FACT: Fast Algorithm for Categorizing Text

Saket S.R. Mengle
Computer Science Department
Illinois Institute of Technology
Chicago, Illinois, U.S.A
saket@ir.iit.edu

Nazli Goharian
Computer Science Department
Illinois Institute of Technology
Chicago, Illinois, U.S.A
nazli@ir.iit.edu

Alana Platt
Computer Science Department
Illinois Institute of Technology
Chicago, Illinois, U.S.A
platt@ir.iit.edu

Abstract— **With the ever-increasing number of digital documents, the ability to automatically classifying those documents both quickly and accurately is becoming more critical and difficult. We present Fast Algorithm for Categorizing Text (FACT), which is a statistical based multi-way classifier with our proposed feature selection, *Ambiguity measure(AM)*, that uses only the most unambiguous keywords to predict the category of a document. Our empirical results show that FACT outperforms the best results on the best performing feature selection for the Naïve Bayes classifier namely, Odds Ratio. We empirically show the effectiveness of our approach in outperforming Odds Ratio using four benchmark datasets with a statistical significance of 99% confidence level. Furthermore, the performance of FACT is comparable or better than current non-statistical based classifiers.**

Keywords: Text classification, Text processing, text mining, knowledge discovery.

I. Introduction

There is an overflow of data in digital format. Vast volumes of online text are available via the World Wide Web (WWW), news feeds, electronic mails, corporate databases, medical patient records and digital libraries. The problem of classifying and storing these documents pose a significant challenge. Companies already spend significant amounts of their resources on classifying documents manually; and the feasibility of manual classification decreases as the number of documents increase over time. As the number of documents is large, a fast and scalable automatic classifier is needed to classify the existing and incoming documents accurately and efficiently.

We propose a novel method called Fast Algorithm for Classifying Text or FACT. FACT works as a feature selection algorithm, which is also used as a statistical-based crude classifier similar to Naïve Bayes classifier. FACT uses our proposed method, Ambiguity Measure (AM) as a feature selector that selects the most unambiguous features, where unambiguous features are those features whose presence in a document indicates a high degree of confidence that the document belongs to a specific category.

Text classification involves scanning through text, and assigning categories to documents to reflect their content. The main applications of text classification are filtering and routing. In particular, large companies filter incoming e-mail and store them in folders or route them to concerned departments. News agencies may also use classification tools for filtering or routing the news from different sources to their appropriate client. Medical categorization tools are used to assign medical keywords to text written by clinicians both to allow compilation of performance statistics for hospitals, and to enable retrieval of relevant text. Other applications of text classification are in the field of knowledge-base extraction, e-commerce and information extraction.

Different machine learning algorithms are used to automatically classify text. One of such algorithms is Naïve Bayes, which creates a statistical model using training data. Algorithms that differentiate useful features are called feature selection algorithms. Examples of such feature selection algorithms are odds ratio, information gain, correlation coefficient, GSS coefficient [2,11,13]. Among these feature selection algorithms odds ratio consistently leads to statistically significant improvement in classification effectiveness in comparison to the full feature set and other feature selection algorithms [5,6,7]. Based on statistics used in Naïve Bayes and Odds Ratio, one could characterize them as *compatible* in the sense that the features with higher Odds Ratio weights are expected to be more influential within the Naïve Bayes classifier [5]. Similar to Odds Ratio, our proposed algorithm, FACT, selects the best features, but unlike the odds ratio, FACT only considers the categories in which document term exists, i.e. positive categories. This is explained in detail in sections 2 and 3. We compared FACT and Odds ratio using four different datasets from different subject domains, namely news, web pages, and bio-medical text, and concluded that FACT outperforms odds ratio feature selection. We also compared our results with other current text classifiers showing comparable or better results.

II. Prior Work

Feature selection helps to achieve two objectives: to reduce the size of feature set in order to optimize the classification efficiency, and to reduce noise in the data in order to optimize the classification effectiveness [5]. We present the commonly used feature selection approaches below.

$$IG(t_k c_i) = \sum_{c \in (c_i, \bar{c_i})} \sum_{t \in (t_k, \bar{t_k})} P(t, c) log_2 \frac{P(t, c)}{P(t) P(c)}$$

$$CHI(t_k c_i) = \frac{N[P(t_k, c_i). P(\bar{t_k}, \bar{c_i}) - P(t_k, \bar{c_i}). P(\bar{t_k}, c_i)]^2}{P(t_k). P(c_i). P(\bar{t_k}). P(\bar{c_i})}$$

$$CC(t_k c_i) = \frac{\sqrt{N}[P(t_k, c_i). P(\bar{t_k}, \bar{c_i}) - P(t_k, \bar{c_i}). P(\bar{t_k}, c_i)]}{\sqrt{P(t_k). P(c_i). P(\bar{t_k}). P(\bar{c_i})}}$$

$$OR(t_k c_i) = \frac{P(t_k|c_i). [1 - P(t_k|\bar{c_i})]}{[1 - P(t_k|c_i)]. P(t_k|\bar{c_i})}$$

$$GSS(t_k c_i) = P(t_k, c_i). P(\bar{t_k}, \bar{c_i}) - P(t_k, \bar{c_i}). P(\bar{t_k}, c_i)$$

Where IG = Information gain [11]
CHI = Chi Squared [11][13]
CC = Correlation Coefficient [13]
OR = Odds ratio [13]
GSS = GSS coefficient [2][13]

The description of these feature selection algorithms is given is [2], [11] and [13], thus we have omitted their mathematical justification. The feature selection algorithms given above use the knowledge about the presence of terms in relevant categories (c_i) as well as in non-relevant categories ($\bar{c_i}$). In our approach, we have proposed a feature selection method that only uses the knowledge about the presence of terms in relevant categories.

Naïve Bayes classifier is shown to perform well if good feature selection algorithm is used. Using odds ratio feature selection algorithm as compared to all other feature selection algorithms improves the effectiveness of Naïve Bayes classifier significantly [5]. *Odds ratio* is a feature selection algorithm, which specifies the ratio of the odds that a term is related to a particular category to the odds that a term is not related to that category [5]. In our study, we use odds ratio as our baseline to evaluate FACT.

Efforts have been made to use only the relevant categories for feature selection [1]. [1] considers *tf-idf* weight; *tf* refers to term frequency with respect to a given category and *idf* is modified as *icf* that gives a ratio between the total number of categories to the number of categories a document may belong to. Some of the terms may only appear in one category for few numbers of times. Although these terms appear in only a single category, they are purged during feature selection as they have a low term frequency. Furthermore, some terms frequently appear in a few categories (i.e. a high icf) with a similar distribution of occurrence in all categories. Such terms are ambiguous, as they do not point strongly to only a single category. But as the term frequency of such terms is high, these terms may be selected as good features. Our feature selection method avoids such situations by only considering the ratio between the numbers of occurrences of a term in a given category to the total number of occurrences of the term in training set. Thus both these situations are avoided.

In addition to the efforts in statistical based classifiers, non-statistical classifiers are also reported to perform well to classify text. Examples of such are SVM and kNN, which are shown to perform better than Naïve Bayes algorithms [9]. It is shown that variations of SVM perform the best for text classification [8]. To show that the results of our feature selection method are comparable or better than such non-statistical approaches, we compare our results with results presented in [8] and [9]. Multi labeled classification using maximum entropy method, which is a variation of SVM [8], uses the correlation between the different categories to estimate the class. Although FACT also uses multi-labeling to a very small extent, it does not use the co-relations of categories. FACT categorizes a document into multi-labels, if the probabilities among the top categories are very similar. Some methods such as Drag Pushing [9] take advantage of the training error to successively refine the classification model of a base classifier. For Refined Centroid classifier (RCC), which is proposed in [9], the centroid of a correct class is dragged towards a misclassified example while the centroid of an incorrect class is pushed away from the misclassified example. These refinements to the existing text categorization algorithms have shown improvement in the results. FACT does not use information about training error and only calculates the ambiguity measures during training session.

III. Methodology

Initially, we describe the intuitive motivation behind our approach and then provide a formal definition of our method. We consider the human perception of the topic of a document by a glance at the document and capturing the *keywords*. Instead of using all the terms in a document to determine the subject of a text, normally one bases his/her decision on the most unambiguous words that the eye captures. The person then has an idea of the topic of the document. Some words can easily suggest the category in which the document can fall into. For example, if the document has words like "Chicago White Sox" and "MLB World series Champion", then one can suggest that the document relates to baseball in particular and sports in general. The sample text below is taken from www.cnn.com. By having a glance at this text, the reader can have a guess about the category of this text.

"This week, the United Nations created the position of czar in the global fight against a possible avian influenza pandemic. Meanwhile, officials here in the United States acknowledged the country is unprepared if this never-before-seen strain of flu, known to scientists as H5N1 virus, were to hit this winter".

The text seems to be about "Avian Flu". Our human perception is based on our knowledge of the domain or what we hear daily on various subjects in daily life. Thus, if someone has heard about "avian influenza" and has heard about H5N1 virus, then without reading the text can confidently claim that the text belongs to Epidemic rather than *Terrorism* or *Computer Crimes.*

Furthermore, some terms may be stronger indicators that a given text belongs to a certain category than to others. Thus, we can give a particular weight as to how strongly a term suggests a particular category. We clarify this by giving the following hypothetical example.

"Carolina Panthers lost the Superbowl title to Chicago Bears due to the final minute touchdown"

In the above sentence, we have terms such as *Bears* and *Panthers*, which are related to *wildlife*. On the other hand, they are also the names of famous NFL football teams. Here we notice uncertainty in classifying the text to *Wildlife* or to *Sports* categories. Considering the terms such as *Superbowl* and *touchdown*, in the same given text, suggests more confidently that the text is about *Sports*.

FACT classifies the documents based on non-ambiguous document terms in respect to a given category or topic. The algorithm is based on the idea that some terms are ambiguous for some categories and non-ambiguous for other. By identifying ambiguous and non-ambiguous terms for each category, we classify any new incoming document to a category based on number of non-ambiguous terms of that document with respect to a given category. We call these non-ambiguous terms the *keywords*. A term such as “America” can occur in any category; and thus, it is not a good indicator of membership of a document that has the term “America” to any category. Furthermore, we also consider the strength of membership of a given term to each category. Some keywords may appear in multiple categories but are a stronger evidence of membership to a given category as compared to other categories.

We define an *ambiguity measure, AM,* for each term and use that to identify whether a word is a *keyword* on which to base the classification decision on. In the above example the term *touchdown* has a lower *ambiguity measure* than that of the terms *Bears* and *Panthers*. We then assign a higher weight to the less ambiguous terms. Ambiguity measure is explained in detail in 3.1. FACT performs the categorization in two steps. Initially, a model is built and then any new incoming data are classified into one or more categories. Our algorithms for both steps are described below in sections 3.1 and 3.2.

3.1 PHASE 1: BUILDING THE MODEL

FACT takes advantage of the existing inverted index for mapping of the terms to the documents; it also takes advantage of the existing categorized data, i.e., training data, to calculate the *Ambiguity Measures, AM*, to build the model. A map of documents to categories is kept in memory that is used to calculate the ambiguity measure of a term to a particular category. In the case that there is no existing inverted index, the document collection is parsed and the statistics that define the AM value are extracted. For the same example given above, the number of times the term “H5N1” appears in the given corpus for each category is calculated. The frequency counts for each category indicate a confidence level as to how well the word “H5N1” defines a particular category.

Formally, *Ambiguity measure (AM)* is defined as the probability that a term falls into a particular category and is calculated using the following formula. Closer the AM value to 1 then the term is considered less ambiguous. Conversely, if AM is closer to 0, the term is considered more ambiguous with respect to a given category.

$$AM(term\ t_i \mid category\ C_j) = \left(\frac{frequency\ of\ t_i\ in\ c_j}{collection\ \ frequency\ of\ t_i} \right)$$

The result of the calculation of Ambiguity *measure (AM)* for the term “H5N1” is given in table 3.1, indicating *Epidemic* category for the term. For various datasets, we have empirically determined a threshold for AM value. The explanation on threshold is provided in section 6 and via figures 6.2.2.

Table 3.1. Ambiguity Measure (AM) example

Term	H5N1		Virus		Officials	
Category	**Count**	**AM**	**Count**	**AM**	**Count**	**AM**
Pornography	10	0.01	150	0.15	150	0.15
Epidemic	990	0.99	800	0.80	240	0.24
Drug trafficking	0	0	00	0.00	330	0.33
Terrorism	0	0	50	0.05	280	0.28

A term can be part of more than one category if the AM is above the threshold in more than one topic. Empirically, we set our threshold value to 0.60. This signifies that the document has occurred in more than 60% of the total documents. In the example (Table 3.1) the term “virus” belongs to the *Epidemic* category with an AM value of 0.80 that satisfies the 0.60 threshold. The category *Pornography* has an ambiguity measure of 0.15 that indicates a very weak relevance of the document. The other two categories *Drug Trafficking* and *Terrorism* have extremely low indication of relevance to the document with an AM of 0.00 and 0.05 respectively. In some cases the AM value is lower than the threshold and thus the term cannot be assigned to any of the categories. Example of such is the term “Officials” (Table 3.1), which does not satisfy the AM threshold in any of the categories and thus, is considered ambiguous for all categories. Consequently, the term “Officials” is not qualified as a keyword. The terms with an AM value below the threshold for every single category do not satisfy the threshold condition and are filtered out. Otherwise, if a term is qualified for at least one category then the term is kept. This filtering of the terms saves both the space and increases the accuracy by not considering the terms that do not point with confidence to at least one category.

3.2 PHASE 2: CLASSIFICATION

In the classification (testing) phase, each new incoming document is classified into one or more, or no categories. The document terms that are qualified as keywords, i.e., are below the threshold are considered. These document *keywords* together provide the probability that a given document belongs to a given category. This probability is calculated as the product of the individual AM values of the *keywords* in a document and is given as:

$$\text{P(category } \mathrm{C_j}) = \prod_{i=1}^{n} AM(term\ t_i \mid category\ C_j)$$

where *n* is the number of keywords in the document.

The highest probability value indicates that the document is closely related to the category and is chosen as the predicted category. If the highest and second highest probability values are close enough, then the document is assigned to both categories. The closeness measure is defined empirically. We noticed that if a third category is also selected, the classification accuracy drops as more false positives are introduced. Moreover, if the highest probability product is still low then the classification is not accurate; thus, FACT labels these cases as *uncertain*.

IV. EXPERIMENTAL SETUP

We empirically evaluated the effectiveness of our approach using four benchmark data sets, which are commonly used in text classification evaluation. We chose the datasets such that we cover different types of document domains for which text classification is used, namely news articles, web pages and bio-medical documents.

Table 4.1. Benchmark datasets used for our experiments

Datasets	**No. of documents**	**Avg. doc length**	**No. of Categories**	**Size MB**	**Domain**
Reuters 21578	21578	200	Top 10	28	News Articles
20 News Group	20000	311	20	61	News Articles
WebKB	8282	130	7	43	Web Pages
OHSUMED	54710	64	Top 50	382	Bio Medical Documents

Details about these datasets and our experimental plans are as follows:

4.1 Experimental plan using Reuters- 21578 Dataset

The Reuters 21578 corpus [2] contains Reuters news articles from 1987. The documents range from being multi-labeled, single labeled, or not labeled. Reuters dataset consists of a total number of 135 categories (labels). However, ten of these categories have significantly more documents than the rest of the categories. Thus, commonly the top 10 categories are used for experimentations and to compare the accuracy of the classification results. The top 10 categories of Reuters 21578 are “earn”, “acq”, “money-fx”, “grain”, “trade”, “crude”, “interest”, “wheat”, “corn” and “ship”.

We performed two sets of experimentations using different training and testing splits on Reuters-21578. 1) The standard *ModApte* split of 9603 training documents and 3299 testing documents (the remaining 8676 documents are never used as they do not have any labels); 2) Stratified 10-fold cross validation with 7855 training documents and 811 testing documents; and The results and analysis of the results are given in section 6.

4.2 Experimental plan using 20 Newsgroup (20NG) Dataset

20 Newsgroup (20NG) [3] consists of a total of 20000 documents that are categorized into twenty different news groups. Each category contains one thousand documents. The size of the documents is much larger than those in Reuters data set. Some of the newsgroups categories are very closely related to each other (e.g. comp.sys.ibm.pc.hardware and comp.sys.mac.hardware), while others are highly unrelated (e.g. misc.forsale and soc.religion.christian). This characteristic contributes to the difficulty of categorization of documents that belong to very similar categories [3]. We performed experimentations using stratified 10-fold cross validation.

4.3 Experimental plan using WebKB dataset

WebKB dataset is a collection of web pages from different college websites namely Cornell, Texas, Washington, Wisconsin and some miscellaneous web pages. These web pages are pre-classified as student, faculty, staff, department, course, project and others (7 categories). WebKB contains 8282 web pages. We train with data from three universities at a time and the miscellaneous pages, and test on the other remaining college pages. Thus, our results are based on 4-fold cross validation.

4.4 Experimental plan using OHSUMED dataset

OHSUMED is a collection of Medline documents, i.e., medical citations, from 1987 to 1991, and commonly used for bio-medical literature search evaluation and classification. The average document length in the collection is 64 words that is less than in most of the other datasets used in the experimentations. We use 54170 documents from 1987 and top 50 MESH categories. Stratified 10- fold cross validation is used for evaluation.

V. EVALUATION MATRICES

To evaluate the accuracy of our approach and compare FACT to the state of the art feature selection research results, we use the commonly used evaluation metrics precision, recall, F1 measure, and accuracy, as are defined below:

$$\text{Precision (P)} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall (R)} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{F1 measure} = \frac{2PR}{P+R}$$

$$\text{Accuracy} = \frac{1}{n}\sum_{i=0}^{n} f(x(i), y(i))$$

Where, *n* is the total number of documents; and f(x(i), y(i)) is a function which returns one when category x(i) is equal to y(i) else it returns zero[8].

To measure the overall effectiveness, i.e., overall precision, recall, and F1 measure over all the categories, we use *micro* and *macro averaging*. Micro-averaging calculates the total true positives, false positives, and false negatives over all categories and uses the same precision, recall and F1 formulas given above to calculate each of the measures over all categories. Macro-averaging uses the individual category precisions, recalls, and F1 measures to build the average of each of these measures over all categories. The macro-averaging may lead to an inaccurate evaluation if the categories have very skew distribution, as the number of categories is considered in building the average. Thus, a category with small number of documents and high precision may mislead the overall precision.

VI. RESULTS

We organize the results into three subsections. In section 6.1, we present the effectiveness of our approach on four benchmark datasets covering different subject areas. In section 6.2, we compare FACT with odds ratio, which is reported to be the best feature selection algorithm on Naïve Bayes [5,6,7], and show that FACT outperforms odds ratio on all four datasets used. In section 6.3, we compare FACT with the non-statistical algorithms and show that FACT outperforms them.

6.1. Results of FACT on datasets of different domains

Table 6.1.1 shows the results of FACT (accuracy, micro and macro averaging of precision, recall, and F1 measure) using ModApte and stratified 10-fold cross validation. We present the results for the best run and the average over all ten runs. The results of the best runs are given with respect to the best micro average F1 measure. The results using both 10-fold cross validation and bootstrapping are over 90% for accuracy, micro average precision and recall, and F1 measures. In ModApte split they are statistically significantly lower than 10-fold (99% confidence level). For experiments using ModApte split, only one iteration is used, unlike in the other cases in which 10 iterations are used.

We performed stratified 10-fold cross validation on 20 News Group (20NG) dataset. The performance of FACT in categorizing the documents in 20NG dataset is shown to be better (with 99% confidence level) than FACT's performance on the Rueters-21578 dataset. In 20NG, FACT achieves a micro-average F1 of 94.96 and a macro-average F1 of 91.74, while on the Reuters-21578 dataset FACT has a micro-average F1 of 90.54 and a macro-average F1 of 88.92. The contributing factor in this is that the training set for 20NG is balanced with the same distribution of documents in each category. Moreover, the documents in 20NG dataset are larger than those in Reuters and thus have larger number of keywords to reduce the ambiguity of classification.

We performed 4-fold cross validation on WebKB. As explained in section 4.3 the data set is divided into 4-folds. The performance on WebKB (micro-avg. F1 of 74.05 and macro-avg. F1 of 50.34) is not as good as Reuters or 20NG. There are fewer keywords found in web pages because many of the terms in a web page document are part of the headers or structure of a website. These words are ambiguous and very few of them can be used as keywords. In short, the number of keywords found per document in WebKB dataset is less than those found in the other datasets.

We performed stratified 10-fold cross validation on OHSUMED dataset. FACT's performance in categorizing the documents in OHSUMED (micro-average F1 of 59.88 and macro-average F1 of 49.82) is shown to be worse than FACT's performance on datasets like Reuters 21578, 20 News Group and WebKB. However, comparing with the results of recent research on OHSUMED, FACT achieves statistically significant improvement, as shown in section 6.3.3. The lower accuracy on OHSUMED dataset is due to the bio-medical domain with many ambiguous terms overlapping in the closely related categories.

6.2 Comparing FACT with Odds Ratio Feature Selection

In this section, we compare FACT with odds ratio. A brief comparison of FACT and odds ratio based on a variety of datasets is given in Table 6.2.1. In this set of experiments, we keep the training and testing split the same for both approaches (Odds Ratio and FACT) and obtain results for different threshold values. We report the best results for both cases.

As described in paper [5], the features with higher odds ratio weights are expected to be more influential within the Naïve Bayes classifier. This characteristic of odds ratio can be seen in Figure 6.2.2 (A-D). The X-axis shows the threshold values for filtering the features, and the Y-axis represents the value of F1 measure. All the features that fall above the threshold are selected for testing. As both the values (Odds ratio and Naïve Bayes) are ratios, we only change the value of threshold between the scales of 0 to 1.

In all cases, (6.2.2 A-D), by finding an optimal threshold, FACT outperforms the Odds Ratio.

Table 6.1.1 Results of FACT on Reuters 21578, 20 News group, Web KB, OHSUMED benchmark datasets

Datasets			Mic- avg. Precision	Mic-avg. Recall	Mic-avg. F1	Mac-avg.	Mac-avg.	Mac-avg.
Reuters 21578	**ModApte Split**		88.52	88.41	88.46	83.51	75.64	79.38
	Stratified 10-fold cross validation	Best run wrt. micro-average F1	93.82	87.48	90.54	89.08	83.71	86.31
		Average of 10 runs	92.36	85.72	88.92	87.82	82.48	84.82
20NG	**Stratified 10-fold cross validation**	Best run wrt. micro-average F1	93.87	96.09	94.96	90.26	84.67	87.38
		Average of 10 runs	91.68	91.69	91.74	92.17	83.3	87.5
WebKB	**Stratified 4-fold cross validation**	Best run wrt. micro-average F1	75.35	74.43	74.89	51.23	50.15	50.86
		Average of 10 runs	74.34	73.76	74.05	50.93	49.56	50.34
OHSUMED	**Stratified 10 fold cross validation**	Best run wrt. micro-average F1	68.75	59.67	63.83	55.24	47.94	51.33
		Average of 10 runs	65.93	54.84	59.88	53.37	46.72	49.82

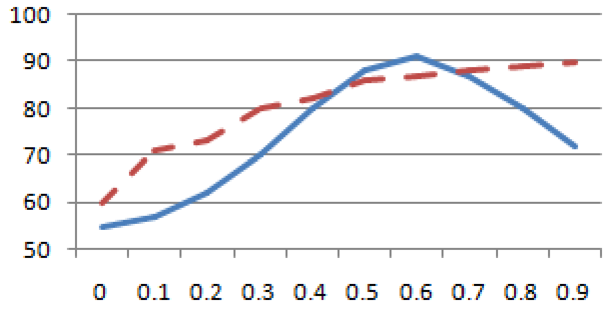
We evaluated this outcome using Reuters, 20NG, WebKB and OHSUMED datasets.. The results given in table 6.2.1 contain the results for FACT and odds ratio each at their own best thresholds. The results show that FACT outperforms odds ratio on all the four datasets. The results for FACT on all the datasets are statistically significantly better (99% confidence) than of odds ratio.

Table 6.2.1 Comparison between FACT and odds ratio on different datasets

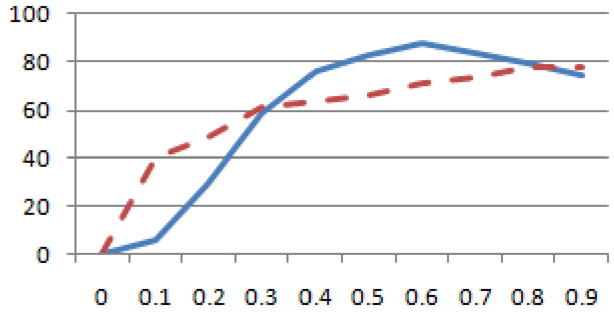
Dataset	Algorithm	Micro precision	Micro recall	Micro F1
Reuters 21578	FACT	92.36	85.72	88.92
	Odds ratio	90.52	86.92	88.68
20 News Group	FACT	91.68	91.69	91.74
	Odds ratio	93.12	70.28	80.10
WebKB	FACT	74.34	73.76	74.05
	Odds ratio	71.34	70.12	70.72
OHSUMED	FACT	65.93	54.84	59.88
	Odds ratio	45.16	42.99	44.05

6.3. Comparison with current text classifiers

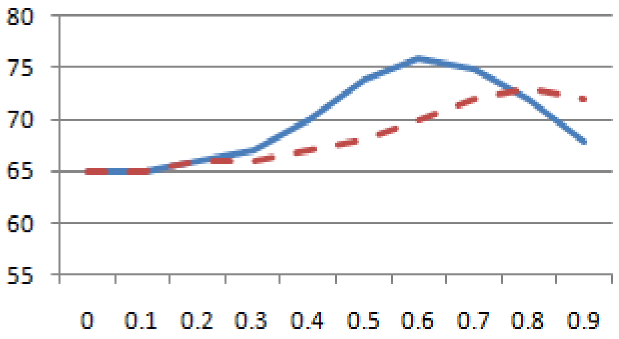
Naïve Bayes is an efficient approach to text classification [5,7]. FACT works as a feature selection method for naïve Bayes algorithm. SVM and its variations are shown to work more effectively than Naïve Bayes algorithm on different datasets. Thus, we also favorably compare our results with the results of recently published state of the art text classifiers.



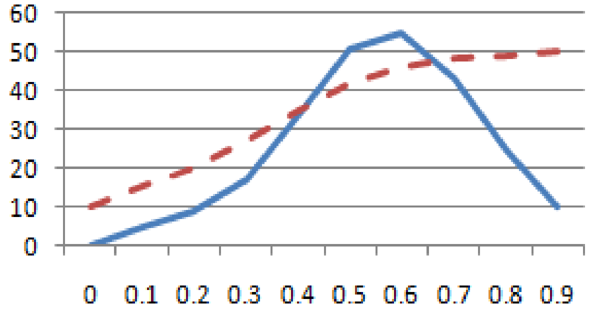
6.2.2A F1 measure comparison for *Reuters 21578*



6.2.2B F1 measure comparison for *20 News Group*



6.2.2C F1 measure comparison for *WebKB*



6.2.2D F1 measure comparison for *OHSUMED*

X-axis represents the different threshold values used for different runs; Y-axis represents Micro F1

Table 6.3.1 Comparison of FACT with the multi labeled classifiers on 9-1 split for Reuters 21578 dataset for top 10 categories reported in [8]

	FACT	**Multi-label maximum entropy**	**Combinational method**
Avg. Accuracy **P-value**	**90.88%**	89.35% 0.0098	88.51% 0.002
Micro-Avg. F1	91.74%	**91.80%**	91.04%

In Table 6.3.1, we compare FACT with the state of the art variant of SVM that similarly is a multi label algorithm [8]. The experiments are based on 9-1 split on Reuters-21578. Table 6.3.1 shows their reported average accuracy using *Multi-label maximum entropy* (MLME) method that considers the correlation between the various categories for multi labeled classification; and using *Combinational method* (COMB) that uses one-versus-all method. FACT outperforms MLME and COMB methods in the average accuracy. The micro-average F1 is comparable in all cases. Using Wilcoxon signed-rank statistical significance test [10], a nonparametric paired test to compare the performance of different methods, we show that FACT outperforms the state of the art results in [8] with a confidence level of 99%.

Table 6.3.2 Comparison of FACT with different algorithms using 20 Newsgroup reported in [9]

	RCC	**NB**	**RNB**	**KNN**	**SVM**	**FACT 3- fold**	**FACT 10-fold**
Micro-Avg F1	88.0	83.5	85.4	84.8	88.9	**90.54**	**91.74**

Table 6.3.2 shows the comparison of FACT with the existing baseline systems presented in [9], as well as their proposed approach of refined Centroid model (RCC) and refined Naïve Bayes model (RNB) that outperformed the existing approaches. Both approaches use DragPushing, which is a utility to manipulate the distance in Centroid model and the probabilities in Naïve Bayes to correct the classification error. The baseline classification algorithms used in this comparison are Naïve Bayes (NB), K-Nearest Neighbor (kNN), and support vector machine (SVM). The authors in [9] presented their results by the value of the F1 measure, based on 3-fold cross validation with a split of 66% for training and 33% for testing. As the precision and recall values for other algorithms are not reported in [9], we only compare our method with theirs using F1 measure. The results of FACT, both using 3-fold and 10-fold, significantly outperform the results of the state of the art work presented in [9].

VII. Complexity Analysis of FACT

In this section, we compare FACT with other algorithms available in terms of time complexity and space complexity. Table 7.1 gives the comparison of FACT with other popular algorithms like NB (Naïve Bayes), SVM (Support Vector Machine), kNN (k Nearest Neighbor), LLSF (Linear Least Square First), RR (Ridge Regression) and LR (Logistic Regression) with respect to time complexities. The time and space complexities reported in this section except for Naïve Bayes and FACT are taken from [12]. The discussion on time complexity of Naïve Bayes and FACT are given in sections 7.1 and 7.2.

Table 7.1 Time complexity for various text classification algorithms

Classifier	**Training time**	**Testing time per document**
FACT	O(N L_d +M V)	O(M L_v)
Naïve Bayes	O(N L_d +M V)	O(M L_v)
SVM	O(M N^c) c≈1.2~1.5	O(M L_v)
KNN	O(N L_d)	O((N/V) L_v^2) + O(N)
LLSF	O(N^2 k_s)	O(M L_v)
RR	O(M I N L_v)	O(M L_v)
LR	O(M I N L_v)	O(M L_v)

N - number of training documents
L_d- average document length
M- number of categories
k_s - value of k_s is empirically chosen through validation
I - number of iterations for iterative algorithms
L_v - average number of unique terms in document
V – size of vocabulary (features)

7.1 Analysis of time complexity for Naïve Bayes

During the training phase, document terms are parsed which equates to NL_d. A count for frequency of terms with respect to each category is maintained. It also calculates the posterior probabilities. For every term in vocabulary, M different posterior probabilities are calculated which takes O(M V) time. Thus, the training time for Naïve Bayes is $O(NL_d + MV)$. Generally it is considered only $O(NL_d)$ as $MV << NL_d$. During the testing phase, Naïve Bayes calculates the product of posterior probabilities and prior probability with respect to each category. This process takes $O(M\ L_v)$ time, where M is the total number of categories and L_v is the average length of a test document.

7.2 Analysis of time complexity for FACT

The time complexity of FACT is similar to the time complexity of Naïve Bayes. During the training phase, for each term in the training documents a count for the frequency of terms with respect to each category is maintained. Thus, FACT also parses NL_d terms during the training phase. Instead of posterior probabilities that are calculated in Naïve Bayes, FACT calculates ambiguity measures. For every term in the vocabulary, M different ambiguity measures are calculated which takes O(M V) time. Thus, the training time for FACT is also $O(NL_d + MV)$ and as generally $MV << NL_d$ it equates to $O(NL_d)$. During the testing phase, FACT calculates the product of ambiguity measures of terms present in the training document with respect to each category. This process takes $O(M\ L_v)$ which is the same for Naïve Bayes. Consequently, both FACT and Naïve Bayes are efficient and run in linear time. The comparison of space complexity of FACT and other algorithms is shown in Table 7.2. (all parameters are defined below table 7.1; *q* is size of working set in SVM).

Table 7.2 Space complexity for various text classification algorithms

Classifier	**Space Complexity**
FACT	O(M V)
Naïve Bayes	O(M V)
SVM	$O(N L_v + q^2)$
KNN	$O(N L_v)$
LLSF	O(N V)
RR	$O(N L_v)$
LR	$O(N L_v)$

FACT stores a table of all the terms in vocabulary (V) and their ambiguity measure with respect to all M categories. Thus, space needed by FACT is O(M V) that is the same as Naïve Bayes, which stores posterior probabilities for every term in vocabulary with respect to all the M categories. Many of the features are filtered while using FACT, thus only few of the features and their AM scores are stored. If using feature selection on Naive Bayes, then both Naïve Bayes and FACT have similar space requirements otherwise FACT requires less space. The analysis of space complexity for other algorithms except FACT and Naïve Bayes is given in [12] and presented in table 7.2.

VIII. Conclusion.

We proposed FACT (Fast Algorithm for Categorizing Text) as a new feature selection algorithm that is also used as a crude classifier similar to Naïve Bayes algorithm. The underlying premise behind the FACT approach is the quick identification of defining unambiguous words. Initially, FACT creates a training vocabulary. That is, unambiguous terms (keywords) are selected and a classification model is built. Based on this model, the documents that are to be classified are scanned to identify the keywords; and calculate the ambiguity measures (AM) of the keywords are used to calculate the probability that the document falls in a specific category. The category with the highest probability is selected as the category for that document.

We empirically evaluated the performance of FACT using four standard benchmark data sets (Reuters 21578, 20 News Groups, WebKB, and OHSUMED collection). Results using these collections show that FACT is better or equal to many existing state of the art algorithms. We compared FACT with odds ratio, which is considered to be the best feature selection algorithm for Naïve Bayes [5].

References

[1] Bong Chih How, Narayanan Kulathuramaiyer: An Empirical Study of Feature Selection for Text Categorization based on Term Weightage. Web Intelligence 2004: 599-602.

[2] Galavotti, L., Sebastiani, F., & Simi, M. (2000). *Experiments on the use of Feature Selection and Negative Evidence in Automated Text Categorization.* In proc. of ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries (Lisbon, Portugal, 2000), 59-68.

[3] Lang K.. Original 20 Newsgroups Dataset. http:// people.csai.mit.edu/jrennie/20Newsgroups .

[4] Lewis D., Reuters-21578, http://www.daviddlewis.com/resources/ testcollections/reuters21578.

[5] Mladenić D., Brank J, Grobelnik M., Milic-Frayling N. Feature Selection using Linear Classifier Weights: Interaction with Classification Models. In The 27th ACM SIGIR Conf. on Research and Development in Information Retrieval, pg 234-241, 2004.

[6] Mladenić D.,Grobelnik M. Feature selection for unbalanced class distribution and Naïve Bayes. Proc. 16th Int. Conf. on Mach. Learning. San Francisco: Morgan Kaufmann, pp. 258–267, 1999.

[7] Mladenic D. and M. Grobelnik. Feature selection for classification based on text hierarchy. In Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98). Carnegie Mellon Univ., Pittsburgh, 1998.

[8] Shenghuo Zhu Xiang Ji Xu W, Gong Y. Multilabelled Classification Using Maximum Entropy Method. In *The 28th ACM SIGIR Conference on Research and Development in Information Retrieval)*, pg 274-281, 2005.

[9] Tan S, Cheng X, Moustafa M. Ghanem, Wang B, Xu H. A Novel Refinement Approach for Text Categorization. In *The 14th ACM CIKM Conference* on Information and knowledge management, pg 469 - 476, 2005.

[10] Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics*, *1*, 80. 1993.

[11] Yang, Y., and Pedersen, J. (1997). *A comparative study on feature set selection in text categorization*. In Proc. of the 14th International Conference on Machine Learning, pages 412—420.

[12] Yiming Yang, Jian Zhang, Bryan Kisiel, A scalability analysis of classifiers in text categorization. Proc. of the 26th ACM SIGIR conference on Research and development in Information retrieval.

[13] Zheng, Z., Srihari, R, (2003). *Optimally Combining Positive and Negative Features for Text Categorization.* ICML 2003 Works