A Review of Text Classification Approaches for E-mail Management

Upasana Pandey and S. Chakraverty

Abstract—**The continuing explosive growth of textual content within the World Wide Web has given rise to the need for sophisticated Text Classification (TC) techniques that combine efficiency with high quality of results. E-mail filtering and email organization is an application rife with the potential to streamline the management of the vast amount of information that accumulates in the inbox. Even though a large body of research has delved into this area, there is a paucity of survey that indicates trends and directions. This paper attempts to categorize the prevalent popular techniques for classifying email as spam or legitimate and suggests possible techniques to fill in the lacunae in the arena of automatic management of emails. Our findings suggest that context-based email organization has the most potential in improving quality by learning various contexts such as n-gram phrases, linguistic constructs or users' profile based context to tailor his/her filtering scheme.**

Index terms—**Context Based TC, Context Interpretation, E-mail Management, Statistical TC**

I. Introduction

Text Classification (TC) is the task of automatically sorting a set of documents into categories such as topics from a predefined set. The task falls at the crossroads of information retrieval (IR) and Machine Learning (ML). It has witnessed a booming interest in the last ten years from researchers and developers alike due to its ever-expanding horizon of applications such as document classification, text summarization, essay scoring and user-specific presentation of textual material [1].

The Email affects every user of the Internet. However, emails also bloat and flood the inbox quickly leading to a morass of unorganized information. Even though many email providers allow the creation of folders and sub-folders where emails can be routed based on sender's address, date, subject etc. the whole process is largely manual. There is an urgent need for automatically segregating emails based on their relevance to the user.

As a basic need, spam filtering classifies messages into two categories, *viz. spam* and *non-spam*. Besides being undesired, spam email consumes a lot of network bandwidth. This is not a typical TC application. Over time, spammers resort to deceptive and deluging methods to get around antispam software thereby leading to a gradual degeneration of the filter's efficacy. To counter this, innovative TC approaches with good generalization, continuous adaptive learning and context sensitivity need to be applied. Extending this concept to the general case of filtering emails into several categories based on their relevance to the user, we can investigate TC approaches for personalized management of all emails.

Predominantly, *statistical approaches* have been applied for text classification. These approaches are based on the word occurrences i.e. frequency of one or more words in a given document. Several algorithms based on this method have been reported and have given good results in web applications [2-4, 6-15]. An alternative approach is *Context based text classification* that takes into account how a word *w1* influences the occurrence of another word *w2* in the document. Thus, the presence or absence of *w1* affects a classification based on *w2*. Even though some recent papers [16, 19] have reported techniques and algorithms for finding relevancy among words, significant work has not yet been carried out in the field of context based text classification for email applications. In this paper we present a survey focusing on statistical as well as some recent context based approaches for TC with focus on spam filtering and email applications.

Performance Measures: The following parameters are important performance indices for spam filtering. A *false positive* is result that classifies a legitimate email as a spam email. A f*alse negative* is a result that classifies a spam email as a legitimate email. A *False-positive* error that diverts a legitimate email as spam is generally considered more serious than a *False-negative.*

Now, out of all the spam emails, let *a* numbers of them be categorized correctly as spam (true positives) and the remaining *b* be categorized as legitimate (false negatives). Likewise, out of all legitimate emails, let *c* of them be erroneously categorized as spam (false positives) and remaining *d* be categorized as legitimate (true negatives). Let *N* be the sum total of *a, b, c, d.* The following scores are defined:

1. *Spam recall RE= a/ (a+b)*
2. *Spam precision PR= a/ (a+c)*
3. *F1-score:* This is defined as $2(1/RE + 1/PR)^{-1}$
4. *Accuracy AC=(a+d)/N*
5. *Error ER=(b+c)/N*

Macroaveraged results use a simple average of the above scores across all categories, giving equal weight to each category. Microaveraged results on the other hand first aggregate the*, b, c, d* and *N* values for *all* categories and then compute the above scores. Since all documents are given equal weight, this results in more frequently occurring categories being given greater weight.

II. Statistical Approaches

A. Naïve Bayes Classifier

A naïve Bayes classifier applies Bayesian statistics with strong independence assumptions on the features that drive

Upasana Pandey, (e-mail:upasana1978@gmail.com)
S. Chakraverty, (e-mail:apmahs@rediffmail.com)
Division of Computer Engineering ,Netaji Subhas Inst. of Technology, New Delhi-110078

the classification process. Essentially, the presence or absence of a particular feature of a class is assumed to be unrelated to the presence or absence of any other feature. *Bayesian spam filtering* is a form of e-mail filtering that uses the naïve Bayesian classifier to identify spam e-mail [2]. Suppose the suspected e-mail message contains the word *W*. Then the probability *Pr(S|W)* that the message is a spam is given by the formula:

$$\Pr(S|W) = \frac{\Pr(W|S).\Pr(S)}{\Pr(W|S).\Pr(S) + \Pr(W|H).\Pr(H)}$$

where *Pr*(*S*) is the overall probability that any given message is spam, *Pr*(*W*|*S*) is the probability that *W* appears in spam messages, *Pr*(*H*) is the overall probability than any given message is ham (not spam), *Pr* (*W* |*H*) is the probability that *W* appears in ham messages. During its training phase, a naïve Bayes classifier learns the posterior word probabilities.

The main strength of naïve Bayes algorithm lies in its simplicity. Since the variables are mutually independent, only the variances of individual class variables need to be determined rather than handling the entire set of covariances. This makes naïve Bayes one of the most efficient models for email filtering. It is robust, continuously improving its accuracy while adapting to each user's preferences when he/she identifies incorrect classifications thus allowing continuous rectified training of the model. In [3], the authors constructed a corpus *Ling-Spam* with 2411 non spam and 481 spam messages and used a parameter λ to induce greater penalty to false positives. They demonstrated that the weighed accuracy of a naïve-Bayesian email filter can exceed 99%. Variations of the basic algorithm for example, using word positions and multi-word N-grams as attributes have also yielded good results [4].

However, the naïve Bayes classifier is susceptible to *Bayesian poisoning*, a situation where a spammer mixes a large amount of legitimate text or video data to get around the filter's probabilistic detection mechanism.

B. Decision tree

A Decision Tree (DT) is a predictive model that expands a tree of decisions and their possible consequences, including chance event outcomes, and resource costs. The outcomes can be discreet or as in case of regression trees, continuous. Each leaf represents a unique classification and branches represent the conjunction of features that lead to the classifications at various leaves. Popular decision tree based learning methods are CART, ID3, C4.5 and Naïve Tree [5].

1) CART: - Classification and Regression Tree or CART based methods progressively split the set of training examples into smaller and smaller subsets on the basis of possible answers to a series of questions posed by the designer. When all samples in each subset acquire the same category label, each subset becomes Pure; such a condition would terminate that portion of the tree.

Text documents are typically characterized by very high dimensional feature spaces. Such excessive detailing or noisy training data run the risk of *overfitting.* In order to avoid overfitting and improve generalization accuracy, it is necessary to employ some pruning technique. CART uses the *Gini Impurity* parameter to pick only the most appropriate features for each parameter [5].

2) ID3:- The ID3 algorithm computes *entropy based Information Gain* for optimized feature selection. The recursive feature selection algorithm continues until there is only one class remaining in the data, or there are no features left.

3) C4.5:- C4.5 takes as input the tree generated by ID3 and attempts to reduce it by applying *rule post pruning.* The algorithm converts the tree into a set of if-then rules, and then prunes each rule by removing preconditions if the accuracy of the rule increases without it. The rules are then sorted according to their accuracy on the training set and applied in that order during classification.

4) Naïve Tree (NT):- Kohavi proposes a hybrid algorithm that combines the elegance of a recursive tree-based partitioning technique such as C4.5 with the robustness of naïve Bayes categorizers that is applied at each leaf [6]. By applying various datasets as inputs to NT, C4.5 and naïve Bayes, the average accuracy of NT is show to be 84.47%, 81.91% for C4.5 and 81.69% for naïve Bayes. In general the tree size learned by NT is smaller also than C4.5. Thus NT turns out to be more accurate, faster and more scalable than its constituents.

The main strength of DT based algorithms is their ability to generate understandable rules without complex computations. The *Information Gain* provides a clear indication of which features are most important for classification. Also DT can handle missing data by assuming it is randomly distributed within the dataset. In [7], the authors use a UCI Machine Learning Lab dataset containing 4600 emails, where 39.4% is spam emails and 60.6% is legitimate emails. The decision tree classifier filters the spam messages with a good overall accuracy of 97.17%.

One of the weaknesses of decision tree is that for a continuous attribute the information gain of many points within each variable has to be computed, adding to the computational cost. The process of growing a decision tree incurs the additional cost of sorting all candidate fields before the best split can be found. Pruning too bears the cost of generating and comparing several sub-trees. Due to these reasons, an issue with decision trees is: how to ensure that its performance scales well with the size of training data. The work in [8] proposes a framework for improving the scalability for any given DT method. Fast DT algorithms have been developed [9], that have a time complexity of $O(m.n)$ as compared with $O(mn^2)$ for C4.5, where *m* is the number of instances or records and *n* is the number of attributes.

C. Support Vector Machine (SVM)

An SVM is a supervised learning method based on *structural risk minimization* [5]. It subjects every category to a separate binary classifier. SVM's forte is that it is relatively immune to the dimensionality of the feature space, focusing instead on *maximizing the margin* between positive and negative examples of training documents. It avoids the use of many training documents, employing only those near the classification border, to construct an irregular border separating positive and negative examples. By employing a suitable kernel functions, it can learn polynomial classifiers, radial basis functions and three-layered sigmoid neural nets, thus acquiring universal learning ability.

1) *Soft Margin SVM*: Since a sharp separation is not always possible, the *Soft Margin* SVM chooses a hyperplane that splits the example as cleanly as possible, while still maximizing the distance between the nearest cleanly split examples.

2) *Combined Classifiers*: In [10], Tretyakov tried combining two filters, both showing a low probability of reporting false positives. Such a combination filter reports a message as a spam if either of the constituent filters categorizes it as spam. The combination is used to yield better precision. A combination of soft margin SVM and naïve Bayes filter was tested on PU1 corpus. It reported 94.4% correct classifications, 12.7% false negatives and 0.0% false positive. In comparison, the accuracy is of the basic soft SVM was 98.1%, with 1.6% false positives and 2.3% false negatives. Parameter tuning of the soft SVM reduced the false positives to 0.0%, but this resulted in a marked degradation of accuracy to 90.8% and false negatives to 21%. We thus observe that the combined SVM tackles the more serious problem of false positives while still maintaining accuracy at an acceptable level.

The main strength of the SVM is its ability to exhibit better performance even if a plethora of features is used; it self-tunes itself and maintains accuracy and generalization. Therefore, there is no compelling need to find the optimum number of features. In [11], SVM employed for spam filtering and tested on the public corpora, Trec06p/full and Trec06c/full [12] and private corpora, X2 and B2 described in the paper, gave encouraging results with an average accuracy of 91.89%, 3.95% false negatives and 2.64% false positives. Comparing various inductive learning based classifiers in [13] using the Reuters 21578 corpus [14], the authors give the best report card to linear SVM in terms of accuracy and training time. However, choice of an appropriate kernel function, high memory requirement and increasing training time with training data size are its problems.

D. *Fuzzy logic*

Fuzzy logic uses linguistic variables, overlapping classes and approximate reasoning to model a classification problem [15]. The works in [16, 17, and 18] show that fuzzy logic lends well to spam detection as indeed the classes *spam* and *non spam* messages overlap over a fuzzy boundary. Sayed *et al* employ fuzzy-based spam detection by first pre-processing the documents (removing all stop words such as 'he', 'the' and 'it' as well as HTML tags), building a fuzzy-model of overlapping categories **{spam, valid}** with membership functions derived from the training set and**,** and classifying input messages by calculating the fuzzy similarity measure between the received message and each category [16]. The authors tested their classifier with various fuzzy conjunction and disjunction operators using 4 datasets, two for training and two for testing. Averaging over the 4 cases, the best results were obtained for Bounded Diff. with an accuracy of 97.2%, spam recall of 90.5% and spam precision of 97.6%. In paper [17], Kim *et al* retained hyperlinks because spammers can minimize text but list hyperlinks. They demonstrate that feature selection by fuzzy inference is superior to conventional methods such as Information Gain. This indicates that the linguistic modeling in fuzzy logic is well-suited for both feature extraction and TC.

A good feature about fuzzy similarity based spam filtering is that it scans the content of the message to predict its category rather than relying on a fixed pre-specified set of keywords. Therefore it can adapt to spammer tactics and dynamically build its knowledge base. Fuzzy association method avoids ambiguity in English word usage by capturing the relationship or association among different index terms or keywords in the documents [18].

However, fuzzy modeling has its pitfalls in that there are many ways to interpret fuzzy rules, combining the output of several fuzzy rules and defuzzifying the output. The performance of the email filtering engine therefore needs to be optimized by experimentally fine tuning all the relevant parameters.

E. *K Nearest Neighbors (KNN)*

The KNN technique [1] proceeds by choosing first random data points as initial *seed* clusters. Next, it enters a learning phase when training data points are iteratively assigned to a cluster whose center is located at the nearest distance (*e.g.* Euclidean distance). Cluster centers are repeatedly adjusted to the mean of their currently acquired data points. The classification algorithm tries to find the K nearest neighbor of a test data point and uses a majority vote to determine its class label. The performance of KNN classifier is primarily determined by (i) an appropriate choice of *K* which can be quite tricky if either the data is non-uniformly distributed or if there is noisy data, and (ii) the distance metric applied. The value of *K* may need to be tuned for a given application.

In [19], Nakov *et* al applied latent semantic analysis and KNN classification with 10 fold cross validation to two document collections: *Ling_Spam* corpus [3] and a personal collection od emails containing 940 non spam and 525 spam messages. They achieved an accuracy of 99.65% for moderate values of *K* set to 3 and 4. In [20], the authors applied KNN to the SA2 corpus with 10 point validation. They demonstrated that when the value of *K* is set at 3, the overall accuracy is 93% with a distinct split in accuracy as 98.6% for good email and 79.8% for spam mail. With a *K* neighborhood of 21 members, accuracy improves to 94% overall, with a distinct split in accuracy as 96.1% for good emails and 90.9% for spam mail.

The main strength of the KNN algorithm is that it provides good generalized accuracy on many domains and the learning phase is fast. But it is slow during instance classification because all the training instances have to be visited. The accuracy of the KNN algorithm degrades with increase of noise in training data.

Table III presents a concise comparison of all the statistical approaches covered in the above analysis.

III. Context Based Approaches

A. *Motivation*

Statistical approaches obviate the need to analyze the contextual relevance of words in a document. But it is obvious that groups of words build up the overall context of a document. For instance, word-groups are used as indexing features in search engines. Single word variations such as synonyms for a concept and polysemous words with multiple meanings not only increase the feature space in the *bags-of–words* approach, but may also reduce the recall rate. Such problems can be overcome with *context sensitive* classification methods that essentially identify and make use of word association information to improve the classification effectiveness. They allow the context of a word to influence how its presence or absence will contribute to a classification outcome.

B. *Context Interpretation*

Context is an intuitive term connoting high level semantics. It can be interpreted along various dimensions and applied in a variety of ways. We classify the context based TC approaches on the basis of how the context has been interpreted and what features have been utilized to derive it.

1) *Latent Semantic Analysis (LSA):* LSA implicitly captures the main associative patterns between groups of words and documents using unsupervised dimension reduction through the Singular Value Decomposition (SVD) technique. It is really a statistical approach but handles word-dependencies implicitly by vector semantics.

LSA based document Indexing (LSI) works by constructing a non-linguistic vector space that helps to identify generic word associations. By itself therefore, it cannot be applied to situations requiring analysis of natural language. However, researchers have fruitfully harnessed this technique in combination with methods that cull out *concrete* references to terms in a specified set. In [30], the authors start with a category name and automatically generate a set of keywords per category name from lexical references in WordNet and Wikipedia in the form of synonyms, their hyponymns and their derivatives [31,32]. Initial classification is based on the cosine similarity function of these references and given documents. Next, false positives that may be caused by either lexical ambiguity or because of passing references in the first part are removed by a performing a general fit by employing LSA. Their results for TC using Reuters-10 and 20NewsGroup corpora [34] show improved performance by this combined approach as compared to only approaches that use only references or only context.

2) *Lexical Units:* Lexical units are co-occurring word-expressions associated with a meaning. In [23] Cohen and Singer propose a *sleeping experts* algorithm that entails a set of active lexical units called experts to predict a document's classification. Experts are groups of co-occurring words bearing a prescribed order but allowing variable gaps (arbitrary number of words) in between. A master algorithm learns appropriate weights for each expert during the learning phase adaptively and makes an overall prediction based on individual experts' predictions and a prescribed *threshold* during test phase. While there may be any number of such experts, only a few active ones actually post predictions on any given example; the remainder are said to be "sleeping" on that example.

The paper [23] also presents the *RIPPER* algorithm to construct non linear classifiers that learn lexical units as Boolean function in the form of conjunctive conditions between words in a document. *RIPPER* carries itself through two stages. Stage 1 constructs an initial rule set using a variation of IREP (Incremental Reduced Error Pruning [22]); a context sensitive algorithm that helps derive a compact set of rules that can be triggered to classify a new document. The algorithm IREP* constructs one rule at a time, removes all examples covered by a new rule, randomly partitions the uncovered examples into two subsets, two third examples comprising a growing set to add clauses to a rule and remaining one third examples comprising a pruning set to remove clauses. A rule is expanded by adding conditions that maximize the *Relative Information Gain,* a factor that measures the growth of positive examples' density, and then pruned by removing those conditions that maximizes the differential between positive and negative examples. Stage 2 optimizes the initial rule set to further improve its accuracy. Each rule either (a) *revised* by growing it further with additional literals or (b) *replaced* by another new rule that is first grown and then pruned so as to minimize the error of the *entire rule set* or (c) *retained* as such. The final choice depends upon which course of action minimizes a critical parameter called *description length.* An adjustable parameter called *loss ratio*, defined as cost of false negatives to false positives, trades off between recall and precision to guide the learning process and minimize misclassifications of new data.

The results as presented in [23] using the AP and TREC-AP corpora [12] for (i) RIPPER (ii)*sleeping experts* algorithms using four word phrases (E4) and single word phrases (E1) and (iii) a statistical linear classification algorithm called *Rocchio* [24], are summarized in Table I for ease of reference. Tests on both corpora reveal that all context-based methods report fewer errors than the statistical approach Ro. Specifically for AP Title Corpus, both Ri and E4 have higher recall than Ro. E4 also has better precision than Ro. For TREC AP Corpus, sleeping experts E4 reports the best recall and precision among all context based methods.

The authors also evaluated RIPPER (Ri), Sleeping experts E4, E3, E1 with four, three and one word phrases respectively, and Rochhio (Ro) algorithms on the Reuters-21578 [14] corpus. Table II shows the performance index *micro-averaged breakeven,* at which precision equals recall. These results clearly indicate the superior performance of context based methods as compared to the statistical approach adopted in Rochhio. They also provide an encouraging indicator to the fact that the largest group of associated words, four as in the case of sleeping experts algorithm E4, gives the best results.

TABLE I: Summary of Results on AP Title and TREC-AP Corpus

Learner	AP Title Corpus			TREC-AP Corpus		
	No of errors	Recall	Precision	No of errors	Recall	Precision
Ro	91.11	44.23	77.39	498.1	28.7	72.0
Ri	84.56	51.41	74.46	456.6	56.4	78.5
E4	80.33	49.78	81.41	439.7	57.1	80.2

E1	92.33	42.61	67.32	476.8	30.0	72.1

TABLE II: SUMMARY OF RESULTS ON REUTER-21578

Microaveraged Breakeven for Reuter-21578		
Split of corpus data set	**Method**	**Microaveraged breakeven**
ModLewis split: No of Examples: -13,625 for training -6188 for test	E3	0.769
	E2	0.753
	Ri	0.689
	Ro	0.668
	E1	0.677
ModApte split: Examples -9603 for training -3299 for test	E3	0.827
	E2	0.823
	Ri	0.819
	Ro	0.776
	E1	0.798

3) *Syntactic Constructs:* NLP makes use of syntactic structures of tokens that encapsulate grammar rules. Such structures such as Parts Of Speech (POS) and their combinations can be utilized to derive context. In [21], the authors studied the efficacy of using complex syntactic linguistic constructs as core features for context based TC. They used various concatenations of *lemma, POS* and *words dependency* or modifier. They also use IREP [22] to build a rule-base as described earlier. The newly constructed rule is then evaluated by the whole set of training data and added to the repository only if it reaches a stipulated threshold.

The authors applied various combinations of complex syntactic feature sets on large and small classes taken from the dataset Reuter-21578 [14]. Their experimental results reveal that the most complex features do outperform words as features, thus pointing towards their potential to improve performance of context sensitive text classification.

4) *Ontology and Semantic labels:* In [26], the authors propose superimposing concepts derived from background knowledge onto the classical word vector feature representation of documents that makes use of only word stems. Knowledge is derived from an ontology and context in the form of related words, syntactical patterns, morphological transformations and word sense disambiguation. They use the Adaboost Boosting ML technique [27], whereby simple rules learned by several weak learners are combined as per an additive model.

Authors evaluated their approach with experiments on the Reuters, OHSUMED [33] and FAODOC [35] corpora and utilized the WordNet, the MeSH and the AGROVOC [36] ontologies. These experiments reveal consistent improvements in the microaveraged as well as macroaveraged error rate, precision, recall, F_1 measure and Breakeven Point scores, when compared with classification with only term vectors. The authors analyze two kinds of concept integration that are responsible for the observed improvements: (1)Lexical level improvement by multiword expression detection and synonym conflation (2)Conceptual level improvement using ontology structures to *generalize* and thereby derive hypernyms and integrate them with word stems. Results further reveal that an appropriate choice of ontology affects the quality and consistency of results significantly.

In [28] semantic labels such as *who* did *what* to *whom, when, where why, how* etc. are tagged to syntactic constituents surrounding a predicate. The shallow semantic parsing of sentences extends well to applications such as question answering, summarization, information extraction. The authors employ SVM to identify each non-copula verb or *predicate* in a sentence and tag syntactic constituents with distinct semantic arguments. SVM tuning comprises a pruning process which removes NULL constituents as identified by the first binary classifier. Next, *N* One Versus All (OVA) binary classifiers classifies each of the *N* NON NULL constituents.

For training and testing, the authors use the PropBank corpus [37] which provides sentences annotated with verb predicates and their syntactic arguments. In their baseline approach, they include features such as the *predicate,* the *Path* from constituent to predicate, the *Position* of a constituent *w.r.t.* the predicate, the *Head word* etc. Results were further improved when many new innovative features such as verb clustering, named entities and head word part of speech were added. The SVM approach reports best results with 84% precision and 75% recall. However, it is also observed that the trained system worked poorly in terms of coverage on another corpus. This is mainly because of domain differences and also because the range of some of the important features such as *predicate* and *Path* is very large.

To enable classification that is independent of syntactic parsing, the authors formulated the semantic labeling problem at a word by word level, through which each word was separately tagged. Experiments reveal a distinct fall in quality of results in the word-by-word approach as compared with the constituent-by-constituent approach. This reflects that syntactical context reinforces learning of semantics.

5) *Term weighting with contextual features:* Term weighting has applications in question answering and information extraction. In [29], the authors adopt a combined approach by integrating both statistical and NLP based features to define a term weighting *context function* that progressively refines the weights of terms in a document based on the influence of surrounding words. Each term's weight or score is recursively evaluated by a combination of its current score denoting its implicit relevance and by the context function that generates the influencing score. The context function encapsulates both statistical features such as information gain, gain ratio as well as contextual features such as syntax roles of the words, POS and lexical metrics such as WordNet distances, combining all these into a single measure of influence. Its parameters are iteratively learned by an ML technique called resilient parameter adaptation.

The authors exemplify their approach with crossword clues to generate single word answers. The term weighing performance is judged by the weight-sorted position of the correct answer. The authors formulate three metrics: Mean Reciprocal Rank *(MRR)*, Success Rate *(SR)*, both discreet functions, and a third differentiable function – the *soft MRR* to measure the performance of term weighting schemes. In experiments solving the WebCrow crossword cracking QA system, the context function driven term weighting shows a significant improvement in quality of results when compared with term weighting methods that use features derived from word frequencies such as TF-IDF or use purely statistical features.

C. Discussion on context sensitive techniques:

The research and results summarized above indicate certain strengths of context sensitive TC over context independent methods.

1) Instead of relying on externally input, static set of constructs, the use of contextual information makes TC robust and more immune to noisy data. One can tap the vast knowledge accumulated and techniques available in the domain of AI-based learning methods. ML techniques specialized for TC has been reported such as IREP [21, 22, and 23] Adaboost [26, 27], weight learning algorithms [23, 29] and SVA [28]. A plethora of generic and domain specific corpora, carefully annotated [12,14,31-25] and ontological documents [35, 36] are available for training and testing classifiers. These methods and tools can be tapped for categorizing email messages.

2) Context is an intuitive and human-oriented way of text interpretation and can naturally be introduced in a variety of ways. Their application can be generic or suitable for specific problem domains. These include implicit context as captured by LSA [30], lexical sense as implied by sparse matrices [23], syntactic meanings as in POS phrases [21], semantic meanings [26, 28] and term weighting [29]. Adaptive information retrieval systems also make use of user profiles [25]. The user's web interactions and feedback such as deleting a spam or transferring a message from one folder to another, can be examined to build and dynamically adapt his/her profile and frame it as user-centric context for organizing emails. It is indeed both a challenge and a potential opportunity to cull out useful contexts from the rich space of context oriented features.

3) Experimental results reported all the papers discussed do indicate positive directions for contexts sensitive TC as discussed above. They outperform other methods either by improving upon the quality of results with reduced error rates or by ushering in larger corpuses within the ambit of solvable problems, being effective on large noisy corpora. In general it can be seen that context sensitive methods performs well across a large category of TC classification problems.

4) The variety of techniques and interpretations of contexts leads to a great possibility of combining these techniques to exploit and reinforce the advantages of each. For example, rule based methods can derive antecedents which become initial input phrases for a group-of-words based method.

5) As the number of email users explodes, it will become a necessity to use innovative methods to automatically recognize and organize the messages. Statistical methods have been used for long for email filtering and have reached a saturation point where they are unable to foil spammers' circumventing methods. Context based classification techniques can be explored for next generation email management.

Real time performance, adaptive learning and sensitivity to user-profiles are important criteria for email management. The TC model employed must have simple statistical assumptions and give linear-time performance. Techniques such as symbolic representation of features and attributes and efficient weight learning algorithms help reduce the search space. Minimal inputs from the user, such as category names should suffice to categorize incoming emails. Classifiers trained on ontology-driven semantics can be useful for domain specific classification. Dynamically adaptable learners will be needed to tune the classifier to changes in user's profile.

Table IV compares the context approach, feature set, ML technique, testing environment, main conclusions and application of the context-based TC methods.

TABLE III: COMPARISON OF STATISTICAL METHODS FOR TC

Algorithm	**Basic techniques employed**	**Strengths**	**Weaknesses**	**Highest accuracy reported (%)**
Naïve Bayes	1. Supervised learning 2. Probability based classifier.	. Simple and robust algorithm. . Independence assumption minimizes computational complexity. . Wide applicability	1.Susceptible to 2.Bayesian Poisoning and unrelated video insertion	99.99[3]
Decision Tree	1. Supervised learning. 2. Graph based classifier. *n: no of instances* *m: no of attributes*	1. 2. Simple and intuitive rule based approach 2. Can handle both continuous as well as categorical variable. 3. Missing data can be handled easily 4. Fast DT of order *O(m.n)* available	Computationally expensive for continuous variables to calculate the information gain of several values for each variable.	97.17[6]
SVM	1. Supervised learning 2. Hyperplane based classifier	There is no need to calculate all features in the training data ser to achieve desired accuracy.	Entails a long training time.	98.10[7]
Fuzzy Logic	Fuzzy rules based classifier	Same fuzzy engine can be utilized for both feature extraction and text classification.	Fuzzy modeling is difficult for discrete data.	97.20[10]
KNN	1. Unsupervised learning. 2. Neuron based classifier.	Easy to implement and modify	Performance is degraded with increase of noise in training data.	99.65[13]

TABLE IV: COMPARISON OF CONTEXT BASED METHODS FOR TC

Characteristic	Barak, Dagan & Sgnarch [30] 2009	Cohen & Singer (Sleeping Experts)[23] 1999	Bloehdorn & Hotho [26], 2004	Pradhan *et.al.*[28],2005	Ernandes *et.al.*[29], 2007	Wong, Lee & Yeung [21]
Approach for deriving Context	Keywords output by lexically referenced category names are used as terms in LSA	Sparse matrix Lexical units called experts are used. Only active experts contribute to results.	Generalization semantics are derived from an Ontology with Lexicon	Semantic labels (who, what whom, where when etc) are derived by shallow semantic parsing	Term weighting done by a parametric Context Function learned by ML	Rules are learned to make Boolean combination of words
Feature set	Synonyms, Hyponymns, derivatives, Holonyms, meronymns,	Sparse set of ordered words	Detecting Candidate term, POS, morphology	Predicate, Path, Position, Head Word, POS, voice sense	A)Statistical: Word frequencies, Distances, Information Gain B)Linguistic:POS, Syntax, lexical	Concatenation of Lemma | POS | Modifier
ML technique and optimization	Unsupervised dimension reduction by SVD	Master algo for learning weights of experts	Boosting on weak learners	SVM with OVA classifiers	Resilient Parameter Adaptation	Learning rules IREP*
Corpus & Parameter used for training and testing	Reuters-10, 20NewsGroups Precision, Recall, F1	Rueters 21578 Micro averaged breakeven	Rueters 21578, OHSUMED, FAODOC Ontologies: MeSH, AGROVOC Classification Error, precision, recall, F1	PropBank Precision, Recall, F1	WebCrow Soft Mean Reciprocal Ratio, MRR and SuccessRatio SR	Rueters 21578 Recall, Precision
Main Conclusion	Combining referenced categories with LSA improves result compared with using only references or only LSA.	Microaveraged Breakeven is best for experts with more words as compared with statistical method	Consistently, improves results when concepts derived from ontologies are combined with terms	SVM allows new features such as verb clustering, to gives best semantic argument identification as compared with past methods.	Context based algorithms outperform frequency based algorithms for term weighting	TC with Complex concatenation of syntactic features outperform single words
Applications	Category name based or keyword based TC.	General TC applications.	Ontology based TC.	Question Answering systems	Single answer QA systems	General TC applications.

IV. CONCLUSION

This paper presented a quantitative as well as qualitative comparative evaluation of existing text classification techniques with focus on email filtering and potential application to general email management. We presented the accuracy results of different text classifiers on different data sets for spam filtering. Significant work has been done in the field of statistical text classification and their results have indeed been applied to a wide range of web applications. The direction now points towards extracting correlation among words, *i.e.* context based approaches. Several of heuristics have recently been proposed for context based text classifiers. There is yet more scope for future research in the promising field of context based TC by applying proven learning methods, introducing useful contexts and combining techniques symbiotically. It is worth examining these techniques for the challenging application of organizing emails

REFERENCES

[1] Machine learning An Algorithm Perspective, Stephen Marsland, Chapman & Hall/CRC, 2009

[2] Naïve Bayes Classifier, Wikipedia, http://en.wikipedia.org/wiki/Naive_Bayes_classifier

[3] I. Androutsopoulos et al, "Learning to filter spam email: a comparison of a naïve Bayes and a memory based approach," Procs of the workshop "Machine Learning and Textual Information Access", 4th European Conference on Principles and Practice of Knowledge discovery in Databases, 2000.

[4] Johan Hovold, "Naïve Bayes Spam Filtering Using Word Position Based Attributes", International conference of Email and Anti spam, 2005.

[5] R. O. Duda; P. Hart; David G.Stork, Pattern Classification 2nd edition, Wiley.

[6] Ron Kohavi, "Scaling up the Accuracy of Naïve Bayes Classifier: a Decision tree Hybrid", Data mining and visualization, Silicon Graphics, Inc., 1996.

[7] S. Youn and D. McLeod, "Efficient Spam Email Filtering using Adaptive Ontology," Procs of the Intl. Conf. on Information Technology, Pages 249-254, 2007.

[8] Gehrke, J. E., Ramkrishnan, R. and Ganti, V. "Rainforest- A framework for fast decision tree construction of large datasets", Procs of the 24th VLDB Conference New York, USA, 1998.

[9] Jiang Su; Harry Zhang, " A Fast Decision Tree learning algorithm", Procs of the 21st conf. on AI-Vol. 1 Boston, Pages 500-505, , 2006.

[10] K. Tretyakov, "Naïve Bayes Spam Filtering Using Word Position Based Attributes", Machine Learning Technique in Spam Filtering, Data Mining Problem oriented Seminar, MTAT.03.177, pp. 60-79, May 2004,

[11] Q. Wang, yi Guan, X. Wang, "SVM Based Spam Filter with Active and Online Learning", In Procs. of the TREC Conference, 2006.

[12] TREC Data Collection http://trec.nist.gov/data.html

[13] Dumais, John Platt, Mehran Sahami, David Hekerman, "Inductive learning algorithm and representations for text categorization", n Procs of the 7th Intl conf. on Information and Knowledge Management, 1998.

[14] Reuters-21578 Text Categorization Collection, http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[15] L. A. Zadeh, "The Concept of a Linguistic Variable and Its Application To Approximate Reasoning-I", Information Science, 8 pp. 199-249, 1975.

[16] El-Sayed M. El-Alfy, Fares S. Al-Qunaieer, "A Fuzzy Similarity Approach for Automated Spam Filtering", Procs of the 2008

IEEE/ACS International Conference on Computer Systems and Applications - Volume 00, Pages 544-550, 2008.

[17] Jong-Wan Kim and Sin-Jae Kang, Y. Hao (Eds.): "Feature selection by Fuzzy Inference and its Application to Spam mail filtering", CIS '05, Part I, pp. 361 – 366, Springer-Verlag Berlin Heidelberg, 2005.

[18] Fuad, Deb, Hossain, "A Trainable Fuzzy Spam Detection System", PDCN 2005, Innsbruck, Austria, Pages 399-404, 2005.

[19] P.I. Nakov, P.M. Dobrikov, "Non–Parametric Spam Filtering Based on KNN and LSA", Procs of the 33th National Spring Conference, 2004

[20] W.Yerazunis, C.Siefkes, S.Chhabra, "Sorting Spam with K-nearest –neighbor and Hyperspace Classifiers". http://crm114.sourceforge.net/docs/KNN_Hyperspace_Filters.pdf,

[21] A.K.S. Wong, J.W.T.Lee, D.S.Yeung, "Using complex linguistic features in context sensitive text classification techniques", Proc. Of 2005 Instl Conf. on Machine Learning and Cybernetics, Vol 5, pp 3183-3188,2005.

[22] Furnkranz J. and Widmer G. "Incremental reduced error pruning, Proc of the 11th annual conference on machine learning, New Bruncwick, NJ, Morgan Kaufmann Publishers Inc., San Fransisco, CA, 1994.

[23] William W. Cohen and Y. Singer, "Context Sensitive Learning Methods for Text Categorization," ACM Transaction on Information Systems, Vol 17 No.2, Pages 141-173, 1999.

[24] Rocchio, J, "Relevance feedback information retrieval", Science 253, pp 974-980, 1971.

[25] Young-Woo Park Eun-Seok Lee, "A new generation method of a user profile for information filtering on the Internet" Procs of 13th Intl. conf. on Information Networking, 1998, Tokyo, Japan, 1998.

[26] S. B. Andreas and A. Hotho, "Boosting for Text Clssification with Semantic Features," In Procs. of the MSW 2004 Workshop at the 10th ACM SIGKDD Conf on Knowledge *Discovery and Data* Mining, 2004.

[27] R.E. Schapire and Y. Singer, "A boosting-based system for text categorization," Machine Learning, 39(2/3), pp 135-168, 2000.

[28] S. Pradhan *et.al.,* "Support vector learning for semantic argument classification," *Machine Learning,* 60, pp 11-39, 2005.

[29] M. Ernandes *et. al.*, "An adaptive context based algorithm for term weighting- application to single-word question answering," *Proc. Of Intl. Joint Conf on* Artificial *Intelligence (IJCAI) 2007*, pp:2748-2753, 2007.

[30] L. Barak *et.al.* "Text categorization from category name via lexical reference," *Procs of NAACL HLT 2009* pp33-36, June 2009.

[31] WordNet, http://wordnet.princeton.edu/

[32] Wikipedia,www.wikipedia.org/

[33] OHSUMED, ftp://medir.ohsu.edu/pub/ohsumed 20NewsGroups, http://www.ai.mit.edu/people/jrennie/20Newsgroups/

[34] FAO Document Repository, www.fao.org/documents/

[35] Agricultural Ontology Service, www.fao.org/agris/aos/

[36] Palmer M, Kingsbury P, Gildea D (2005). "The Proposition Bank: An Annotated Corpus of Semantic Roles". *Computational Linguistics* **31** (1): pp71–106