How Much Noise is too Much: A Study in Automatic Text Classification

Sumeet Agarwal*
IIT Delhi
asumeet@cse.iitd.ernet.in

Shantanu Godbole
IBM India Research Lab
shantanugodbole@in.ibm.com

Diwakar Punjani*
Carnegie Mellon Univ.
diwakar@cs.cmu.edu

Shourya Roy
IBM India Research Lab
rshourya@in.ibm.com

Abstract

Noise is a stark reality in real life data. Especially in the domain of text analytics, it has a significant impact as data cleaning forms a very large part of the data processing cycle. Noisy unstructured text is common in informal settings such as on-line chat, SMS, email, newsgroups and blogs, automatically transcribed text from speech, and automatically recognized text from printed or handwritten material. Gigabytes of such data is being generated everyday on the Internet, in contact centers, and on mobile phones. Researchers have looked at various text mining issues such as pre-processing and cleaning noisy text, information extraction, rule learning, and classification for noisy text. This paper focuses on the issues faced by automatic text classifiers in analyzing noisy documents coming from various sources. The goal of this paper is to bring out and study the effect of different kinds of noise on automatic text classification. Does the nature of such text warrant moving beyond traditional text classification techniques? We present detailed experimental results with simulated noise on the Reuters-21578 and 20-newsgroups benchmark datasets. We present interesting results on real-life noisy datasets from various CRM domains.

1 Introduction

The importance of text mining applications is growing proportionally to the exponential growth of electronic text. Along with the growth of the Internet, many other sources of electronic text have become really popular over the last decade. With the Internet penetrating the lives of more and more people, email, chat, newsgroups, blogs, discussion fora etc. have become very popular and they generate a huge amount of text data everyday. Other big contributors to the pool of electronic text documents are call centers and CRM organizations that have text in the form of call logs, problem tickets, complaint emails, electronic text generated by Optical Character Recognition (OCR) processes, on hand-written or printed documents, conversational data converted automatically to text, and mobile text such as Short Message Service(SMS).

Though the nature of these documents is varied, all of them share a common effect - the presence of textual noise. Text produced under such circumstances is typically highly noisy containing spelling errors, abbreviations, non-standard words, false starts, repetitions, missing punctuations, missing letter case information, pause-filling words (like *um* and *uh*), and other text and speech disfluencies. Very often such data requires cleaning and preprocessing before applying state-of-the-art text analytics techniques. *Noisy Text Analytics* is defined as a process of information extraction whose goal is to automatically extract structured or semistructured information from noisy unstructured text data[1]. However one of the commonly used text mining applications, quite different from extraction of information, is *text classification* or *text categorization*.

The text classification task is one of learning models for a given set of classes and applying these models to new unseen documents for class assignment. Text classification has many important real life applications. For example, categorizing news articles according to topics such as *politics*, *sports*, or *education*; email categorization; building and maintaining web directories like Dmoz; spam filters; automatic call and email routing in contact centers; pornographic material filters and so on. Two types of classifiers are commonly employed viz. statistical and rule based classifiers. In statistical classifiers a *model* is learned on a corpus of pre-labeled data, and once trained, the system can be used for automatic assignment of labels to unseen

*Work done while working at IBM India Research Lab

[1] http://en.wikipedia.org/wiki/Noisy_text_analytics

data. Rule based classifiers, on the other hand, are good at finding class boundaries based on presence or absence of words and/or phrases. In both statistical as well as rule based text classification techniques, the content of the document is traditionally the *sole* determiner of the category to be assigned. However noise in the text distorts the content and hence users can expect the categorization performance to get affected. Classifiers are essentially trained to identify correlations between extracted features (words) and different categories which can be later utilized to categorize new documents. For example, email containing text like *exciting offer, get a free laptop* might have a stronger correlation with the category *spam* emails than *non-spam* emails. Noise in text distorts this feature space as *excitinng ofer get a tree lap top* will be a new set of features and the categorizer might not be able to relate it to the *spam emails* category. The feature space explodes as the same feature can appear in different forms due to spelling errors, poor recognition and wrong transcription. Noisy text categorization in particular has important practical applications in the form of problem determination in contact centers, call routing, categorization of hand-written customer complaints and automatic SMS routing.

There is another kind of noise apart from feature noise which often gets ignored in the text classification setting. The class label(s) associated with a document is(are) always assumed to be sacrosanct. Often, and as we will see in Section 4, the labeling of documents (class assignments) is uncertain and error-prone. This has classically been called **label noise** and it has been studied in the context of assuming a *s*mall amount of noise in the label assignment in a classification problem [27]. We will see that in many real life domains where classification is important, label noise actually crosses tens of percentage points casting a question on the supervised learning setting itself. In this paper, while we focus on feature noise, in the experimental section we will also look at label noise and point out potential research directions arising from our observations.

Our Contribution: In this paper we will show the effect of different kinds of noise on text classification performance by doing detailed experiments on synthetic as well as real-life noisy datasets. Here, we are reporting our observations based on experiments and not proposing any new method to combat noise in text for classification. Our experiments show that text classification algorithms are quite robust even in the presence of a high degree of typographical noise or noise introduced by Automatic Speech Recognition (ASR) systems. We feel such work is an important pre-requisite to better study noisy text classification. The rest of this paper reviews related work (§2), describes noise in text (§3), and describes our experimental study (§4) before conluding (§5).

2 Related Work

In this section, we present some relevant work in the following related areas - (1) noisy text analytics, (2) text classification and (3) noisy text classification.

Noisy Text Analytics: There has been a lot of work on analyzing noisy text; two prominent areas being automatically correcting noisy text and information extraction from noisy text. A comprehensive survey of techniques pertaining to detecting and correcting spelling errors in text can be found in [11]. There is some recent work on correcting the output of SMS text [4], OCR errors [18] and ASR errors [23]. Information Extraction (IE) aims to automatically extract structured information from unstructured documents. The task becomes non-trivial in the presence of noise. Authors in [16] have shown how to automatically annotate noisy postings on the internet. A study on the performance of the gene name annotator when trained on noisy data can be found in [25]. Authors in [17] have measured the effect of OCR noise on IE performance. It has been shown that in the absence of punctuations, extraction of different syntactic entities like parts of speech and noun phrases is not accurate [19]. It has been shown that it is possible to build aggregate models from ASR data [22] by extracting important words and phrases from automatic transcriptions of telephonic conversations between customers and contact center agents.

Text Classification: The two broad types of classification methods used are discriminative and generative methods. Discriminative methods like SVMs [10] or logistic regression (LR) [26] are two-class classifiers that find separators between documents of two classes in some space of representations. Other discriminative models include maximum entropy methods [20] and boosted decision trees in the ADABoost framework [7]. Generative methods are typified by naive Bayes (NB), aspect model [9], Latent Dirichlet Allocation [3]. Discriminative methods are widely accepted to be more accurate. The industry has also made significant advances in the development and deployment of real-world high-performance text classification systems [14] using combinations of rule-based, hand-tuned, and statistical techniques. However, not all the techniques used in commercial systems are publicly known, and few general principles can be derived from these systems.

Noisy Text Classification: Electronically recognized hand-written documents and documents generated from OCR process are typical examples of noisy text. Authors in [24] have studied the characteristics of noise present in such data and its effects on categorization accuracy. A generic system was proposed in [2] for text categorization based on statistical analysis of representative text corpora. They evaluate their system on the tasks of categorizing abstracts of paper-based German technical reports and business letters concerning complaints. They claim that the tasks achieve recognition scores of approximately 80% and are very robust against recognition or typing errors. OCR systems essentially produce word substitutions while ASR systems give rise to word substitutions, deletions and insertions. Moreover, ASR systems are constrained by a lexicon and can give as output only words belonging to it, while OCR systems can work without a lexicon (this corresponds to the possibility of transcribing any character string) and can output sequences of symbols not necessarily corresponding to actual words. Such differences are expected to have a strong influence on the performance of systems designed for categorizing ASRed documents in comparison to the systems for OCRed documents. We are not aware of any work dealing with ASR document categorization, it's relevant issues and experimental results, though researchers have looked at call-type classification [8].

3 Noise in Text

We define noise as *any kind of difference in the surface form of an electronic text from the intended, correct or original text*. We see such noisy text everyday in various forms. Each one has characteristics unique to it and hence requires special handling. Some domains of noisy text data are:

- **On line Noisy Documents:** Emails, chat logs, newsgroup postings, threads in discussion fora, blogs, scrapbook entries, etc. fall under this category. People are less careful about the lexical accuracy of written content in such informal modes of communication. These are characterized by frequent misspellings, commonly and not so commonly used abbreviations, incomplete sentences, missing punctuations and so on. Automatic routing of emails and topical catergorization of newsgroup postings are typical applications for such documents.
- **SMS**: Short Message Services is becoming very common. Language usage over SMS texts significantly differs from the standard form of the language. An urge towards shorter message length facilitating faster typing and the need for semantic clarity, shape the structure of this non-standard form known as the *texting language* [4]. Automatic classification of SMSes sent to service providers to gather business intelligence is an important application.
- **Text Generated by ASR Devices:** Automatic Speech Recognition (ASR) is the process of converting a speech signal to a sequence of words. An ASR system takes speech signals such as monologues, discussions between people, telephonic conversations, etc. as input and outputs a string of words, typically not demarcated by punctuations, known as a *transcript*. An ASR system consists of an acoustic model, a language model and a decoding algorithm. The acoustic model is trained on speech data and their corresponding manual transcripts. The language model is trained on a large monolingual corpus. ASR converts audio into text by searching the acoustic model and language model space using the decoding algorithm. Automatic call routing, problem identification depends heavily on accurate categorization of ASR transcripts.
- **Text Generated by OCR Devices:** Optical character recognition, or OCR, is a technology that allows digital images of typed or handwritten text to be transferred into an editable text document. It takes a picture of text and translates the text into Unicode or ASCII. For optical character recognition on hand-written text, the rate of recognition is 80% to 90% with respect to clean handwriting. OCR systems give rise to some typical substitution errors such as *iii* instead of *m*, *5* instead of *s* etc. Categorization is essential for segregation of handwritten and printed materials based on topics.
- **Call Logs in Contact Centers:** Today's contact centers (also known as call centers) are increasingly contributing to the pool of noisy text by the means of *call logs*. Contact center agents are expected to record summaries immediately after completing interactions with customers before starting the next. As the agents work under immense time pressure, the summary logs are very poorly written and sometimes even difficult for humans to interpret. Analysis of such call logs is important to identify problem areas, evaluate agent performance, predict evolving problems etc. Contact center interactions also produce a huge amount of unstructured data in the form of emails, call transcriptions, SMS, chat transcripts etc. Automatic classification has many applications in contact centers such as problem identification, atuomatic routing, customer satisfaction analysis.

Next, we present the setup and results of our experimental study looking at textual noise and it's effect. We have used data from the domains outlined above and also conducted detailed experiments with benchmark datasets containing different types of simulated noise.

4 Experiments

This section describes our detailed experimental evaluationWe evaluate the performance of standard text classification algorithms on multiple datasets in different settings. We use *rainbow* from the BOW toolkit [15] for multinomial naive Bayes (NB) classifiers and SVM-Light [10] for Support Vector Machine (SVM) classifiers. These classifiers represent the spectrum of generative and discriminative models and are the most often used learners for their ease of use and state of the art accuracy respectively. Whenever we performed feature selection, we used the information gain measure; it is very widely used and known to be as good as other statistical measures.

4.1 Datasets

We now describe the datasets used in our evaluations. We used real-life datasets from a few contact centers and created some synthetic datasets from benchmark text classification datasets by injecting noise. The objective was to see the variation of classification performance with noise on synthetic datasets as well as validating the propositions on real-life datasets. For each dataset used, we summarize its domain and statistics.

We list below a spectrum of textual domains with noise of varying characteristics in them. We use datasets from these domains in our study.

- News Articles - These are typically edited and well-written by reporters with minimum noise - e.g the Reuters-21578 dataset.
- Newsgroup data - Rampant on the web, these are typically well written by users but have some noise due to carelessness - e.g. the 20-newsgroups dataset.
- Email - These are typically quite well written but can contain lots of noise depending on the author - e.g. CCMail.
- Contact center data - Generated at contact centers and in CRM practices - quite noise as these contain agent summaries and customer comments to name some - e.g. CCSum and CCFb datasets.
- SMS - Very noisy data and often barely English.

Intuitively, the amount of noise present increases as we go down this list but we don't exactly quantify this here.

4.1.1 Real-life data

Contact Center agent summaries: This dataset is collected from a contact center for a telecommunications company. It contains call-logs for around 25,000 customer calls made to the contact centers; for each call, there are some structured data fields, plus a summary of the call content typed in by a human agent. Each call is also manually classified into two categories; a high-level (chosen from amongst 7 categories), and the other a more precise marker of the complaint (chosen from amongst 100 categories). In this dataset, noise is naturally introduced by the human agents when entering in the call summaries, since these have to be done under great time pressure. Thus the summaries contains many spelling errors and abbreviated forms of words. We denote this dataset **CCSum**. A real example[2]: (Agent1) /01/06/2005/ - SPK TO (CustName) BILL NOT RECD (PhoneNo) THE COMMUNICATED SLA TO SUBSCRIBER IS 02/06/2005 05:46:00 PM (Place1)COURIER (Place2)/2/6/2005 -D BILL DELVERD & RECIVED BY (Recepient) DATE 02/06/2005......(Agent2).

Contact Center customer feedback: This dataset is collected from multiple accounts of a contact center for various kinds of companies (telecom, eCommerce and web services). It contains nearly 10,000 customer feedback records from each of the 3 businesses; each record has multiple fields which are entries from a feedback form filled in by a user after concluding an interaction with an agent at the contact center. The key field is the *verbatim*, which is free-form text, and is used by human labelers to classify the customer's complaint under one of a set of around 10 to 40 categories indicating the broad reason for the customer's dissatisfaction. Example categories include *Communication problems* (where the customer is the not happy with, say, the agent's accent) and *Time Adherence problems* (complaints about delays in resolving issues). In this dataset, there is substantial noise arising out of various spelling and grammatical errors made by customers while filling up the feedback forms. In addition to this, there is also significant *label noise* (i.e., the labels assigned by the human labelers are inconsistent in their definitions), due to substantial vagueness and overlap in the way the semantics for each category are set out. We will discuss label noise after presenting experimental results. We denote this dataset **CCFb**. A real example: look at muy pasrticular problem and give me a response directed towards my problem rather than a generic answer.

Contact Center email: This dataset is collected from the contact center e-mail process for a financial services company. It contains records of about 30,000 email interactions between customers and contact center agents. Based on the initial e-mail sent by the customer, each interaction is manually classified into one of over a hundred different categories, indicating the precise nature of the customer's communication. There is some noise here due to typographical and other kinds of errors made whilst typing e-mails. We denote this dataset **CCMail**. A real example: I am moving to (Place1) from (Place2) as i am going to join in FIG commodities division of (BankName) center office.Please send all my statements to the address which i shall confirm u before next week end. If possible please send a statement dated 24th january by mail to this mail id or to the following address where my parents resides for this jan only.

4.1.2 Benchmark Datasets

We now summarize the system setup used to introduce two types of noise in benchmark datasets; (1) spelling errors and

[2] We have encrypted confidential details inside parenthesis.

(2) ASR errors. Following this, we describe the Reuters-21578 and 20-newsgroups datasets we used.

Spelling Error Simulation : We developed a program to introduce spelling errors in a text data corpus, *SpellMess* for creating synthetic datasets. SpellMess can be customized to introduce *Damerau-type errors*, i.e., insertion, deletion or substitution of a letter or transposition of two letters [5]. It requires two configuration files - (i) *KBMatrix* encoding the keyboard layout in a system understandable format so that the probability of a key getting pressed instead of the intended one can be computed. We assume any of the 8 surrounding letters can be substitute a letter by a wrong keypress, but the two letters on either side in the same row have more chance of getting substituted. (ii) *Weights* containing overall error probability and probability of 5 different types of errors viz. insertion, deletion, transposition, substitution and duplication[3]. For example, one can specify the overall error probability to be 0.25 and individual probabilities of each of the 5 types of errors to be 0.2. In that case, given a text file, 25% of the words (randomly chosen) will be misspelt by any of the 5 equally likely methods. We consider only words of length more than two characters for the purpose of injecting errors.

Automatic Speech Recognition System: We used the automatic speech recognition system developed by IBM Research [1] for generating ASR versions of documents. The acoustic models of the system were built using about 100,000 utterances by 500 speakers which amounted to about 120 hours of speech data. For acoustic front-end processing, 13-dimensional cepstral vectors [21], each representing a 25 msec duration of speech at every 10 msec were used. First and second-order derivatives are used to capture the dynamics of speech variation and hence a 39-dimensional vector is used to represent speech in the cepstral domain. 9 frames (four previous and four forward frames) of cepstral vectors were concatenated. This forms a 117-dimensional vector on which dimensionality reduction (LDA) [6] was applied to form a 39-dimensional vector. The Language Model has been trained on a text corpus of 10 million words that represents text from different categories. It consists of a trigram model with an open vocabulary and an unknown word probability of 0.00025.

Reuters-21578: This text classification benchmark dataset is collected from Reuters newswire articles[4]. Articles may belong to multiple categories simultaneously. The 10 most populated classes of this dataset are typically chosen in literature for supervised learning experiments. We also choose the 90 class subset of this dataset; classes chosen have at least one training and one test document. We denote these sets as R10 and R90. These R10 and R90 subsets of the dataset have emerged as well accepted standards for experiments among researchers.

In this dataset, the base level of noise is virtually zero, since the articles have been revised and proof-read. So, in order to estimate the effect of noise, we artificially introduce varying levels of noise in the data and see how it affects the accuracy of automatic classification. Two kinds of artificial noise are introduced: spelling errors as described in Section 4.1.2, ranging from 0-100% of the words in the corpus; and noise introduced due to ASR transcription as described in Section 4.1.2 (these transcriptions were generated only for a subset of 200 documents; 20 from each of the top 10 classes). These generated transcripts are made available for download from the UCI KDD archive for the benefit of the noisy text analytics research community[5]. Figure 1 shows an example from the R10 test set a document changing with varying amounts and types of noise. Note how words are transformed beyond recognition as noise increases. We also note that the ASR transcript contains valid English words, however proper nouns, especially regional ones, are always wrongly identified. Such words are replaced with approximations because the speech models are not trained on specialized vocabularies.

20-newsgroups: This text classification benchmark dataset is collected from on line newsgroup postings; there are about 20,000 documents evenly distributed across the 20 newsgroups[6]. In this dataset, the level of noise is quite low though still not as clean as Reuters; these postings are typically more carefully written and revised than any of the other real-life datasets mentioned above. Here too, we introduce artificial noise to see how it affects accuracy. We denote this dataset 20NG.

4.2 Results

We report results of our experimental study in this subsection. All results are using the NB and SVM classifiers on specified train-test splits. In a classification problem, the classification system is trained on the training data and effectiveness is measured by accuracy on test data which is the fraction of correctly predicted document–class mappings. We report micro-averaged accuracy in this section. A detailed discussion of various evaluation measures can be found in [13]. Our aim here is not to compare algorithms,

[3] Several other types of errors such as *run-ons, splits* can also be injected. However we believe the effect of those on classification will also be similar to errors mentioned.

[4] Available at `http://www.daviddlewis.com/resources/testcollections/`

[5] Available at `http://kdd.ics.uci.edu/databases/reuters_transcribed/reuters_transcribed.html`

[6] Available at `http://people.csail.mit.edu/jrennie/20Newsgroups/`

Original: Sumitomo Bank Ltd is certain to lose its
status as Japan's most profitable bank as a result
of its merger with the Heiwa Sogo Bank, financial
analysts said.
40% noise: sumitomo bank ltd is certain to lose its
stxtus as Japan's mozt profitable babk as a ressult of
its merger with the heiwa sogo bank fianncial analysts
said
70% noise: sumitomo bakn ld is certan to loes is
satus as Jpaan's mpst profitbale bank as a reqult
of its meregr with thye heiwa sogo bakn financial
analystrs sazid
100% noise: sumtomo bnk ld is ceetain to loes is
sta6us as Jpaan's mst proifitable bagk as a rexult
of igs mergfer wih thye heiuwa soogo bxnk fnancial
analy5sts sasid
ASR Transcript: soon is certain lose its status as
chip warns most profitable bank cuts result of its
merger with the high were so woman financial analysis
said

Figure 1. Snippet of a Reuters document with varying levels and types of noise

models, and their effectiveness; rather we want to study the effect of *feature noise* in detail.

In Figure 2 we see the accuracy of R10 containing varying amounts of noise described in Section 4.1.2. The NB classifier used here was trained on the original training set without any introduced noise. We conducted experiments with the test set corrupted with 0% to 100% noise (in steps of 10%); for brevity we report results only at $0\%, 40\%, 70\%, 100\%$ noise.

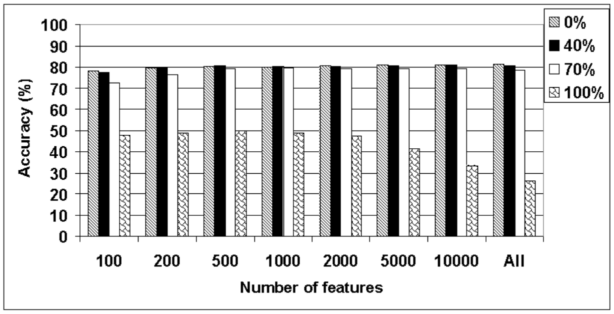

Figure 2. R10 - train clean, test noisy

To our surprise we see that even at 40% noise (i.e. on an average 4 out of every 10 words are misspelt), there is little or no drop in accuracy for different numbers of features selected by information gain. The accuracy drops at 70% noise, though only slightly. The accuracy drops significantly at 100% noise – at this level of noise, every word in the test corpus has a spelling error, rendering these words very different from those encountered during training. For this dataset, we also ran SVMs in one-vs-others configuration and achieved very good accuracy numbers. As per traditional use of SVMs we did not perform feature selection and left learning of feature weights to SVM's optimizer. At $0\%, 40\%, 70\%, 100\%$ test noise, the accuracies were 86.2%, 85.1%, 81.4%, and 39.3% respectively – the absolute numbers being higher than NB as per traditional text classification wisdom.

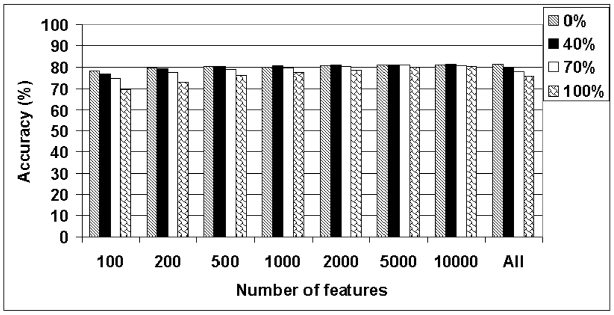

Figure 3. R10 - train noisy, test noisy

In Figure 3 we repeated the above experiment with the difference that noise (of varying degrees) was also introduced in the training set. The previous experiment is justified in the setting that clean training data for a setting might be available (it is possible to expend resources to build clean domain models), while data to be classified during deployment or testing may be noisy. The current experiment tries to ascertain if there are consistent patterns in the noise that may be learned to help in classification. As we see from the figure, this is not true. Noisy training data leads to worse off models leading to slightly lower accuracies. This is not unexpected, however, once again, the relationship with the amount of noise in training and test data is interesting. Feature selection proves to be very important in this case. Note how even 40% noise leads to low accuracies at the suboptimal (small) number of features. At about 5000–10000 features, even 70% noise leaves enough patterns to learn in the training data. One observation comparing these results to the previous set, is that even at 100% noise the accuracy degradation is graceful. We suspect this has to do with the similar nature of noise creeping in during training in this experiment.

For this setting, the four accuracy numbers for SVM were 86.2%, 86.4%, 84.8%, and 83.5%. Note that even at 100% training and test noise, SVMs essentially learnt the random pattern in the noise (similar corruptions of short words and unchanged very short words) for classification. The second kind of noise that was introduced for R10 was that caused due to errors made by an Automatic Speech Recognition (ASR) system, as described in Section 4.1.2. The objective of this experiment was to see effect of ASR (a different *kind* of noise compared to spelling errors).

A fair comparison can only be done if we create a parallel corpus for which we already have classification accuracy numbers on the clean dataset. The models trained on the training set were then tested both on the original subset of 200 documents, and on the set of their ASR transcripts. The results are shown in Figure 4.

The accuracy of an ASR system is commonly measured as Word Error Rate (WER), which is derived from the Levenshtein distance [12] and works at the word level instead of the character level. WER can be computed as $WER = \frac{S+D+I}{N}$, where S is the number of substitutions, D is the number of the deletions, I is the number of the insertions, and N is the number of words in the reference.

In this case, even though the word error rate is very high at 66.67%, there is evidently only slight drop in accuracy. This suggests that enough of the key discriminating features between classes get retained in the transcripts, even as a lot of rarer and less relevant words may be corrupted. We would like to note again that proper nouns most likely get recognized as other words during ASR.

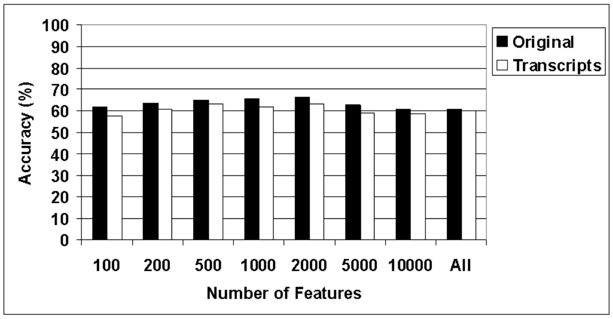

Figure 4. R10 - train clean, test transcripts

These experiments clearly show that text classification does not seem to be very susceptible to feature noise as long as the corpus is large. For small corpora, clearly even a little noise will disturb the training and test distributions significantly, violating classification's central assumption of similar train and test distributions. These experiments prompted us to investigate the exact relationship between noise, abundance of common features, statistical feature selection, and sparsity of the text classification vector space. *We will return to this investigation in 4.3 after summarizing results for all the other datasets.*

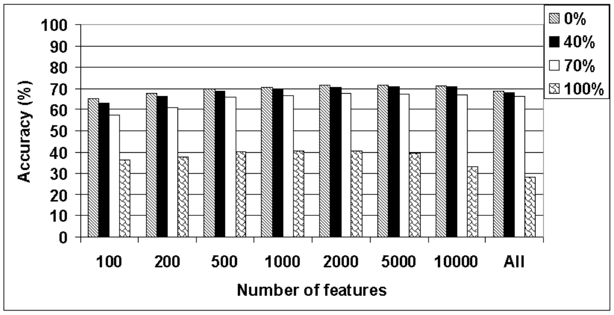

Figure 5. R90 - train clean, test noisy

In Figures 5 and 6, the same two experiments were performed with the Reuters 90-class subset dataset. For this dataset too, the observations were similar. When clean training data was used, there was only a small drop in accuracy at 40% noise; the drop became prominent at 70% and 100% noise as expected. This is also consistent with our discussion above with the R10 dataset and other effects like importance of feature selection when noise is present during training. Again, as is well known, for this dataset too SVM outperformed NB in terms of accuracy. For clean training data, the noisy test accuracies (noise at 0%, 40%, 70%, 100%) were 85.6%, 82.9%, 77.7%, 38.3% respectively. For noisy training, the noisy test accuracies were respectively 85.5%, 82.5%, 79%, 75.9%.

Figures 7 and 8 show the graphs for the same settings for the 20-newsgroups dataset. Once again our observations are similar – the marked difference being the lower absolute accuracy values. The Reuters data is known to be easy to classify given a few terms while the 20NG dataset is a little more noisy. It covers a broader spectrum of topics and has a wider vocabulary because the articles are newsgroup postings, not reviewed for quality.

Figure 9 shows the test accuracies of a wide range of real-world noisy text classification datasets. These datasets have been described and characterized earlier. The main observation in these graphs is that achievable accuracy levels vary drastically with the domain in question, irrespective of the noise perceived to be present in the domain's documents. It would seem that agent summaries of contact center interactions would be the noisiest to classify since they are written under severe time constraints. We achieved text classification accuracy of 85.9% at the first level of the hierarchy of labels (7 categories) and as much as 82.6% accuracy when considering the second level of the hierar-

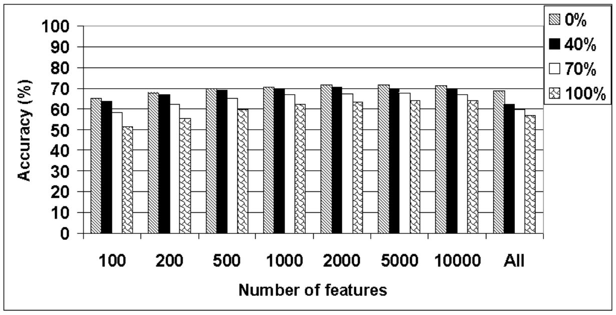

Figure 6. R90 - train noisy, test noisy

chy (100 categories). Accuracy with SVMs for first level touched 88.3%.

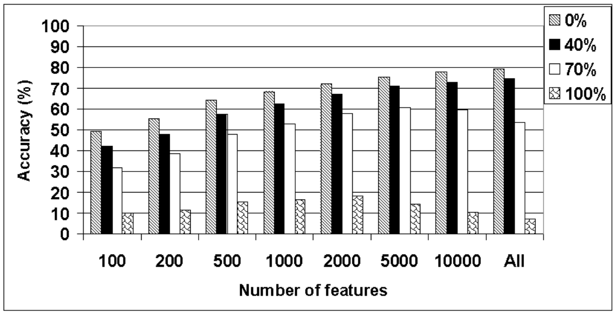

Figure 7. 20NG - train clean, test noisy

We would like to point out an important difference between the classification setting for these datasets against our train-on-noisy and test-on-noisy simulation on the benchmark datasets. In these real-world datasets, noise of at least some kinds tends to be uniform. Customers and agents alike use standard abbreviations and make common spelling mistakes unlike the other situation where spelling errors are introduced randomly.

The results on these datasets are more instructive, but the best approximation to study such effects in benchmark datasets was to perform experiments in the two settings we described above. We would like to mention that we did not perform hierarchical classification but treated the first and second levels of the hierarchy as flat label-sets. In this domain it was not clear if the hierarchy of labels was constructed for convenience or if it had been factored into designing the label-set. Without loss of generality we used the first and second levels of the hierarchy for experiments. We expected the email domain to be the cleanest

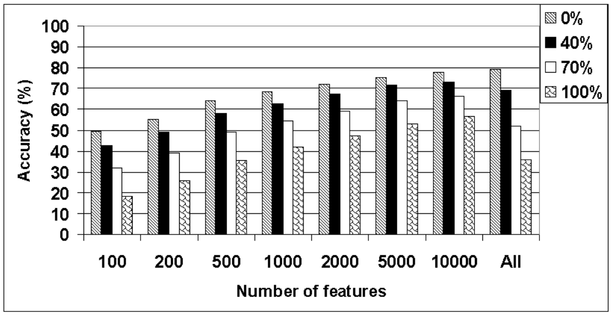

Figure 8. 20NG - train noisy, test noisy

in terms of quality of language. While this was true, the problem in this domain was the very large number of categories defined. The process of handling email complaints in typical contact centers necessitates on the fly definitions of categories with obvious overlap and redundancy leading to a bad label-set from a classification perspective. We restricted our attention to only those 50 categories with over a 100 emails. This domain's dataset was not a cleanly defined classification problem. However, we found it instructive to run text classification experiments in this interesting domain from a noise point of view. We achieved 60.1% accuracy with NB for this dataset, and 65.6% with SVMs.

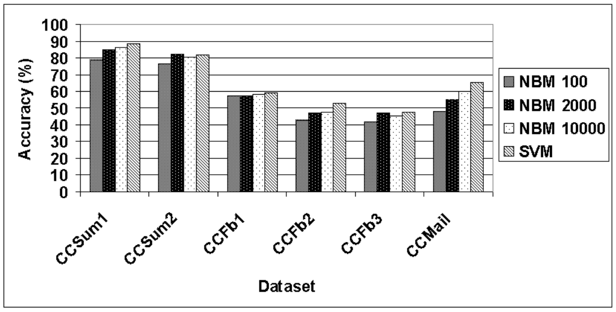

Figure 9. Real life noisy datasets – accuracy for datasets with NB (100, 2000, 10000 features) and SVMs

The most interesting domain we handled was the contact center customer feedback domain CCFb. Feedback to contact centers tends to be short, crisp, and often contains abusive remarks from customers. Many a times the verbatims are very short in length and ambiguous in nature. Also in this domain categories, often known as *call drivers*, may make business sense but seldom have enough data to train models. A harder problem is that the classes defined are often confusing, overlapping, and there is no consistent procedure for labeling comments. This leads to a separability problem to train an automatic classification system driving accuracies down as a whole. For three different datasets, we got accuracies of 58.3%, 47.9%, 47.6%, and 59.1%, 53%, 47.8% for NB and SVM respectively.

The root problem in this domain is not feature noise, which we have been discussing throughout, as much as label noise. One of the accounts in the contact center we dealt with, we asked 200 cases to be multi-labeled by two quality analysis domain experts. Multi-labeled classification allows a text document to be associated with more than one label at a time. A week later the same exercise was repeated. A statistical ANOVA Gauge Reproducibility and Repeatability test showed that multi-labeling results were *not reproducible* 53% of the time across humans and the same expert could *not repeat* his own multi-labeling 35% of the time. While multi-labeling has clearly contributed to these very low consistency rates , it points to a larger problem of bad label-set design and the lack of a consistent labeling process. Such an observation is known to some extent to text classification practitioners and about 30% disagreement amongst expert human labelers is accepted [14]. In designing real-life systems, label noise emerges as a very important kind of noise to consider. However, we will restrict further discussion on label noise in this paper.

We would like to note here that domain specific efforts to improve operational text classification systems have been successful in dealing with feature as well as label noise to some extent. This, however, requires significant care and cost not afforded in general. In this study we attempted to systematically study such noise across various domains.

4.3 Discussion

In this section we return to inspect the relationship between abundance of terms, sparsity of feature vectors in text classification, statistical (information gain based) feature selection and noise. We noted that corrupting the test set for benchmark datasets like R10 did not lead to large drops in accuracy. This remained true at moderate (40%) and high (70%) levels of noise. We investigated the top 10 most informative features ranked by information gain (IG) learned with noisy training data.

In Table 1 we show the top 10 features ranked by information gain with 0, 40, 70, 100% training noise. Note that there is very little difference between the first two sets of features – even 40% training noise finds abundant patterns in the rest of the training data. Even at 70% noise the important words can be still be seen to be occurring though some spelling mistakes (e.g., teh) have now assumed the status of signal-in-the-noise. At 100% noise, as expected, all words are mangled, and short words (with higher chance of similar corruption due to abundance) emerge as discriminative features. Also as mentioned in section **??**, words of length less than three characters are not tampered with, valid 2-char words appear frequently in the list of top features. Note the sharp drop in information gain absolute values as noise increases. These numbers are indicative and roughly comparable as they are over the same training corpus and document labeling – only feature noise has been introduced in the form of spelling errors. The drop in information is expected because a lot of information is lost as at 40% and 70% noise there is that much probability that each word in the corpus is corrupted. However the abundance of important words repeatedly throws up similar information gain rankings even at high degrees of noise.

Original data		40% noise	
IG	Feature	IG	Feature
0.37063	lt	0.22173	cts
0.27613	cts	0.16753	lt
0.19878	net	0.14588	net
0.16231	wheat	0.13415	wheat
0.14117	shr	0.11304	trade
0.13849	qtr	0.10931	tonnes
0.12909	trade	0.10072	oil
0.12275	revs	0.09164	shr
0.12116	tonnes	0.08861	revs
0.1163	agriculture	0.08379	bank
70% noise		100% noise	
IG	Feature	IG	Feature
0.13281	cts	0.10363	teh
0.09416	wheat	0.09123	cst
0.08846	trade	0.0901	te
0.08594	tonnes	0.08862	cs
0.0852	teh	0.07532	thhe
0.08326	lt	0.0622	nte
0.08104	te	0.05835	ctts
0.07753	net	0.05734	ol
0.07081	cs	0.05437	oli
0.06959	oil	0.05046	tge

Table 1. Information gain for most informative features of R10

We made very similar observations studying the 20NG dataset. There were consistent drops in information gain values with addition of noise. Coupled with our discussion on label noise in real life noisy text classification domains, our observations lead us to believe that feature noise is an important aspect to consider while building text classification systems, but large corpora often soften it's effects. Feature noise seems to have limited effect and can be effectively countered with known feature engineering and feature selection techniques coupled with the choice of a robust classification model. The most care needs to be spent in actually tackling label noise, designing a good separable set of classes, and setting up a consistent labeling process.

However there are multiple points to consider while designing such systems. An abundance of important features is important in learning robust text classification models[10]. If such an abundance can be confirmed then feature selection needs to be executed carefully – since the state-of-the-art accuracy achievable on the dataset at hand will be quickly estimated using simple NB models. Consistent with traditional wisdom, SVMs outperform NB, but require more training time and tuning.

5 Conclusion and Future Work

In this paper, we have studied various aspects of noise with detailed experiments to study it's effect on automatic text classification systems. The most interesting observation we made for benchmark datasets was that introducing as much as 40% *feature noise* in documents did not affect text classification accuracy much. Feature noise seems to have limited effect and can be effectively countered with feature engineering and selection techniques coupled with the choice of robust models. We experimented with many real-world CRM domains capturing a broad spectrum of noise (call summaries, customer emails, feedback forms). A striking observations here was the stark presence of *label noise* highlighting the need to properly design label-sets.
We would like to continue studies with a broader spectrum of real-life noisy datasets like time-constrained handwritten summaries. We believe these scenarios will be emergent with the growing customer focus of businesses and the ever-growing amount of information present in the real world. We would like to better explore label noise.

Acknowledgements:

We would like to thank Nitendra Rajput of IBM India Research Laboratory for helping with the ASR system.

References

[1] L. R. Bahl, S. Balakrishnan-Aiyer, J. Bellegarda, M. Franz, P. Gopalakrishnan, D. Nahamoo, M. Novak, M. Padmanabhan, M. Picheny, and S. Roukos. Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task. In *Proc. ICASSP '95*, pages 41–44, Detroit, MI, 1995.

[2] T. Bayer, U. Kressel, H. Mogg-Schneider, , and I. Renz. Categorizing paper documents. In *Computer Vision and Image Understanding*, volume 70, pages 299–306, 1998.

[3] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. In *Proc. of NIPS 14*, 2002.

[4] M. Choudhury, R. Saraf, V. Jain, S. Sarkar, and A. Basu. Investigation and modeling of the structure of texting language. In *Proc. of AND07 Workshop in conjunction with IJCAI*, 2007.

[5] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176, 1964.

[6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, 2000.

[7] Y. Freund and R. Schapire. A short introduction to boosting. In *Japanese Society for AI 14(5), 771-780. 11*, 1999.

[8] P. Haffner, G. Tur, and J. Wright. Optimizing svms for complex call classification. In *Proc. of ICASSP*, 2003.

[9] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of UAI*, 1999.

[10] T. Joachims. Making large-scale support vector machine learning practical. In A. S. B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[11] K. Kukich. Technique for automatically correcting words in text. *ACM Comput. Surv.*, 24(4):377–439, 1992.

[12] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8, 1966.

[13] D. D. Lewis. Evaluating text categorization. In *HLT '91: Proceedings of the workshop on Speech and Natural Language*, pages 312–318, Morristown, NJ, USA, 1991.

[14] D. D. Lewis, R. Ghani, D. Mladenic, I. Moulinier, and M. Wasson. In *3rd Workshop on Operational Text Classification (OTC), in conjunction with SIGKDD*, 2003.

[15] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[16] M. Michelson and C. A. Knoblock. Semantic annotation of unstructured and ungrammatical text. In *Proc. of IJCAI*, pages 1091–1098, 2005.

[17] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named entity extraction from noisy input: speech and ocr. In *Proc. of the sixth conference on Applied natural language processing*, pages 316–324, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[18] T. Nartker, K. Taghva, R. Young, J. Borsack, and A. Condit. Ocr correction based on document level knowledge. In *Proc. of Intl. Symp. on Electronic Imaging Science and Technology*, pages 103–110, Santa Clara, CA, 2003.

[19] T. Nasukawa, D. Punjani, S. Roy, L. V. Subramaniam, and H. Takeuchi. Adding sentence boundaries to conversational speech transcriptions using noisily labeled examples. In *Proc. of AND07 Workshop in conjunction with IJCAI*, 2007.

[20] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proc. of IJCAI-99 Workshop on Machine Learning for Information Filtering*, 1999.

[21] L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. Prentice-Hall Inc., New Jersey, 1993.

[22] S. Roy and L. V. Subramaniam. Automatic generation of domain models for call-centers from noisy transcriptions. In *Proc. of ACL-COLING*, 2006.

[23] A. Sarma and D. Palmer. Context-based speech recognition error detection and correction. In *Proc. of HLT-NAACL*, 2004.

[24] A. Vinciarelli. Noisy text categorization. In *Proc. of ICPR'04 Volume 2*, pages 554–557, Washington, DC, USA, 2004. IEEE Computer Society.

[25] A. Vlachos. Active annotation. In *Proc. of the EACL 2006 Workshop on Adaptive Text Extraction and Mining*, 2006.

[26] J. Zhang and Y. Yang. Robustness of regularized linear classification methods in text categorization. In *Proc. of SIGIR*, 2003.

[27] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In *Proc. of ICML*, 2000.