, L. Ozgur, T. Gungor, “Text Categorization with class-based and corpus-based keyword selection”, 20th International Symposium, Istanbul, Turkey, October 26-28, 2005

[25] R. Caruana, A. Niculescu-Mizil, “Data mining in metric space: an empirical analysis of supervised learning performance criteria”, KDD’04, Seattle, Washington, USA, August 22–25, 2004

[26] S. Rahamat Basha, J. Keziya Rani, J. J. C. Prasad Yadav, G. Ravi Kumar, “Impact of feature selection techniques in Text Classification: an experimental study”, J. Mech. Cont.& Math. Sci., Special Issue, No. 3, pp. 39-51, 2019

[27] G. Ravi Kumar, K. Nagamani, “A framework of dimensionality reduction utilizing PCA for neural network prediction”, International Conference on Data Science and Management, Bhubaneswar, USA, February 22-23

[28] G. Ravi Kumar, K. Nagamani, “Banknote authentication system utilizing deep neural network with PCA and LDA machine learning techniques”, International Journal of Recent Scientific Research, Vol. 9, No. 12, pp. 30036-30038, 2018

[29] M. V. Lakshmaiah, G. Ravi Kumar, G. Pakardin, “Framework for finding association rules in big data by using Hadoop Map/Reduce tool”, International Journal of Advance and Innovative Research, Vol. 2, No. 1(I), pp. 6-9, 2015

[30] G. Ravi Kumar, G. A. Ramachandra, K. Nagamani, “An efficient prediction of breast cancer data using data mining techniques”, International Journal of Innovations in Engineering and Technology, Vol. 2, No. 4, pp. 139-144, 2013