# A Comparison of Classification Techniques for Technical Text Passages

Mark M. Kornfein, Helena Goldfarb

***Abstract*— Our work explores the use of several text categorization techniques for classification of manufacturing quality defect and service shop data sets into fixed categories. Although our work was in the area of manufacturing quality the technique is applicable to free form, short text summaries of data that may be stored in a database, file, or document. We refer to these types of text as "technical text passages". These summaries may not follow standard grammar conventions; they commonly contain abbreviations, technical phrases, misspelled words and industry specific acronyms. Typical types of text to be classified include aircraft engine repair shop findings, industrial manufacturing quality problems and corrective actions, and standardization of attributes in a bill-of-materials. In this paper, we will present our results in using machine learning and rule based algorithms to categorize text. Our results show that the rules based approach is as good as several machine learning approaches. For example, using Support Vector Machine algorithms we were able to achieve 82% accuracy on validation set, using 1,645 training samples and 823 validation samples. Each category had 50 or more samples. Using rule-based approach we were able to achieve 80% accuracy.**

***Index Terms*— Industrial quality text passages, Manufacturing quality defect categorization, Support Vector Machines, Text classification**

## I. Introduction

Businesses produce a large amount of descriptive documentation as a by-product of processes such as manufacturing and servicing equipment. Often some of this documentation is in the form of short technical text passages that are stored in databases, files or documents. Examples of this kind of text data include service reports such as findings during a repair, documentation of manufacturing quality problems and customer help desk notes to name just a few. Typically these text passages are created by technicians, or engineers, as a way of documenting their work. The text passages are usually concise, contain lots of industry specific terms, abbreviations, and are often not grammatically correct.

These technical text passages represent an important corporate resource, especially in the area of quality control. They may contain information that point to systemic design, manufacturing or deployment problems. As such they need to be reviewed on an ongoing basis to ascertain if there are systemic problems, quality issues, etc. In the past, personnel have been employed to specifically review and analyze this type of data, but it is costly to do so, and automation of this process represents a significant cost savings to the business.

Since the 1960s as noted by Sebastani [1] there have been automated approaches to categorizing text such as news list articles and later web pages, for example.

### A. *Problem Description*

The specific problem that is addressed in this paper involves categorizing manufacturing quality defects. The manufacture of large industrial items such as jet engines or power turbines involves many operations from initial design, to creation of a component parts, to final assembly and shipping of the product. When a quality issue occurs that requires the component to be reworked, re-designed, or reassembled, GE employees record defect notes. For each defect note, in addition to enumerated data a problem description and the corrective action is entered. The personnel who enter the defect notes are typically shop floor manufacturing personnel or service desk personnel who enter a quality issue as quickly as possible using abbreviations, acronyms and phrase fragments. Engineers then review these problem descriptions to determine the root causing process, look for systemic problems, and forward the issue to the engineer responsible for that process. It should be noted that the manufacturing personnel do not typically have the domain expertise or time to determine and enter a causing process for a defect. For the engineers, reading and reviewing defect notes is a time consuming, subjective and error prone process.

In this domain all text is categorized and items that do not fit a known category are placed in a miscellaneous category. The fits the reality that some problems are unique and do not fit a known categorization.

This paper reviews and compares several machine-learning approaches that were developed in addition to a rule-based system that was developed. The machine learning approaches included naïve Bayesian learner, decision trees, and Support Vector Machines (SVMs). The rule-based system is built in Visual Basic and supports a grammar for allowing users to automatically create rules. Rules for the rule engine were

Manuscript received March 5, 2007

Mark Kornfein is with the Sensor Informatics Technology Laboratory at GE Global Research, Niskayuna NY 12309 USA (phone: 518-387-5843; fax: 518-387-6104; e-mail: Kornfein@research.ge.com).

Helena Goldfarb is with the Enterprise Architectures Laboratory at GE Global Research, Niskayuna NY 12309 USA (e-mail: goldfarb@research.ge.com).

determined by multiple methods including domain experts, tool feedback and a natural language tool, XDoX developed by Hardy et al. [2] that determined some non-obvious rules.

.

## II. Methods used and their comparison

### A. Data Description

The data analyzed represented 12 years of manufacturing quality data for major quality issues at a GE industrial business. Each quality issue represented a 300,000 US dollars or higher cost to the business to resolve. The data in this quality system is stored in a relational database but is extracted into Excel spreadsheets for analysis. There were 2468 records total in the system.

The problem description and corrective action text in the database ranged in size from a small phrase to a couple of paragraphs; typically a few sentences were found. The average text size was about 75 words with a standard deviation of about 35 words.

The initial analysis of the text showed that there were a number of differences between technical text passages and document text one might find in a news article or email: many domain specific acronyms were in the text, abbreviations were often used, some misspellings were found, and phrases were often used instead of full sentences. Initially several pre-processing techniques including stemming, automated spelling correction and synonym evaluation via the use of WordNet developed by Miller et al. [3] were considered. The analysis indicated that in this context few synonyms were used and the terminology across the industry was fairly consistent. Further tests with stemmers such as the one developed by Porter [4] showed lack of improvement as stemming did not decrease the word count significantly, also a number of issues were found on stemming the technical jargon. Automated spelling correction was considered but not done due to time limitations and some initial findings that suggested it did not account for many errors. However, automated spell checking is an area where the technology may be expanded in the future.

In addition to text quality issues outlined above, we faced two other problems that effected text classification. First, concept drift occurred over the course of the twelve years of data we were analyzing. Concept drift affected both the classification categories, which changed slowly over time, and input text, which changed due to changing manufacturing methods.

The second issue we faced was the quality of the training data. We had access to twelve years of data which had been analyzed manually by the engineering staff to determine causing process over the twelve years. The manufacturing quality organization had concerns about the consistency of this training set and just prior to the start of our work re-analyzed the entire set of data. This resulted in almost a 25% change in causing process. Additional reviews of the training data have led to further changes due to the subjective nature of some classifications, manual errors, and the engineer having expertise in only a subset of the quality issues found. Using Six-Sigma quality analysis this led to a process capability of 2.19s for the manual process.

### B. Rule Based Categorization

We describe the results of extensive experiments on large text passage collections using optimized rule-based induction methods. There are many techniques and algorithms for automatic text classification of different types of text but none of them address specific challenges related to classifying technical text passages. One of the closer examples of text categorization is the classification of military messages by Carr [5], these text messages whereas similar tend to be more structured and date-time specific than our technical text passages.

In addition there exist a number of automated rule learning algorithms. One example of such an algorithm was published by Sasaki and Kita [6]. These algorithms automatically extract rules from existing documents. We, on the other hand, focused on developing comprehensive and easy to use grammar to enable our users to efficiently record their domain knowledge.

The Rule Based Classifier consists of a set of rules that represent categories and logic that determines whether a document satisfies a rule. If a document satisfies a rule, it is classified into the category the rule represents. For example, “assembly && part” rule will match any document that has assembly and part words in it. Rule representation is relatively easy for people to understand and a convenient way to record human knowledge in cases where training data is not available.

The rule-based classifier efficiently categorizes large sets of text summaries in a reasonable time frame and with acceptable accuracy. It provides human readable classification rules that can be easily defined and fine-tuned and understood. It needs no training, providing good asset for our problem type because in most cases there is no training set.

We implemented a rule-based classifier embedded as an Excel add-in for ease of use. The classifier supports a rule set that includes AND, OR, NOT operators, phrase matching, substring matching and whole word matching. In a second version of the tool simple support for synonyms was added. WordNet was reviewed first to fill the synonym need but it was felt due to the nature of the industrial vocabulary it would not be appropriate. Instead a simple mechanism was built allowing users to list synonyms on a spreadsheet, which the rule classifier used. In some cases the organization had already built a technology specific synonym table for other uses that we were able to use. Using synonyms reduced number of rules and simplified their complexity.

On most data sets domain experts and a few iterations running the categorizer quickly led to a set of rules that handled most text passages. In a couple of cases domain experts had a hard time determining a set of rules which gave a high degree of accuracy. In those cases the documents were categorized by hand and then troublesome categories were run though a natural language tool, XDoX developed by Hardy et al. [2] that output rules that were not apparent to the domain experts. These machine-generated rules were then added to the rule-based categorizer

The tool can be used on any dataset that contains textural data to be categorized. The data can be entered in multiple Excel worksheets and multiple rows inside a worksheet. The tool also provides automatic result validation, generation of confusion matrix, and visual feedback of mismatched results when test data is available. The accuracy of the method was at least 80% on validation sets.

### *C. Support Vector Machines Categorization*

Support Vector Machines (SVMs) is another method that we used to categorize technical quality text passages. SVMs are a relatively new machine learning method introduced by Cores et al. [7]. SVMs typically show better performance then some other machine learning techniques like Naïve Bayesian Methods or Artificial Neural Nets in text categorization as described by Kwonk [8]. SVMs algorithms can work efficiently with very large feature sets because it finds the margin of separation of the data rather than matches on features. SVMs algorithm is linear, efficient and scalable to large data sets. Basu, Watters, and Shepherd [9] also showed that SVMs are robust in the presence of noise.

SVM classification algorithm solves two-class problems. It is based on finding a maximum separation between hyperplanes defined by classes of data, shown in Fig. 1.

Non-linearly separated data are mapped into much higher dimensional feature space where they can be linearly separated.

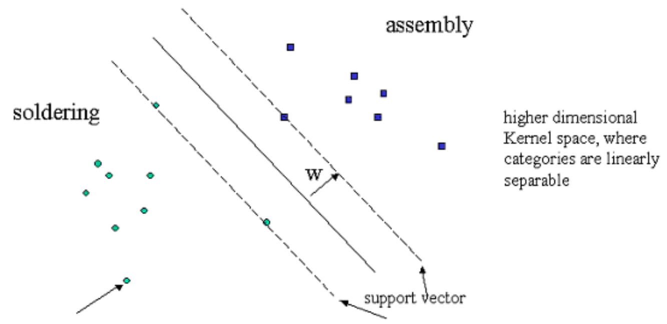
assembly
soldering
higher dimensional Kernel space, where categories are linearly separable
w
support vector

Fig. 1 Example of SVM's hyperplane pattern.

The first step in preparing data was to convert each document into a feature vector. The length of the feature vector is the dimension of a dictionary and each of its entries specifies whether the dictionary term occurred in the document or not.

As described above, our data consisted of technical text passages that may not follow standard grammar conventions; they commonly contain abbreviations, technical phrases, misspelled words and industry specific acronyms. As a result, it was not feasible to use traditional full dictionaries. We created our own dictionary by extracting all unique words from the existing text passages. Even though we had only 2,468 text passages, our dictionary contained over 11,000 terms that included part numbers as well.

In order to reduce the size of the dictionary, we removed terms that had special characters and numbers in them such as part numbers and names. We also removed words that occurred less then three times. This step removed most of misspelled words and non-standard abbreviations. We also manually generated a list of generic stop words, that is words that occur frequently in a text and do not add any valuable information, and removed them from the dictionary as well. The resulting dictionary contained about 5,000 words.

Next, we mapped each text passage to a vector of individual term values for that text. We used the following representation in (1) of each document as a feature vector.

Let

$$F = (F1, F2, \dots, Fi, \dots, Fn), \ . \qquad (1)$$

where n is the number of words in the dictionary and Fi is the ith term of the feature vector defined by (2)

$$Fi = Ni * Log\,(Nd / Ndi), \qquad (2)$$

where Ni is the number of times ith dictionary term appeared in this document, Nd is the total number of documents in the training set, and Ndi is the number of documents in the training set where the ith dictionary term appeared.

The feature vector of each passage is Fnorm (3), which is the norm of the vector F,

$$Fnorm = norm\,(F). \qquad (3)$$

The feature representation works well even with stop words, for example words such as “a”, “the”, “when”, “then”. Stop words result in a ratio Nd/Ndi that is close to 1, which results in their corresponding logarithmic term being close to zero. As a result, a stop word term does not affect much the feature vector representation. This explains why removing more stop terms did not result in accuracy improvement. Removing stop words just reduced the dimension of the feature vector but did not improve the accuracy of the classifier.

On the other hand, removing words that occurred infrequently improved performance of the classifier on the validation set. Our experiments gave the best results when we removed words that occurred less then 3 times. This took care of misspelled words and part names that are too specific to individual text passages. However, removing words that occurred more then three times decreased accuracy. This can be related to the fact that we were removing some essential terms. Vector representation, in some way, also takes care of infrequently occurring words by vector normalization, which prevents one large term to greatly influence the whole vector.

We used SVMlight implementation of Support Vector Machines to create classifiers and classify the data. SVMlight was developed by Joachims [10]. SVMlight solves two-class problem. Using 1,645 training samples we trained one classifier for each category. Each category's classifier determines whether a sample belongs to the category. To classify our data we ran each sample though every category classifier and chose the category that resulted in the largest boundary separation. Our first results had 77% accuracy on the training set and 71% accuracy on validation set, which was 1/3 of our samples. Minimum sample size was 50 samples.

After examining our experimental results we noticed that SVM algorithm was very conservative. It never assigned a wrong category to a sample, although it would leave a sample not classified and therefore in our miscellaneous classification. We moved the decision boundary to include more samples in a category and were able to achieve 96% accuracy on training set and 82% accuracy on the validation set, with minimum sample size of 47 samples. The degree of how conservative one wants the decisions be depends much on the application. For example, an application that decides automatically whether to approve insurance payments should be more on the conservative side. It is better to request manual review of an application then to authorize insurance payments to an unqualified person.

### D. *Other machine learning techniques*

In addition to the rule-based engine and support vector machines a number of other machine learning techniques including Naïve Bayes, decision trees and maximum extents were experimented with using the Mallet classification engine developed by McCullum[11]. These techniques in general were not as good as the rule-based or SVM approaches and were not pursued. Table 1 shows the initial accuracy from the confusion matrix created for each method.

TABLE 1 - MACHINE LEARNING CLASSIFICATION

| Method | Test set accuracy |
|---|---|
| Naïve Bayes | 69% |
| Decision trees | 52% |
| Max Extents | 79% |

## III. CONCLUSION

Both Rule Based and SVM methods worked well in classifying technical text passages achieving an acceptable, accuracy of at least 80%. The simpler rule-based method was surprisingly effective while having the advantage of being most user-friendly method for entering and utilizing data. The use of Support Vector Machines methods can achieve even greater effectiveness when customized to particular dictionaries and specifics of context. The rule-based approach is easy to modify, does not require training, and is a white box approach, increasing users' level of acceptance. The rule-based method did not require a large set of training data, which turned out to be crucial for customer acceptance. In addition, because of the unique characteristics of the text, traditional web mining or natural language techniques, like semantic parsing, WordNet or spell checking, did not appear for the most part applicable. We have successfully applied our methods to classify over 100,000 cases on several different data sets.

Our approach can be applied in classification of any industry specific free form text like shop findings or medical summaries for insurance applications.

## ACKNOWLEDGMENT

We would like to thank Mike Cimini of GE Aviation who directed the use of this tool and supplied invaluable data sets to analyze.

## REFERENCES

[1] F. Sebastiani, "Machine learning in automated text categorization." *ACM Computing Surveys*, 34(1) , pp. 1-47, 2002.
[2] H. Hardy, N. Shimizu, T. Strzalkowski, L. Ting, B. Wise, and X. Zhang, "Cross-document summarization by concept classification." *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, Tampere, Finland, pp. 121–128, 2002.
[3] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Five papers on WordNet." Technical Report 43, Cognitive Science Laboratory, Princeton University, Princeton, New Jersey, 1990
[4] M. E. Porter, "An algorithm for suffix stripping." *Program*, 14(3), pp. 130-137, 1980
[5] O. Carr, and D. Estival, "Document classification in structured military messages." *Australasian Language Technology Workshop*. Melbourne, Australia, 2003
[6] M. Sasaki, and K. Kita, "Rule-based text categorization using hierarchical categories." *Proc. of the IEEE Int. Conf. On Systems, Man, and Cybernetics*, LaJolla, CA, USA pp 2827-2830, 1998
[7] C. Cores, and V. Vapnik, "Support-vector networks." *Machine Learning*, 20(3) pp.273-297, 1995
[8] J. Kwonk, "Automated text categorization using Support Vector Machine." *Proceedings of the International Conference on Neural Information Processing*, Kitakyushu, Japan. pp 347-351, 1998
[9] A. Basu, C. Watters, and M. Shepherd, "Support Vector Machines for text categorization." *36th Annual Hawaii International Conference on System Sciences*. Big Island, Hawaii, USA pp103-109. 2003.
[10] T. Joachims, "SVM light online" [online]. Available: http://www.cs.cornell.edu/People/tj/svm_light
[11] A. McCallum, "MALLET: A machine learning for language toolkit. " [online]. Available: http://mallet.cs.umass.edu 2002