U.P.B. Sci. Bull., Series C, Vol. 77, Iss. 2, 2015 ISSN 2286-3540

CLASS-BASED AGGRESSIVE FEATURE SELECTION FOR POLYNOMIAL NETWORKS TEXT CLASSIFIERS – AN EMPIRICAL STUDY

Mayy AL-TAHRAWI[1]

Feature Selection (FS) is a crucial preprocessing step in Text Classification (TC) systems. FS can be either Class-Based or Corpus-Based. Polynomial Network (PN) classifiers have proved recently to be competitive in TC using a very small subset of corpora features. This paper presents an empirical study of the performance of PN classifiers using Aggressive Class-Based FS. Seven of the state-of-the art FS metrics are experimented and compared: Chi Square (CHI), Information Gain (IG), Odds Ratio (OR), GSS, NGL coefficient, Document Frequency (DF), and Gain Ratio (GR).The study is conducted on the Reuters Benchmark Corpus. Experimental results are presented in terms of both micro-averaged and macro-averaged precision, recall and F measures. Results reveal that aggressive Class-Based Chi-Square and DF metrics work best for Reuters using PN classifiers compared to the other five FS metrics experimented in this research.

Keywords: Polynomial Networks, Class-Based Feature Selection, Aggressive Feature Selection, Text Classification, Document Classification, Text Categorization.

1. Introduction

Text Classification (TC) is a learning task, where one or more pre-defined class labels are assigned automatically to documents based on the likelihood suggested by a training set of labeled documents. The task of TC is gaining more importance every day, due to the massive amount of online texts which need automatic classification, such as the World Wide Web documents, digital libraries and email.

A major obstacle in TC is the high dimensionality of the feature space. A medium-sized text collection may consist of hundreds of thousands of unique features (terms). A big part of a document features does not help much in identifying the document topic; in contrast, it adds noise and misleads the classification process. Furthermore, the TC algorithm itself may worsen the high dimensionality to a great extent.

As a result, Feature Selection (FS) has become a crucial preprocessing step in

[1] Senior Assistant Professor, Computer Science Department, Faculty of Information Technology, Al Ahliyya Amman University, Amman, Jordan, email: mayy.tahrawi@gmail.com

constructing (TC) systems. In this step, only the most relevant features are considered in classification. Different FS methods are compared and evaluated in the reduction of a high dimensional feature space in the literature of TC. The authors in [1]reported that FS can effectively remove 50% - 90% of the features while maintaining the classification accuracy.

FS can be either Corpus-Based or Class-Based. In the Corpus-Based approach, a single subset of features is selected for all classes; these features are selected from the corpus-top scoring features. On the other hand, in the Class-based approach, a distinct set of features is selected for each class. With a small number of features, the Class-Based approach achieves much more success by finding more crucial class features. Class-based FS outperformed Corpus-based approach on Reuters using a small number of features in many researches [2-5].

Many TC algorithms such as Support Vector Machines (SVM) [6], k-nearest neighbor (kNN) [7], Neural Networks [8], Naive Bayes (NB) [9], Linear Least SquaresFit, Logistic Regression (LR) [10], Radial Basis Function networks (RBF) [11] and Polynomial Networks (PNs) [2-4] have been applied to TC. A comparison of a set of these techniques is presented in [2- 4, 12].

Polynomial Networks (PNs) have been recently used for TC, and have proved to be competitive to the top performers in this field [2, 4]. PN classifiers demand a lot of memory resources; as a result, they depend largely on aggressive FS. Nevertheless, PN classifiers have proved to be efficient text classifiers using only very small part of a corpus features. The authors in [2, 3] have shown that PN classifiers have recorded competitive results on Reuters and 20Newsgroups using only 0.25%-0.5% of the corpora features.

Unlike the previous studies on PNs that focus on one FS metric [2-4], the performance of seven of the state-of-the-art FS metrics is investigated in this research: Chi Square (CHI), Information Gain (IG), Odds Ratio (OR), GSS, NGL coefficient, Document Frequency (DF), and Gain Ratio (GR). The benchmark Reuters Corpus is used, and the Class-Based aggressive FS approach is applied to select features using each of the FS metrics. Then, the PN classifier performance is evaluated using each of the seven resulting feature subsets independently, and results are presented, compared and analyzed.

The rest of the paper is organized as follows: Section 2 describes the architecture of PN classifiers, the Dataset is presented in Section 3, and Section 4 is devoted for explaining the FS metrics used in this research. Experiments and results are presented in Section 5 and analysis of these results takes place in Section 6. Finally, conclusions are presented in Section 7.

2. Polynomial Networks (PNs)

Polynomial networks, unlike artificial neural networks, have neither biological inspiration nor interpretation, but rather draw on traditional mathematical methods and evolutionary programming concepts to evolve a network of polynomial functions capable of approximating any continuous multivariate function from a collection of input-output data.

PNs were first used for TC in 2008 [2].They were not used earlier in TC, as the requirements of PN techniques grow exponentially with model complexity, and the number of features used. Nevertheless, the authors in [2- 4] have proved that PNs are competitive text classifiers to the state-of-the-art ones, if a properly-selected subset of the corpus feature set is used in classification.

2.1 Architecture of PN Classifier

The PN model adopted for TC consists of two layers. The first layer expands the input data into a high dimensional space by forming the monomial basis terms of the input vector $x\ (x_1, x_2, \ldots, x_N)$, such as $1, x_1, x_2, x_1^2, \ldots$ etc., where N is the number of features (dimensions) of x. Then, the second layer linearly separates data by combining the output of the first layer. The basic embodiment of a K^{th} order PN consists of several parts. The N features of one observation $x(x_1, x_2, \ldots, x_N)$ are used to form a basis function $p(x)$; one $p(x)$ is formed for each observation. The elements of $p(x)$ for a polynomial of degree K are monomials of the form [13]:

$$\prod_{j=1}^{N} x_j^{k_j}, \; where \quad k_j \geq 0 \quad and \quad 0 \leq \sum_{j=1}^{N} k_j \leq K \qquad (1)$$

The second layer of the PN linearly combines all inputs to produce weights (models) of classes. The whole class is represented by one weight, which is computed during the training phase.

2.2 The Training Phase

A PN is trained to approximate an ideal output using mean squared error as the objective criterion. The polynomial expansion of the i^{th} class feature vectors (observations) is denoted by [14]:

$$M_i = [\, p(x_{i,1}) \;\; p(x_{i,2}) \qquad p(x_{i,3}) \ldots \qquad p(x_{i,Ni})]^{\,t} \qquad (2)$$

where N_i is the number of training term vectors for class i, and $p(x_{i,m})$ is the basis function of the m^{th} term vector for class i. After forming M_i for each class i of the nc training classes, a global matrix M is obtained for the nc classes, by concatenating the individual M_i's computed for each class [13]:

$$M = [M_1 \qquad M_2 \qquad M_3 \;\ldots\; M_{nc}\,]^{\,t} \qquad (3)$$

Now, the training problem reduces to finding an optimum set of weights w (one per class) that minimizes the distance between the ideal outputs and a linear combination of the polynomial expansion of the training data such that [13]:

$$w_i^{opt} = \underset{w}{argmin} \left\| Mw - o_i \right\|_2 \qquad (4)$$

where o_i is the ideal output (a column vector which contains N_i ones in the rows where the i^{th} class' data are located in M, and contains zeros otherwise). A class model w_i^{opt} can be obtained in one shot (non-iteratively) by applying the normal equations method [13, 15]:

$$M^t M w_i^{opt} = M^t o_i \qquad (5)$$

Finally, Equation (5) reduces to

$$w_i^{opt} = (M^t M)^{-1} M^t o_i \qquad (6)$$

2.3 Recognition

Recognition of an unseen input consists of identification and verification. Identification involves finding the best matching class of an unseen input, given the feature vector of this input. In the verification phase, the claim made in the identification phase is either accepted or rejected. The identification phase proceeds as follows in the PN technique. The feature vector x of the new unseen input is expanded into its polynomial terms $p(x)$ using the same polynomial degree used with the training inputs in the training phase. Then, the new unseen input is assigned to the class c such that [13]:

$$c = \underset{i}{argmax}\, w_i^{opt} \bullet p(x) \;\; for\ i = 1,2,\ldots,nc \qquad (7)$$

In the verification phase, classifications with scores above a threshold are accepted, otherwise they are rejected.

2.4 Text Categorization (TC) using PNs

The training phase of TC using PNs goes through the following steps. Each training document is represented by a vector of features x using the vector space model. Features are represented using *binary* weight in this research. Then, the k^{th} order PN basis function $p(x)$ is formed for each training document, as in Equation (1); second order PNs are used in the experiments presented in this paper. The polynomial expansion of the training documents of each class is then formed as in Equation (2). Then, the global matrix for all classes is obtained as in Equation (3), and the PN is trained to approximate an ideal output using mean-squared error as the objective criterion (Equation 4). Finally, the training phase ends with finding the optimum set of weights as in Equation (6).

To classify an unseen document, the feature vector x of the unseen document is expanded into its polynomial terms $p(x)$ as in Equation (1). Then, the new unseen document is assigned to class c as explained in Equation (7).

3. DataSet

The benchmark Reuters test collection is used in the experiments conducted in this research. R10, the set of the 10 classes with the highest number of positive training examples of the ModApte version of Reuters-21578 is selected. To use R10 in single-label TC, only documents with a single topic and the classes which still have at least one train and one test example are considered. As a result, the set of the 10 most frequent classes, R10 is reduced to 8 classes (R8). From R10 to R8, the classes *corn* and *wheat*, which are intimately related to the class *grain* disappeared and this last class lost many of its documents. R8 [8] is then used to test different aggressive Class-Based FS metrics using the PN Classifier.

The whole processing steps performed on R8 are as follows:

1) Only letters, hyphens ‘-‘ and underscores ‘_’ are kept; any other character is eliminated. Hyphens and underscores are kept in order to recognize compound nouns.
2) All letters are converted to lowercase.
3) White space (blanks, tabs and newlines) are replaced by single spaces.
4) The Porter Stemmer [17] is used, with the following modification: an ignore list of more than 1000 stop words is defined and used to reduce the number of terms in the dataset. Then, any remaining word after stemming which consists of just one character is removed.

Applying these processing steps, R8 ended up with the distribution of documents and features in the 8 classes as shown in Table 1.

Table 1

Distribution of documents and features among R8 classes

Class #	**Class**	**# train docs**	**# test docs**	**Total # docs**	**# features**
1	Acq	1596	696	2292	7323
2	Crude	253	121	374	2751
3	Earn	2840	1083	3923	7188
4	Grain	41	10	51	1038
5	Interest	190	81	271	1448
6	money-fx	206	87	293	1992
7	Ship	108	36	144	1676
8	Trade	251	75	326	2652
	Total	5485	2189	7674	13891 (after removing duplicates among classes)

The large variation between classes in the number of training and test documents, and in the number of features is clear from this table. For example, *earn* and *acq* are frequent classes; i.e. they have a large number of train and test documents, while *grain* is a rare class with very small number of train documents. So, R8 doesn’t have a uniform class distribution.

4. Feature Selection Metrics

An important phase in building TC systems is the dimensionality reduction phase. This phase is considered crucial in TC due to many reasons. Firstly, many learning methods do not scale well to high problem sizes; the so-called “curse of dimensionality” problem [18]. Secondly, dimensionality reduction reduces overfitting (the tendency of a classifier to perform better on the data it has been trained on than new unseen data). Usually, dimensionality reduction takes the form of feature selection FS: each feature (term) is scored by means of a scoring function that measures its strength or discriminating power, and only the highest scoring features are considered in building the TC system.

FS algorithms fall into one of three paradigms: the filter model [19-22], the wrapper model [23- 26], and the hybrid model [27-29].The filter methods rely on the training dataset to evaluate each feature independently with respect to the class labels in this dataset, compute features scores, and then select a feature subset from the top scoring features. On the other hand, wrapper methods use Artificial Intelligence search methods, such as genetic algorithms, simulated annealing, or greedy hill-climbing to search for the best feature subset of features, repeatedly evaluating different feature subsets via cross validation with a particular induction algorithm.

The wrapper model tends to be more computationally expensive than the filter model [26, 30]. The hybrid models try to maximize the goodness-of-fit of the model and minimize the number of input features at the same time. Filter methods are the simplest to implement and the most scalable; hence, they are used in this research.

FS can be implemented using one of two policies: Corpus-Based or Class-Based. In the Corpus-Based approach, the feature subset is selected from the corpus topmost scoring features regardless of the share of each class in this set. On the other hand, in the Class-Based approach, FS is performed separately for each class, then the individual class subsets are globalized into one set. The latter method ensures that the most discriminating features for each class are included in the final feature subset used for building a classifier.

Class-Based FS has been implemented in recent studies [2- 4, 31, 32]. The research work conducted in [2- 4] has proved that Class-Based FS outperforms

Corpus- Based FS using different learning algorithms and different term weighting schemes.

In fact, the Corpus-Based FS policy favors the prevailing classes and gives penalty to rare classes (those with small number of training documents in the corpus). On the other hand, the Class-Based FS gives equal weight to each class in the FS phase.

Since R8 is a simple dataset that doesn't have a uniform class distribution, Class-based FS is the right choice for this dataset.

Choosing the proper FS metric is a key issue in building efficient TC systems. In order to find the best FS method for R8 dataset using PN classifiers, nine filter FS methods were investigated: Chi Square (CHI), Information Gain (IG), Odds Ratio (OR), GSS, NGL coefficient, Document Frequency (DF), Gain Ratio (GR), Relevancy Score (RS) , and Mutual Information (MI). Selecting a small feature subset using RS and MI was not possible, as a large number of features shared the same high score, so these two metrics were excluded from the experiments, and the other seven FS metrics were kept. An overview of each of these seven FS metrics takes place in the following subsections.

4.1 Chi Square (CHI)

The Chi Square (CHI) FS metric measures the lack of independence between a feature and a class. It has shown to yield good results and has proved to maximize precision of classification, compared to other feature selection methods [2-4, 33-36]. On the other hand, CHI is known not to be reliable for low-frequency features [37]. Chi square score is measured for each feature t in each class c_i in the training set as follows [38]:

$$\chi^2 (t, c_i) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \qquad (8)$$

where:
N is the total number of training documents in the dataset,
A is the number of documents belonging to class c_i and containing t,
B is the number of documents belonging to class c_i but not containing t,
C is the number of documents not belonging to class c_i but containing t, and
D is the number of documents neither belonging to class c_i nor containing t.

The higher this score is, the more discriminating the feature is for that class. If a feature t and a class c are independent, the score produced by this formula is 0.

4.2 Information Gain (IG)

IG is an information-theoretic function which measures the number of bits of information obtained for class prediction given the information regarding the presence or absence of a feature in a document. It measures how much information a feature *t* contains about category *c_i*. The information gain globally determines the quality of a feature *t* with respect to all classification classes on average and can be computed as [34]:

$$IG = -\sum_{i=1}^{m} P(C_i) \log P(C_i) + P(t)\sum_{i=1}^{m} P(C_i \mid t) \log P(C_i \mid t) + P(\overline{t})\sum_{i=1}^{m} P(C_i \mid (\overline{t}) \log P(C_i \mid \overline{t}) \quad (9)$$

where *m* is the number of classes in the corpus. A higher IG score of a feature and class combination means that the feature is more informative about the class, and thus more predictive. IG has proved to perform well in comparison to other feature selection metrics in [34]. On the other hand, as mentioned by [39], IG grows with the increase of dependence between *t* and *c* and unfortunately also with the increase of the entropy of *t*. As a result, features with low entropy have lower IG evaluation although they might be strongly correlated with *c*.

4.3 Odds Ratio (OR)

Odds ratio (OR) was proposed originally by van Rijsbergen [40] for use in information retrieval to select features for relevance feedback. The main idea behind OR is that the distribution of features in the relevant documents is different from that in the non-relevant ones. Later on, OR was used to select features in TC by Mladenic [41]. OR between a feature *t* and a class c_i is defined as follows:

$$OR(t,c_i) = \frac{P(t \mid c_i)(1 - P(t \mid \bar{c}_i))}{(1 - P(t \mid c_i))P(t \mid \bar{c}_i)} \quad (10)$$

OR has been reported to perform well in combination with the Naive Bayes classifier in [41, 42].

4.4 GSS

GSS Coefficient is a simplified variant of the CHI statistics proposed by [43]. It is defined as follows:

$$GSS(t,c_i) =)\, P(t,c_i)\ P(\overline{t},(\overline{c}_i) - P(t,\overline{c}_i)\, P(\overline{t},c_i) \quad (11)$$

GSS has outperformed other FS metrics in some TC researches [44].

4.5 NGL coefficient

NGL of a class c_i and a feature t_k can be defined as follows [45]:

$$NGL(t_k,c_i) = \frac{\sqrt{N}\ (P(t_k,c_i)P(\bar{t}_k,\bar{c}_i) - P(t_k,\bar{c}_i)P((\bar{t}_k,c_i)}{\sqrt{(P(t_k)P(\bar{t}_k)P(c_i)P(\bar{c}_i)}} \qquad (12)$$

NGL can be viewed as a variant of CHI, where CHI (t_k, c_i) = NGL $(t_k, c_i)^2$.

4.6 Document Frequency (DF)

DF is one of the simplest FS metrics. It simply counts the number of documents in a certain class which contain a specific feature. However, many researchers observed DF to be an important metric for selecting informative features, because it favors common features, which seems to be a significant characteristic for TC [34, 36, 42].

4.7 Gain Ratio (GR)

GR is an entropy-normalized version of IG whose use as a FS metric was first proposed in [46]. GR can be computed as follows for *m* classes:

$$GR = \frac{IG}{-\sum_{i=1}^{m} P(c_i)\log P(c_i)} \qquad (13)$$

In opposition to IG, the GR favors variables with fewer values.

From the FS metrics described above, it can be noticed that some of these are one-sided; i.e. they select only the most membership indicative features (OR, DF). Thus, features from non-relevant documents are not considered to be useful. On the other hand, two-sided metrics (IG, CHI) distinguish between positive and negative features and combine them implicitly.

5. Experiments and Results

Each of the seven FS metrics explained above has been used in this research to compute features' strengths in each class. Then, 0.5% of each top-scoring features are selected, using the Class-Based approach, for building and testing the PN classifier. The share of each class in the final feature subset and the total number of features selected is presented in Table 2.

Table 2

The share of each class in the final feature subset

Class	**# features**
Acq	40
crude	15
Earn	36
grain	5
interest	8
money-fx	10
Ship	8
trade	13
Total number of features	**135**

After removing duplicate features among classes using each FS metric, the final number of features used by each FS metric is reduced from 135 to the number shown in Table 3.

Table 3

The final number of features used by each FS metric after removing duplicates

FS Metric	**# features after eliminating duplicates**
CHI	108
DF	98
GR	102
OR	135
NGL	126
IG	101
GSS	119

Regarding term weighting, the commonly used binary weighting is used in this research. In binary weighting, the weight of a feature is considered to be 1 if the feature appears in the document and it is considered to be 0 otherwise. Other various term weighting approaches are studied in the literature of TC [47]. PN classifiers have been tested on Reuters using different term weighting schemes in [3], and binary term weighting has achieved a competitive performance to the other weighting schemes in this research. Add to this, binary term weighting is very simple, and requires much less computational and storage resources.

The PN classifier is evaluated in each of the experiments using both the microaveraged and macroaveraged versions of Precision, Recall, and F1 measures. Precision is the percentage of documents classified into a class c_i that indeed belong to c_i, while Recall is the percentage of documents belonging to c_i that are indeed classified into c_i. Precision and Recall can be computed as follows:

$$Precision = \frac{TP_i}{TP_i + FP_i} \qquad (14)$$

$$Recall = \frac{TP_i}{TP_i + FN_i} \qquad (15)$$

where TP_i, FP_i, and FN_i are explained in Table 4.

Table 4

Class C_i		**Expert Judgement**	
		T	**F**
Classifier Judgement	**T**	TP_i	FP_i
	F	FN_i	TN_i

Since any classifier can be tuned to emphasize precision at the expense of recall, or vice versa, a more realistic measure is commonly used to evaluate classifiers; that is the F1 measure. The F1 measure, introduced by Van Rijsbergen [40], is the harmonic average of both precision and recall. High F1 means high overall performance of the system. F1 is computed as follows [48]:

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \qquad (16)$$

Individual results of the 8 classes are both microaveraged and macroaveraged to give an idea of the classification performance on the corpus as a whole. **Table 5** shows the mathematical definitions of precision, recall, and F1, in both their microaveraged and macroaveraged variants, where C refers to the number of classes in the corpus.

Table 5

	MicroAverage	MacroAverage
Precision	$\frac{\sum_{i=1}^{\lvert C\rvert} TP_i}{\sum_{i=1}^{\lvert C\rvert} TP_i + FP_i}$	$\frac{1}{\lvert C\rvert}\sum_{i=1}^{\lvert C\rvert}\frac{TP_i}{TP_i+FP_i}$
Recall	$\frac{\sum_{i=1}^{\lvert C\rvert} TP_i}{\sum_{i=1}^{\lvert C\rvert} TP_i + FN_i}$	$\frac{1}{\lvert C\rvert}\sum_{i=1}^{\lvert C\rvert}\frac{TP_i}{TP_i+FN_i}$
F-Measure	$\frac{2*\sum_{i=1}^{\lvert C\rvert} TP_i}{2*\sum_{i=1}^{\lvert C\rvert} TP_i + \sum_{i=1}^{\lvert C\rvert} FP_i + \sum_{i=1}^{\lvert C\rvert} FN_i}$	$\frac{1}{\lvert C\rvert}\sum_{i=1}^{\lvert C\rvert} F_i$

As is clear from the equations in Table 5, Microaveraged F-measure rewards classifiers that behave well on classes with many positive examples, as it gives equal weight to each document. On the other hand, Macroaveraged F-measure emphasizes classifiers that perform well also on infrequent classes, as it gives equal weight to each class regardless of its frequency. Both measurement scores are provided in this research to be more informative.

Results of PN classification using each FS metric is summarized in Table 6.

Table 6

Results of PN classification using each FS metric

	MicroAverage F-easure	MacroAverage F-measure
CHI-108	78.2549	55.0456
DF-98	67.7478	35.5463
GR-102	49.4746	8.2748
OR-135	31.7953	6.0312
NGL-126	31.7953	6.0312
IG-101	3.4262	0.82818
GSS-119	3.4262	0.82818

6. Analysis of Results

As is clear from the results summarized in Table 6, Fig. 1 and Fig. 2, CHI and DF FS metrics are the best performers for PNs when both micro- and macro-averaging results.

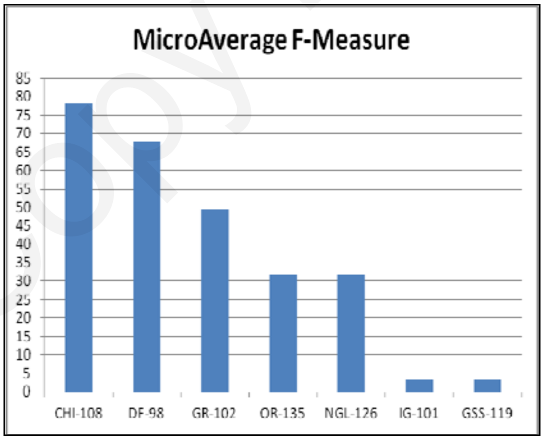

Fig. 1 MicroAverage F Results using each FS metric

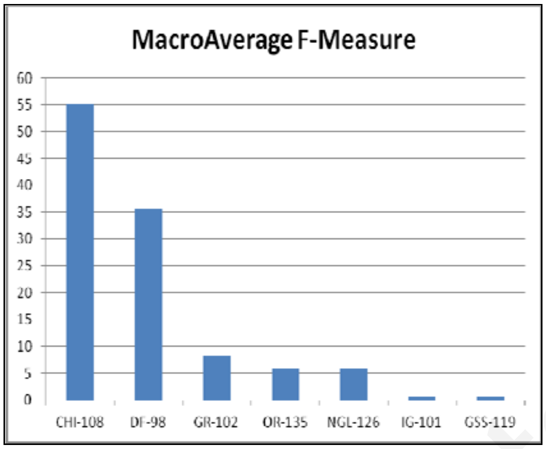

Fig. 2 MacroAverage F Results using each FS metric

CHI has recorded the best performance, although the number of features selected by this metric was not the largest among the other feature sets. CHI is known not to be reliable for low- frequency terms. Add to this, CHI has achieved an optimal performance as both a FS metric and as a weighting scheme in other researches on R8 [2-4]. The next top performer is DF with only 98 features; the smallest set of features in the experiments conducted in this research. The worst performance recorded in this research was using IG and GSS with 101 and 119 features respectively. These observations hold for both micro- F1 and macro-F1 as shown in Fig. 1 and Fig. 2. These results are consistent with the results in [36] where the top performers on Reuters were DF and CHI.

On average, the decrease in performance in going from CHI to IG is much sharper for macroaveraging than for microaveraging. This can be attributed to the fact that microaveraged effectiveness is dominated by the performance of the classifiers on the most frequent classes. In R8, classes that have the highest number of positive test examples are the same classes that have the highest number of positive training examples. These frequent classes contribute much more than the remaining classes in determining the microaveraged performance on R8.

The high skewness in the distribution of the classes in R8 affects the macro-averaged F-measure values in a negative way because macro-average gives equal weight to each class instead of each document and documents of rare classes tend to be more misclassified. By this way, the average of correct classifications of classes drops dramatically. Detailed performance results per class for the top performers are presented in Tables 7 and 8.

Table 7

Detailed performance results per class using CHI

Class	# train docs	# test docs	CHI		
			Precision	**Recall**	**F1**
Acq	1596	696	78.7115	80.7471	79.7163
crude	253	121	67.8161	48.7603	56.7308
earn	2840	1083	84.7282	90.6741	87.6004
grain	41	10	100	40	57.1429
interest	190	81	46.875	55.5556	50.8475
money-fx	206	87	46.0526	40.2299	42.9448
ship	108	36	60	25	35.2941
trade	251	75	44.7368	22.6667	30.0885
Micro-Averaged Results			**78.2549**	**78.2549**	**78.2549**
Macro-Averaged Results			**66.115**	**50.4542**	**55.0456**

Table 8

Detailed performance results per class using DF

Class	# train docs	# test docs	DF		
			Precision	**Recall**	**F1**
acq	1596	696	69.1718	64.7989	66.9139
crude	253	121	44.898	36.3636	40.1826
earn	2840	1083	81.0406	84.8569	82.9048
grain	41	10	6.0606	20	9.3023
interest	190	81	33.6538	43.2099	37.8378
money-fx	206	87	22.549	26.4368	24.3386
ship	108	36	18.5185	13.8889	15.873
trade	251	75	10.2564	5.3333	7.0175
Micro-Averaged Results			**67.7478**	**67.7478**	**67.7478**
Macro-Averaged Results			**35.7686**	**36.861**	**35.5463**

Since microaveraged F-measure is the proportion of the correct positive classification decisions, it can be expected that most positive classification decisions taken concern classes that have many positive training and test examples; which are acq and earn classes in R8. As a result, the microaveraged performance obtained on R8 is heavily influenced by the performance obtained on the two most frequent classes (*acq, earn*), rather than by the performance obtained on the remaining 6 classes. This explains why the above-mentioned decrease in microaveraged F-measure is not very sharp. On the other hand, macroaveraged effectiveness is, by definition, not dominated by any class in particular. Because each of the 6 least frequent classes counts the same as any of the 2 most frequent ones, these 6 difficult classes result in a clear decrease in the macroaveraged performance of the PN classifier.

Results reached in this research are consistent with those reported in [34]. DF and CHI are strongly correlated; thus their performance is close. Both of these FS metrics assign a high evaluation to common features. CHI is known to be dominant for a small number of features; it selects only the most discriminative features. This results in a remarkable increase in the classification performance even for small number of features.

In spite of the simplicity of DF, it was the next top performer after CHI, which is very costly computation-wise.

IG does not perform very well because it favors common features occurring often among more categories. On the other hand, OR selects mainly rare features, which can achieve high classification performance for a larger number of features as reported in [49].

7. Conclusions

In TC, most of the learning takes place with a small but discriminative portion of features for a class. Class-based feature selection, by definition, focuses on this small portion. On the other hand, corpus-based approach finds general features concerning all classes. So, with aggressive FS, class-based approach achieves much more success by finding more crucial class features; Corpus-based approach will not succeed with that small portion.

In this paper, an empirical study of the performance of Polynomial Networks Text Classifiers using Class-Based FS is conducted. Unlike the previous studies on PNS that focus on one FS metric [2-4], the performance of seven FS metrics is investigated in this research. All experiments are conducted on the benchmark Reuters R8 data set, and features are weighted using binary term weighting. PN classifier performance is evaluated in terms of micro-averaged and macro-averaged precision, recall, and F-measure for each FS metric.

Unbalanced class distributions make the classification task difficult and an appropriate FS metric must be chosen carefully. From the results obtained in this research, it can be concluded that Class-Based aggressive CHI and DF FS can achieve high micro-averaged performance on Reuters using PN classifiers. Theoretical justifications are provided for these results.

It is worth noting that CHI and DF were the top performers although they had smaller number of features compared to the other FS metrics investigated this research. So, the FS metric and the corpus under consideration is much more important than just the number of features used in building a certain classifier.

REFERENCES

[1]. *Y. Yang and J. Pederson*, “A Comparative Study on Term Selection in Text Categorization”, in Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 412-420.

[2]. *M. M. AL-Tahrawi and R. Abu Zitar*, “Polynomial Net-works versus Other Techniques in Text Categorization” , International Journal of Pattern Recognition and Artificial Intelligence, **vol**. 22, no. 2, 2008, pp. 295-322. doi:10.1142/S0218001408006247

[3]. *M. M. AL-Tahrawi*, “The Role of Rare Terms in Enhancing the Performance of Polynomial Networks Based Text Categorization”, Journal of Intelligent Learning Systems and Applications, **vol.** 5, 2013, pp. 84-89. doi:10.4236/jilsa.2013.52009

[4]. *M. M. AL-Tahrawi*, “The Significance Of Low Frequent Terms In Text Classification”, International Journal of Intelligent Systems, vol. 29, pp. 389–406, 2014. "doi: 10.1002/int.21643"

[5]. *Arzucan ¨Ozg¨ur, Levent ¨ Ozg¨ur, and Tunga G¨ung¨or*, “Text Categorization with Class-Based and Corpus-Based Keyword Selection”, in ISCIS'05 Proceedings of the 20th international conference on Computer and Information Sciences, 2005, pp. 606–615.

[6]. *T. Joachims*, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, European Conference on Machine Learning (ECML), l998.

[7]. *G. Guo, H. Wang, D. Bell, Yaxin Bi and K. Greer*, “An kNN Model-based Approach and Its Application in Text Categorization”, CICLing, 2004, pp. 559-570.

[8]. *L.¨Ozg¨ur, T. G¨ung¨or and F. G¨urgen*, “Adaptive Anti-Spam Filtering for Agglutinative Languages. A Special Case for Turkish”, Pattern Recognition Letters, **vol.** 25, no. 16, 2004, pp. l8l9–l83l.

[9]. *A. McCallum and K. Nigam*, “A Comparison of Event Models for Naive Bayes Text Classification”, in Proceedings of AAAI Workshop on Learning for Text Categorization, l998, pp. 4l–48.

[10]. *J. Zhang, R. Jin, Y. Yang and A. Hauptmann*, "Modified logistic regression: An approximation to svm and its applications in large-scale text categorization", in Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, USA, 2003.

[11]. *M. J. D. Powell*,, “Radial basis functions for multivariate interpolation: A review”, in Algorithms for the Approximation of Functions and Data, J.C. Mason and M. G. Cox eds. Clarendon Press, Oxford, England, 1987.

[12]. *Y. Yang and X. Liu*, “A Re-examination of Text Categorization Methods”, in Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, Berkeley, US, 1996.

[13]. *K.T. Assaleh and M. AL Rousan*, “A New Method for Arabic Sign Language Recognition”, Personal communications, 2004.

[14]. *W. M. Campbell, K. T. Assaleh and C. C. Broun*, “A Novel Algorithm for Training Polynomial Networks”, International NAISO Symposium on Information Science Innovations ISI’2001, Dubai, March 2001.

[15]. *G. H. Golub and C. F. Van Loan*, “Matrix Computations”, John Hopkins, Washington DC, 1989.

[16]. Ana Site for Data Sets Suitable for Single-Label TextCategorization. http://www.gia.ist.utl.pt/~acardoso/datasets/

[17]. *M. F. Porter*, “An Algorithm for Suffix Stripping”, Program, **vol.** 14, no. 3, 1980, pp. 130-137. doi:10.1108/eb046814

[18]. *T. Hastie, R. Tibshirani and J. Friedman*, “The Elements of Statistical Learning”, Springer, 2001.

[19]. *M. Dash, K. Choi, P. Scheuermann and H. Liu*, “Feature selection for clustering – a filter solution”, in Proceedings of the Second International Conference on Data Mining, 2002, pp. 115–122.

[20]. *M.A. Hall*, “Correlation-based feature selection for discrete and numeric class machine learning”, in Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 359–366.

[21]. *H. Liu and R. Setiono*, “A probabilistic approach to feature selection - a filter solution”, in Proceedings of the Thirteenth International Conference on Machine Learning, 1996, pp. 319–327.

[22]. *L. Yu and H. Liu*, “Feature selection for high-dimensional data: a fast correlation-based filter solution”, in Proceedings of the twentieth International Conference on Machine Learning, 2003, pp. 856–863.

[23]. *R. Caruana and D. Freitag*, “Greedy attribute selection”, in Proceedings of the Eleventh International Conference on Machine Learning, 1994, pp. 28–36.

[24]. *J. G. Dy and C. E. Brodley*, “Feature subset selection and order identification for unsupervised learning”, in Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 247–254.

[25]. *Y. Kim, W. Street, and F. Menczer*, “Feature selection for unsupervised learning via evolutionary search”, iIn Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 365–369.

[26]. *R. Kohavi and G.H. John*, “Wrappers for feature subset selection” Artificial Intelligence, 97(1-2): 1997, pp. 273–324.

[27]. *S. Das*, “Filters, wrappers and a boosting-based hybrid for feature selection”, in Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 74–81.

[28]. *A. Y. Ng*, “On feature selection: learning with exponentially many irrelevant features as training examples”, in Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 404–412.

[29]. *E. Xing, M. Jordan, and R. Karp*, “Feature selection for high-dimensional genomic microarray data”, in Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 601–608.

[30]. *P. Langley*, “Selection of relevant features in machine learning”, in Proceedings of the AAAI Fall Symposium on Relevance, 1994, pp. 140–144.

[31]. *S-H. Lin, C-S. Shih, M. C. Chen and J-M Ho*, “Extracting Classification Knowledge of Internet Documents with Mining Term Associations: A Semantic Approach”, in Proceedings of ACM/SIGIR (l998), Melbourne, Australia, pp. 24l–249.

[32]. *A. P. Azcarraga, T. Yap and T. S. Chua*, “Comparing Keyword Extraction Techniques for Websom Text Archives”, International Journal of Artificial Intelligence Tools, **vol.** 11, no. 2, 2002.

[33]. *G. Forman*, “An Extensive Empirical Study of Term Selection Metrics for Text Classification”, Journal of Machine Learning Research, **vol. 3**, 2003, pp. 1289-1305.

[34]. *Y. Yang, and J. Pederson*, “A Comparative Study on Feature Selection in Text Categorization”, In Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann, 1997, pp. 412-420.

[35]. *K. Fuka, and R. Hanka*, "Feature Set Reduction for Document Classification Problems", IJCAI-01 Workshop: Text Learning: Beyond Supervision, Seattle (August 2001), USA, 2001.

[36]. *M. Rogati and Y. Yang*, "High-Performing Feature Selection for Text Classification", CIKM'02, November pp. 4-9, 2002.

[37]. *T.E. Dunning*, "Accurate methods for the statistics of surprise and c oincidence", in Computational Linguistics, **vol,** 19, no. 1, 1993, pp. 61-74.

[38]. *Z. Zheng, X. Wu and R. Srihari*, "Feature selection for Text Categorization on Imbalanced Data", SIGKDD Explorations, **vol**. 6, no. 1, 2004, pp. 80-89. doi:10.1145/1007730.1007741

[39], *V. Pekar, M. Krkoska and S. Staab*, "Feature Weighting for Co-occurrence-based Classification of Words", in Proceedings of the 20th Conference on Computational Linguistics, COLING-2004, August 2004.

[40]. *C.J. Van Rijsbergen*, "Information Retrieval", Butterworths, London, second edition, 1979.

[41]. *D. Mladenic and M. Globelnik*, "Word sequences as features in text learning", in Proceedings of the 17th Electrotechnical and Computer Science Conference (ERK98), Ljubljana, Slovenia, 1998, pp. 145–148.

[42]. *D. Mladenic and M. Grobelnik*, "Feature Selection for Unbalanced Class Distribution and Naive Bayes", in Proceedings of the l6th International Conference on Machine Learning, Morgan Kaufmann, 1999, pp. 258-267.

[43]. *L. Galavotti, F. Sebastiani, and M. Simi*, "Experiments on the use of feature selection and negative evidence in automated text categorization", in Proceedings of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries (Lisbon, Portugal), 2000, pp. 59-68.

[44]. *Z. Zheng and R. Srihari*, "Optimally Combining Positive and Negative Features for Text Categorization", Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003.

[45]. *F. Sebastiani*, "Machine learning in automated text categorization", ACM Computing Surveys, **vol.** 34, no. l, 2002, pp. l–47.

[46]. *F. Debole and F. Sebastiani*, "Supervised Term Weighting for Automated Text Categorization", in Proceedings of SAC-03, 18th ACM Symposium on Applied Computing, Melbourne, US, 2003, pp. 784-788.

[47]. *G. Salton and C. Buckley*, "Term Weighting Approaches in Automatic Text Retrieval", Information Processing and Management, **vol**. 24, no. 5, 1988, pp. 5l3–523.

[48]. *F. Debole and F. Sebastiani*, "An Analysis of the Relative Hardness of Reuters-21578 Subsets", JASIS, **vol.** 56, no. 6, 2005, pp. 584-596.

[49]. *R. Tesar, M. Poesio, V. Strnad and K. Jezek*, "Extending the Single Words-Based Document Model: A Comparison of Bigrams and 2-Itemsets", DocEng'06, October 10–13, 2006, Amsterdam, The Netherlands. Copyright 2006 ACM.