

# FoodSIS: A Text Mining System to Improve the State of Food Safety in Singapore

Kiran Kate  
IBM Research  
kirankate@sg.ibm.com

Andy Prapanca  
IBM Research  
andypp@sg.ibm.com

Sneha Chaudhari  
Language Technologies Institute,  
Carnegie Mellon University, Pittsburgh, USA  
sschaudh@andrew.cmu.edu

Jayant Kalagnanam  
IBM Research  
jayant@us.ibm.com

## ABSTRACT

Food safety is an important health issue in Singapore as the number of food poisoning cases have increased significantly over the past few decades. The National Environment Agency of Singapore (NEA) is the primary government agency responsible for monitoring and mitigating the food safety risks. In an effort to pro-actively monitor emerging food safety issues and to stay abreast with developments related to food safety in the world, NEA tracks the World Wide Web as a source of news feeds to identify food safety related articles. However, such information gathering is a difficult and time consuming process due to information overload. In this paper, we present FoodSIS, a system for end-to-end web information gathering for food safety. FoodSIS improves efficiency of such focused information gathering process with the use of machine learning techniques to identify and rank relevant content. We discuss the challenges in building such a system and describe how thoughtful system design and recent advances in machine learning provide a framework that synthesizes interactive learning with classification to provide a system that is used in daily operations. We conduct experiments and demonstrate that our classification approach results in improving the efficiency by average 35% compared to a conventional approach and the ranking approach leads to average 16% improvement in elevating the ranks of relevant articles.

## Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining–Machine Learning;  
H.3.3 [Information Search and Retrieval]: Selection Process

## Keywords

Machine Learning, Text Classification, Ranking, Semi-supervised Learning, Information Gathering

## 1. INTRODUCTION

The US Centers for Disease Control and Prevention (CDC) estimated that roughly 1 in 6 Americans or 48 million people fall ill,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*KDD'14*, August 24–27, 2014, New York, NY, USA.  
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...\$15.00.  
<http://dx.doi.org/10.1145/2623330.2623369>.

128,000 are hospitalized and 3,000 die of food-borne diseases each year [6]. WHO estimated that the diseases caused by major food-borne pathogens alone cost up to US \$35 billion annually in medical expenses and lost productivity [4]. In Singapore, the number of food poisoning cases and the number of people affected have increased significantly (by 14 and 17 times, respectively)<sup>1</sup> from 1965 to 2011 [14]. This increasing trend is an important public health concern as more people tend to consume meals outside of their homes. According to the Health Promotion Board (HPB), Singapore residents usually eat out at least four times a week at hawker centers, food courts and coffee shop stalls. The proportion has increased from 49% in 2004 to 60% in 2010 [5]. Food recalls also appear to be on the rise especially in the last decade [1]. Anecdotal evidence suggests similar increases food recalls in Singapore due to concerns regarding contamination in neighboring food growing and processing countries [2] or mislabeling [3]. Poor hygiene practices [14], food recalls, food contamination and mislabeling, poor food processing and related technologies are all factors that could cause food safety threats.

Government agencies such as the National Environment Agency (NEA) in Singapore are responsible for monitoring the state of food safety in the country. NEA is extremely pro-active in terms of monitoring emerging food safety issues in order to be able to take preventive actions wherever possible. One approach used by NEA to accomplish this is to constantly identify emerging food safety issues and developments by tracking the World Wide Web for articles related to food safety from a variety of local and international sources. NEA has a team to browse information including news on a daily basis from a set of identified web data sources. They sift through web articles on these sources and identify those related to food safety such that they are relevant to the function and scope of the Environmental Health Department at NEA. The identified articles are then summarized and compiled into weekly newsletters and circulated to the relevant sections in the organization. This paper describes FoodSIS, a system built for NEA to provide support for the generation of such newsletters.

The main purpose of the newsletters is to stay abreast with information such as food poisoning outbreaks happening worldwide, news about food recalls from neighboring countries, developments in food technology, and new policies/regulations about food. Each newsletter is an aggregated report of relevant information that is used by the operational teams of the agency to make tactical decisions regarding the published food safety alerts that are of rele-

<sup>1</sup>This is not a reflection of hygiene standards. Even as safety and hygiene standards go up, the sheer number of people eating out leads to higher incidents.

vanance. There are two main steps in the newsletter generation process: (i) searching and identifying the content about food safety issues such that it is interesting and relevant for the agency, and (ii) summarizing the identified content and compiling it in the form of the newsletter. Due to overload of information on the web and inability of search engines to capture subtleties of an information need, the manual execution of step 1 very difficult and time consuming. After a set of relevant articles have been identified, it is then relatively straight forward to manually summarize them since they are small in number. As a result, we focus our attention to step 1 in the current system development.

As we explain in later sections, we formulate the identification of relevant content as a document classification problem and relevance ranking based on labeled data. The design of such document classification systems traditionally uses supervised learning with positive and negative labeled documents. However in real life it is rare to find sufficient number of labeled documents. As a result, the design of an operational system to classify and rank documents accurately for daily use needs to exploit two additional sources of information: (i) the sources of data and the semantic and organizational structure of these sources, and (ii) the interactive aspects of how users interact with the system to consume the results of the classification and provide feedback on the ranked list of documents classified as relevant (in order of relevance).

The news sources that need to be examined for relevant documents provide an organization by separating articles by sections (such as politics, entertainment, sports, health etc). This structure provides a way of identifying and ignoring the sections that are irrelevant (such as sports, entertainment, ...) for the topics of interest (food safety). In addition, the marginal distribution of articles from different news sources provides useful prior information for the design of an operational system.

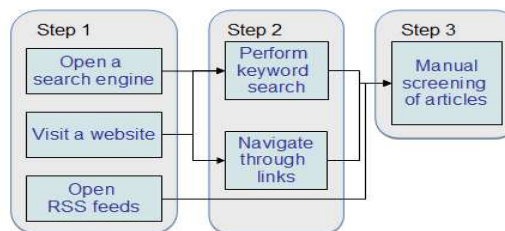
The other important requirement is the aspect of interactive learning. FoodSIS classifies articles into *relevant* and *not relevant* and presents the *relevant* articles as a rank ordered list (in order of relevance). Based on this interface, the user provides inputs by choosing documents that are truly relevant (as per her judgment). This interaction provides information beyond the simple labeling of documents as *relevant/not relevant*. The user implicitly provides pairwise comparisons (not necessarily a total order) between documents and separation among sets of documents by his selection of relevant documents from a set of documents presented to him. This can be modeled and used to adaptively improve the ranking.

The **Food Safety Information System** (FoodSIS) is a pilot system at NEA (National Environment Agency, Singapore) that is in operational test mode for daily operations from last 6 months. FoodSIS is based on a design of a classification and ranking engine using a variety of machine learning techniques, to account for the operational needs and the exploitation of the application specific information outlined in the paragraphs above. This system is now deployed on a pilot infrastructure and is in daily use for the past six months in an effort to evaluate its efficacy. The end users now routinely use FoodSIS as the first point of entry for navigating the web for food safety related news articles and report an increasing dependence on its availability for daily operations. FoodSIS is designed to provide a ranked list of the most relevant articles (for any food safety related search). The users routinely provide inputs both by marking the articles that are deemed as "truly relevant" and also qualitative feedback on the ease of use. This interaction provides data for continual improvement of the operational system.

The rest of the paper is organized as follows: Section 2 provides details about the data sources, data sets and the problem, and provides an analysis of the document classification problem. Sec-

tion 3 describes the design and architecture of FoodSIS. Section 4 provides details regarding the technical approach for building the operationally useful classifiers using machine learning techniques. Experimental evaluation of the classifiers and results are also presented in section 4. Section 5 discusses the use of pairwise ranking inputs provided (implicitly) by the users and how this is used to improve the ranking of articles classified as *relevant*. Section 6 is a discussion of some of the challenges and lessons that were learnt from the deployment of the system. We conclude the paper with a discussion of other application areas where this approach can be used to build information gathering systems in section 7.

## 2. PROBLEM AND DATA DETAILS



**Figure 1: A manual information gathering process**

Search engines are the most widely used tools for information search but there are two fundamental challenges in using them for routine, domain focused, target driven information needs: (i) Digital information overload, (ii) Inability to transform information requirements into precise keyword queries. Figure 1 captures a manual information gathering process that is commonly used today in organizations. As shown in the figure, they use three different information browsing mechanisms depending on the data sources. For example, some websites provide RSS feeds making it easier to access new content. Some websites are meant to host content on the topic of interest and users visit those directly. All other requirements are satisfied by search engines. A final step is to go through all the articles presented as search results, RSS feeds or different sections of a website and shortlist those relevant to the department.

The environmental health department of NEA also followed a similar approach to information gathering. The team identified 37 websites on which they find useful posts about food safety. These 37 data sources are categorized into 3 categories as follows:

1. **Official government websites:** Websites such as World Health Organization (WHO: <http://www.who.int/en/>) and websites of other government agencies.
2. **Media websites:** General global news websites.
3. **Food hygiene and safety websites:** Websites dedicated to information on food safety.

Food safety related information which is of interest to NEA can roughly be described by the following 5 categories: (i) food poisoning outbreaks/food borne diseases, (ii) developments in food technology and practices, (iii) food regulatory/policy developments, (iv) enforcement of food safety violation and recognition for food safety compliance, and (v) investigation reports on food hygiene inspections. The 5 categories only provide an idea of the kind of information that is relevant and are not used for any strict classification.

The team has a list of 19 keyword phrases they use for searching information. The keyword phrases include high level topics such

as “food safety”, “food hygiene”, “food policy”, “food poisoning”. Manually crafting a detailed and complete list of keywords for a domain is a time consuming task and it is tougher for a vast domain like food safety. Also, as we try to demonstrate in the example below, the domain understanding required for correct identification of relevant articles for NEA is beyond a list of keywords.

“People have reported vomiting, diarrhoea and other symptoms of food poisoning after eating products including pizza and lasagne made by a subsidiary of Maruha Nichiro Holdings, the nation’s largest seafood firm. ... Police began investigating the company last month after it revealed some of its frozen food had been tainted with malathion, an agricultural chemical often used to kill aphids in corn and rice fields.”

This is a story of a major food poisoning case in Japan where over 2800 people were affected due to contamination of frozen foods with pesticide. Information about food poisoning outbreaks is one of the important categories of information relevant to NEA. However, the article with the above excerpt and all the other related articles are labeled as not relevant by the users. To understand why, one must understand the cause of contamination reported in the article as well as NEA’s scope of work. The article suggests that the contamination might have happened due to pesticides during manufacturing. NEA does not oversee manufacturing stage of food products. They mainly deal with retail end of the food supply chain. So while they are interested in food poisoning outbreaks, the focus is more on food poisoning incidents which occur in food retail establishments. Extending the keyword search approach, we might need to craft complicated rules to capture these subtleties. For example, a rule such as “does not contain ‘agricultur’ (this word form can be obtained by stemming)” could be in the list of rules, and it should supersede the rule “contains ‘food poisoning’”. This needs a lot of manual effort and such an approach does not have the capability to learn over time. Machine learning techniques, in particular document classification seemed the most appropriate yet simple choice to automate the content identification step when some training data (labeled articles) is available. We also use a supervised rank learner to improve the ranked list of articles that is presented to the user as output. The next section describes the deployed system and its components.

### 3. SYSTEM OVERVIEW

Information gathering with FoodSIS is as simple as opening the web user interface, browsing through a ranked list of articles to identify the relevant content. The list has articles compiled from all the important data sources and ranking captures their relevance to the users. The system not only simplifies step 1 and eliminates step 2 of the manual process (figure 1), but also significantly improves the efficiency of article screening.

The overall system architecture is as shown in figure 2. The system is designed such that it is easy to use, flexible, and easily configurable. The main components of the system are: web crawler, article text extractor and indexer, document classification and ranking, and an interactive web user interface. We cover details of classification and ranking in section 4 and 5 respectively. We discuss the web interface briefly in the next paragraph. Due to space constraints, we omit the description of other components.

The web based user interface is an important component of FoodSIS which allows users to browse through the classified and ranked content. Figure 3 shows the web interface. Functionality provided by the web user interface of FoodSIS is designed with inputs from the users. Each of the features has proved very important for using FoodSIS effectively. Following are two important features available through the web interface: (i) *Ranked list of articles*: The right

pane of on the web interface is a ranked list of articles. After articles are classified as *relevant*, we use a supervised ranking method to rank them in the order of relevance by making use of user feedback. Each article in the list is associated with a check box. Users can label the article as relevant or not using the check box. This feedback is saved in the system and looped back for training every week. The models are updated at the same frequency. This simple mechanism proved practically very useful, since it served as a way of obtaining more labeled data with no/little extra efforts by the domain experts, (ii) *Keyword search*: Keyword search essentially makes the system act like a custom search engine allowing users to search using keywords. For example, if they want to see all articles containing “E. coli”, they can use simple keyword search. It can also be used to identify articles during bootstrapping phase of the machine learning algorithms.

FoodSIS is thus a tool for end-to-end information gathering and includes components for data collection, pre-processing, classification, ranking, and a user interface for interacting with the users effectively.

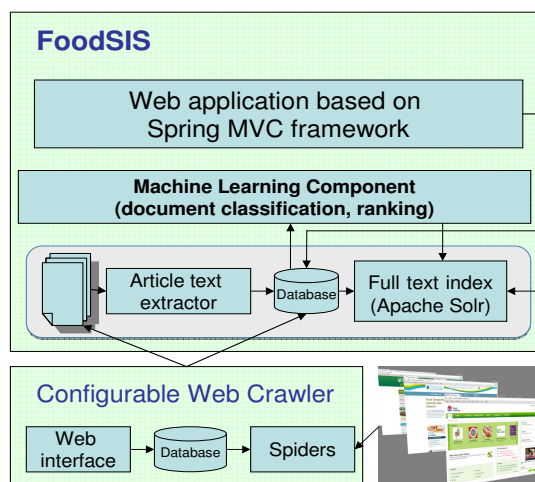


Figure 2: FoodSIS System Architecture

### 4. DOCUMENT CLASSIFICATION

The main challenge in making FoodSIS successful is the ability to present relevant content to the users. As illustrated in section 2, simple keyword search fails to capture the relevance effectively. Machine learning methods, in particular, document classification is well suited for such a task since the task is to separate relevant information from non-relevant. They have the ability to learn complex relationships from data and their performance improves with training. We formulate problem of identifying information relevant to food safety as a binary classification task. The class labels are *relevant* and *not relevant*. *Relevant* is a class of food safety related articles that are relevant to NEA and *not relevant* contains all the other articles that are not relevant. A quick look at the problem and the 5 types of information listed in section 2 might lead us to treating it as a multi-class classification. But we observed the domain and data carefully and noticed that the class boundaries for the 5 classes are not very clear. The 5 types are to provide guidance on the kind of information that is relevant to NEA and the information is not always strictly limited to those. Also, from system usability point of view, the agency does not necessarily need the articles to be categorized into those 5 categories.

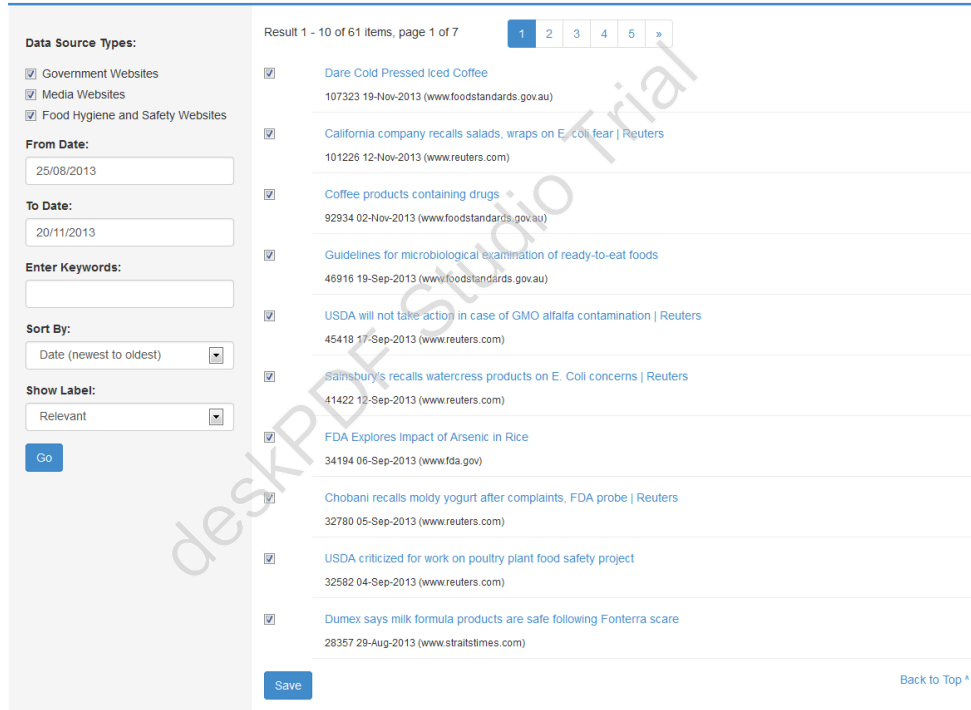


Figure 3: Screenshot of FoodSIS Web UI

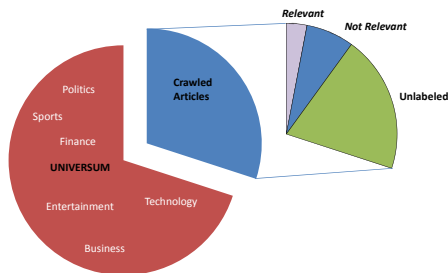


Figure 4: View of different components of the dataset

The most important consideration in a classification problem is the availability of labeled data for training. Figure 4 shows the datasets we have access to due to the nature of our problem. Each of the datasets can be interpreted and obtained as follows:

1. **Relevant:** We received around 6 months of food safety newsletters from NEA in e-mail format. Each newsletter has a list of *relevant* manually summarized articles. We extracted each article summary as a document of the *relevant* class. This seed data contained 53 articles of positive (i.e. *relevant*) class. We developed an initial classifier with this seed data as described in section 4.1. The users used this initial system to provide labels for more articles as discussed in 3. The labeling provided by users allows us to obtain *relevant* as well as *not relevant* articles. However, the amount of labeled data

of class *relevant*, obtained after a couple of months of training is still small, specially since the real class distribution is highly skewed<sup>2</sup>.

2. **Not relevant:** Our seed training data was based on the food safety newsletters and did not have samples of *not relevant* class. We obtained such samples for seed training by randomly sampling 200 articles from unlabeled articles. Though this is a standard practice, is clearly not an ideal way. Labeling by domain experts is the most ideal but expensive option. Our design of the FoodSIS web user interface allows us to obtain such labels with little extra efforts from the users. The size of our *not relevant* dataset continuously increases as users label them using the web interface.
3. **Unlabeled:** We crawl a high number of articles everyday but present only a few to the users based on their relevance. This essentially means we get user feedback only on a subset and a large portion of the crawled articles remains unlabeled. This allows us to use such unlabeled dataset for improving classification performance by employing semi-supervised techniques.
4. **UNIVERSUM:** UNIVERSUM is a set of data points that are known not to belong to any class of interest. For our setting, it is the set of articles from topics such as politics, sports, business etc. which can easily be kept out of the classification task. Websites such as news websites typically follow a categorization of news into multiple topics. This categorization is also reflected in the website structure. We can exploit

<sup>2</sup>just based on 4 out of 37 data sources the ratio of the number of *not relevant* articles to the number of *relevant* articles ranges from (30 to 45):1 per week

such structure and selectively choose only potential *relevant* articles for classification. For example, from news websites, data can be collected only from the “Health” sections, which then needs to be classified as *relevant* or *not relevant*. With this approach to data collection, the classifier does not have to work on articles from topics such as politics, sports, entertainment etc. and they would fall into UNIVERSUM. We followed websites’ structures to collect such UNIVERSUM data set.

Following subsections explain the key settings and methods we tried, reasons behind our choices and experimental evaluation. For all the methods, we used tf-idf scores of unigrams as our features.

## 4.1 Supervised Learning

A simple supervised binary classification was our first attempt at solving the task. We applied a simple Naive Bayes approach to classification since it has shown to work very well for document classification and is simple to implement. The first set of training data used for building a Naive Bayes classifier contained 53 positive samples extracted from the newsletters. Negative samples were obtained by randomly sampling 200 articles from unlabeled dataset. With most of the other components of the system in place and the Naive Bayes classifier, users started to browse the articles on the web interface. The article list was ranked by the probability scores output by the classifier for membership of class *relevant* (no hard classification was performed). We had version 1 running for roughly a couple of months. This allowed us to gather training data which is used for evaluating different paradigms and methods for classification as explained below.

## 4.2 Semi-supervised Learning

Supervised learning works quite well assuming that there is enough training data. In real life applications, this seldom is the case. Even with our interactive setup of obtaining training data from the domain experts, the training data we collected over a couple of months is quite small to obtain good classification performance. Advances in machine learning techniques allow us to employ other learning paradigms to handle this challenge. For our setting, unlabeled articles are available in abundance. Consequently, we chose semi-supervised learning to exploit the availability of unlabeled data to improve classification performance.

Other techniques to handle scarcity of labeled training data are (i) active learning which tries to minimize the labeling effort by selecting the most informative examples for labeling [19], (ii) using domain knowledge to label features instead of examples [13], (iii) domain adaptation from a similar domain using transfer learning [7], and (iv) concept labeling which labels concepts in domain ontologies and uses concept-class associations for generating training data of reasonable quality [12]. Techniques (ii) and (iv) propose to reduce the labeling effort but require the domain experts to work on a labeling task not same as the original classification. Technique (iii) assumes the availability of training data in a similar domain. For our setting, it would be an additional task for the domain experts to understand and work on a different labeling task. The interactive learning that our system engages in with the users can be termed as a simple form of active learning.

A variety of semi-supervised techniques have been proposed and shown to work well [9], [18], [8] [20]. Transductive Support Vector Machines(TSVM)[20] is an extension of the widely used supervised Support Vector Machines (SVM). The formulation of TSVM introduces a separate term in the objective function for slack variables of unlabeled data and hence aims to find a hyperplane that separates both labeled and unlabeled data with maximum margin,

maintaining the class ratio in unlabeled data. [17] discusses suitability of SVMs for information retrieval which shares many characteristics with our problem. [15] discusses the effectiveness of TSVMs for text classification where the authors mention that TSVMs are well suited for problem settings where the document vectors are sparse and input space has high dimensionality, outperforming conventional methods substantially while also being more robust. Therefore, we believe TSVMs is the best approach to apply to our case. Experimental results in section 4.5 show that TSVM performs better than SVM, specially when the amount of training data is very small.

## 4.3 Unlabeled Instances Selection

Incorporating unlabeled data in classification might not always be useful, as noted in [24], [10], [11]. There are many papers discussing cases where unlabeled data hurts the performance and indicating that it should be used carefully. We experimented with randomly sampling 20% to 100% of unlabeled data from the available unlabeled dataset and using it for training the TSVM. The comparison of precision, recall and F1 score for these different runs of TSVM with different amounts of randomly sampled unlabeled data did not show a clear winner. We tried a heuristic for selecting unlabeled instances based on the following property of our data sources: some data sources in our list, specially those under the category “food hygiene and safety websites” have a higher ratio of number of *relevant* articles to the number of *not relevant* articles. If we sample unlabeled data points from such sources of data, they are likely to create a dense region around *relevant* class. Following the TSVM principle, we expect such dense region to guide the separating hyperplane away from labeled data of *relevant* class, resulting in more test data points to be classified as *relevant*. Such a placement of the separating hyperplane will improve recall, reducing precision. During the first year of deployment of FoodSIS, high recall is important since it gives confidence to the users on ability of the system to provide complete results. The exact steps followed in the heuristic based selection of unlabeled data are: (i) identify a set of data source names (i.e. the hostname of the websites) corresponding to labeled examples of class *relevant*, lets call this set  $W$ , (ii) select articles from unlabeled data based on  $W$ , i.e. select only those unlabeled instances which appear on the hosts in  $W$ . We then used this unlabeled data sample set for training a TSVM. We call this approach TSVM-Meta since it uses the meta attribute “data source name” in selecting unlabeled data. Section 4.5 compares the performance of TSVM-Meta with other methods. As demonstrated in section 4.5, though there is no formal theoretical support, TSVM-Meta performs very well empirically, due to the intuition discussed above.

## 4.4 Exploiting UNIVERSUM

UNIVERSUM data has been found to help improve classification performance since it encodes some useful prior knowledge and there have been many methods proposed in the recent past to exploit the availability of UNIVERSUM [22]. Yang et. al. [21] proposed a tri-class support vector machine (3C-SVM) which uses a combination of UNIVERSUM and unlabeled data together as unlabeled data for developing a maximum margin classifier, utilizing the implicit knowledge among all available data. As a by product, it also identifies the irrelevant data from the set of unlabeled examples (which, in our case is a mix of UNIVERSUM and relevant unlabeled data), which in turn helps to improve the classification performance. Further, in 3C-SVM the irrelevant data in fact plays the role of seeking a good subspace of the decision boundary based on two principles as follows: i) the confidence of prediction is re-

lated to the distance of the data point from the decision boundary i.e. if the point is farther, the data point is likely to correctly classified, however, if the data point is near the decision boundary, there is lesser probability or confidence that the data point is classified correctly. As a result, ideally, the relevant data points should lie farther from the boundary and the irrelevant class should lie near to the boundary. ii) the maximum entropy principle indicates that a classifier should rely more on the relevant data, while maximally ignore the irrelevant data. Taking these principles into consideration, 3C-SVM aims to find a maximum margin hyperplane such that irrelevant data points are closer to the hyperplane than relevant examples. As a consequence, it combines the advantages of irrelevant unlabeled data and UNIVERSUM to lead to a maximum margin hyperplane fulfilling the two assumptions.

The identification of irrelevant data points from the set of unlabeled examples is important since 3C-SVM aims to find a hyperplane in which the irrelevant data points are close to the boundary. To accomplish this, it uses two loss functions:

- Symmetric Hinge loss:  $H_1(|u|) = \max\{0, 1 - u\}$ , a loss function to measure risk in unlabeled data
- $\epsilon$ -sensitive loss:  $I_\epsilon(u) = \max\{0, |u| - \epsilon\}$ , a loss function adopted to measure risk in UNIVERSUM data.

Consequently, to separate the unlabeled data into relevant and irrelevant data, 3C-SVM calculates,

$$l_{min}(x) = \min\{H_1(|f_\theta(x_i)|), I_\epsilon(|f_\theta(x_i)|)\} \quad (1)$$

where,  $f_\theta(x)$  is the classifier model. Hence, for an unlabeled data point, according to the value of the  $l_{min}(x)$ , when the error measured by the  $\epsilon$ -sensitive loss is smaller than the error measured by Symmetric Hinge loss, it can be determined as irrelevant data. 3C-SVM learns a decision boundary in which irrelevant data points are closer to the boundary whereas the relevant examples are farther away from the boundary.

## 4.5 Experimental Evaluation

In this section, we report comparison of different classification methods on a dataset obtained by methods described earlier. The dataset we use for all the experiments has 186 *relevant*, 406 *not relevant*, 1796 unlabeled and 900 UNIVERSUM articles. Our system in its current version crawls data from a subset of the 37 different data sources. The number of articles crawled ranges from around 600 to 1300 per day. We have access to a large set of unlabeled and UNIVERSUM articles, but we intuitively feel that since labeled data is limited, using all the other available data for classification might add a lot of noise. So we randomly sampled unlabeled data to roughly 3 times the size of labeled data and UNIVERSUM data close to 1.5 times the size of labeled data. All the experiments are run 10 times with random train-test split of the labeled data and we report the average precision, recall and F1 score of the methods. Figures 5, 6 and 7 show comparison of different classifiers based on F1 score, precision and recall respectively. The values on the x-axis are the percentages of test data in the random train-test split of labeled data. Please note that we vary test data percentage from 50% to 90% mainly because one of the objectives of comparing different methods was to find out which method performs the best with very few training examples. The motivation behind such objective is to be able to apply the system to a domain other than food safety easily. Availability of labeled data would be a major challenge to apply our methodology to a completely different domain. Hence it is important to employ a method that can work well

with a small training data set. The methods we evaluate are 3C-SVM, SVM, TSVM, TSVM-Meta, and keyword search (referred to as KB in the plots). The keyword based approach uses the list of 19 keyword phrases provided by the NEA users to us and classifies an article as *relevant* if one or more of those keyword phrases occur in the article.

Figure 5 shows that 3C-SVM outperforms all other methods in F1 except the case when there are only 10% examples for training, i.e. around 18 examples of positive class and 40 examples of negative class. Improvement in F1 score due to 3C-SVM is about 10% over TSVM, in the range of 3% to 8% over TSVM-Meta, 1.5%-6% over SVM and about 27% to 39% over KB except the case with 10% labeled data for training. SVM performs as good as 3C-SVM when 50% of labeled data is used for training. But the F1 score drops drastically as the training set size becomes smaller. This is expected and is the main reason why semi-supervised learning is effective when training data is too small. Same effect can be observed in comparing SVM with TSVM and TSVM-Meta. Improvement in 3C-SVM over TSVM and TSVM-Meta can be attributed to its use of UNIVERSUM and the way it incorporates such data. An analysis of our data set using 3C-SVM provides the average distances of the 3 classes from the decision boundary (the separating hyperplane). The average absolute distance of the positive class documents from the separating hyperplane is 2.013 ( $\sigma=0.39$ ) while the average absolute distance for the negative class is 0.65 ( $\sigma=0.23$ ). From within the unlabeled data it is now interesting to note that the average absolute distance of the irrelevant data (UNIVERSUM) is only 0.13 ( $\sigma=0.22$ ) while the rest of the unlabeled data has a absolute distance of about 0.72 ( $\sigma=0.31$ ) and therefore can be considered as relevant. These numbers indicate that the underlying principle of irrelevant data being close to the decision boundary while relevant data being far from decision boundary is satisfied for our data sets and the analysis provides a good design for placing the decision boundary.

It is interesting to observe the significant lift in F1 score (4% to 20%) achieved by the unlabeled instances selection in TSVM-Meta over TSVM. As explained in section 4.3, this could be due to the proximity of the selected unlabeled instances to positive examples. It could also be due to reduction in the size of the unlabeled data resulting from the selection. To verify this, we ran experiments with random sampling of unlabeled data varying the percentage size of the sample and ran TSVM on each such dataset. We did not see any trend or conclusive effect of the size of the unlabeled dataset just based on random sampling. This leads us to believe that careful selection of unlabeled instances exploiting important properties of the dataset is effective and we are working towards formalizing such an approach. The F1 score for KB is more or less constant with different percentage of test data since there is no training phase in KB. It is important to note that the performance of most other methods matches or is lower than KB when only 10% labeled data is used for training. This case corresponds to approximately 18 *relevant* articles and 40 *not relevant* articles. It is difficult to apply machine learning techniques for such a small dataset and obtain reasonable performance.

As shown in figure 6 and 7, precision of 3C-SVM is lower than the other learning methods, but recall is very high and beats all other methods. SVM is the reverse case with highest precision and lowest recall for most data splits. TSVM and TSVM-Meta have a good balance of precision and recall compared to 3C-SVM and SVM. KB as expected has poor precision but relatively moderate recall compared to all other methods. For a system like FoodSIS, one of the challenges is to make the users comfortable with the machine learning based concepts and approach. It is difficult for

domain experts to understand that it is highly unlikely for a machine learning method to be 100% accurate. The users want the system to capture all the relevant information i.e. high recall with high precision. Due to the trade-off between precision and recall, when discussed with them, they showed greater preference towards higher recall since it is important for them to see completeness in the results. 3C-SVM would still improve the precision over KB by approximately 35% in most cases. Based on high F1 scores and high recall, we chose 3C-SVM as our classifier for version 2 of FoodSIS.

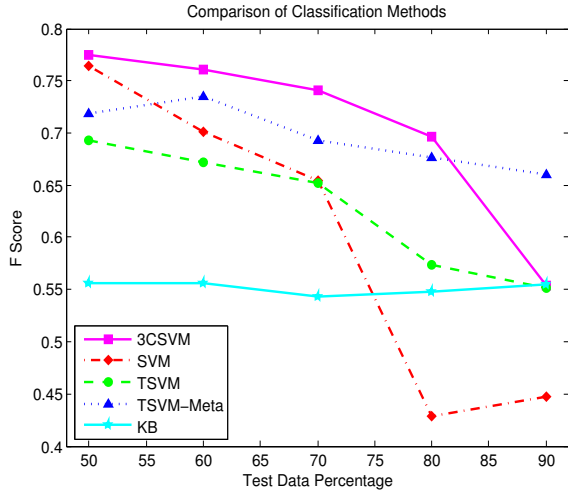


Figure 5: Comparison of different classifiers based on F1 score

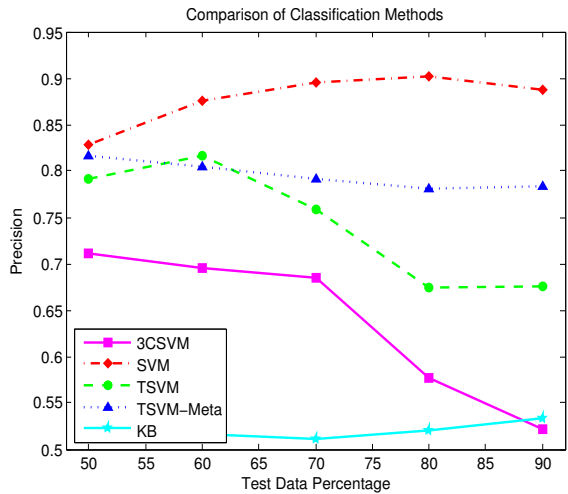


Figure 6: Comparison of different classifiers based on precision

## 5. LEARNING TO RANK

In this section we describe the interactive learning feature of our system. FoodSIS can actively incorporate the knowledge and preferences of users/domain experts into the system and use this feedback to improve the ranking of articles classified as *relevant* by the classifier. We saw how data for training the classifier is obtained from user feedback provided through the web interface. This feedback can also be used to infer partial pairwise comparisons between

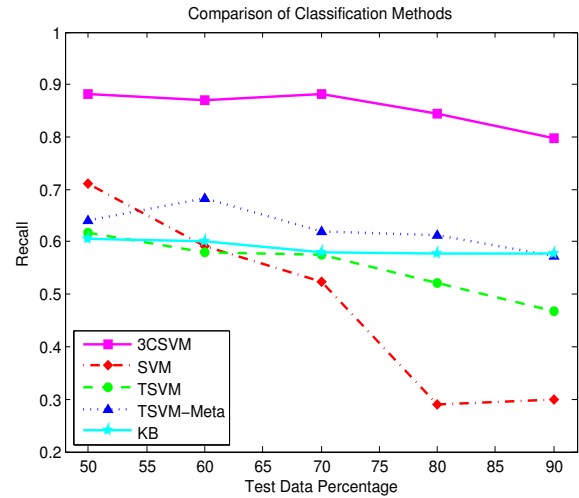


Figure 7: Comparison of different classifiers based on recall

the articles. For example, suppose a ranked list of 10 articles is presented to the user out of which, she labels the 4<sup>th</sup> and 7<sup>th</sup> article as relevant, then we can implicitly determine pairwise preferences between these articles such as :4<sup>th</sup> > 1<sup>st</sup>, 4<sup>th</sup> > 2<sup>nd</sup>, 4<sup>th</sup> > 3<sup>rd</sup>, 4<sup>th</sup> > 5<sup>th</sup>, 4<sup>th</sup> > 6<sup>th</sup>, 4<sup>th</sup> > 8<sup>th</sup>, 4<sup>th</sup> > 9<sup>th</sup>, 4<sup>th</sup> > 10<sup>th</sup>, 7<sup>th</sup> > 1<sup>st</sup>, 7<sup>th</sup> > 2<sup>nd</sup>, 7<sup>th</sup> > 3<sup>rd</sup>, 7<sup>th</sup> > 5<sup>th</sup>, 7<sup>th</sup> > 6<sup>th</sup>, 7<sup>th</sup> > 8<sup>th</sup>, 7<sup>th</sup> > 9<sup>th</sup> and 7<sup>th</sup> > 10<sup>th</sup>. These pairwise preferences are collected and stored in the system which are then used to improve the ranked list of relevant articles in future.

This kind of feedback from domain experts has two characteristics: (i) It is in the form of pairwise preferences between articles, (ii) It is not necessarily a total order, i.e. the pairwise preferences span over a partial set of document pairs. Due to these properties, we explored ranking techniques which can learn from pairwise comparisons and a partial order over the set of articles. One of the most popular ranking techniques for this setting is the Ranking Support Vector Machine(Rank-SVM) [16]<sup>4</sup>. Rank-SVM takes document pairs as instances for learning and transforms the problem of learning to rank into a classification problem. Moreover, it uses SVM as a classification technique to learn the model. The aim is to learn a linear ranking function  $f_{\vec{w}}$  such that,

$$(d_i, d_j) \in f_{\vec{w}} \Leftrightarrow \vec{w} \phi(d_i) > \vec{w} \phi(d_j) \quad (2)$$

where,  $\vec{w}$  is a weight vector of the SVM classification model and  $\phi(d_i)$  is the feature vector for  $d_i$ . Specifically, for each document pair, Rank-SVM assigns a label representing the relative relevance of those documents. Then a classification model is trained using this labeled data which is equivalent to finding the weight vector so that the maximum number of pairwise preferences are fulfilled. Further, this classification model is used to obtain a ranking of the

<sup>3</sup>On the web interface, the user saves labels for 10 articles per page, so if on a page, 2 articles are labeled relevant, 8 others will automatically be saved as not relevant. We have an understanding with the users to label articles in batches of 10 to make this work as expected.

<sup>4</sup>Note that even though we can obtain a ranked list of articles using the classifiers presented in Section 4, the objective behind using Rank-SVM is to:(i) improve the ranking of articles classified as *relevant*, (ii) incorporate the active feedback/preferences give the user.

test documents. For the weight vector  $\vec{w}$ , the documents are ordered by their projection onto  $\vec{w}$ .

For FoodSIS, the *relevant* articles given by the classification model explained in Section 4 are then ordered according to the score given by the Rank-SVM model. The efficacy of this interactive learning system was evaluated with experiments that compare Average Precision(AP) [23] scores of the rankings of documents, given by the classifiers and the Rank-SVM. In particular, we compared the AP scores of ranked lists obtained by using classifiers such Naive Bayes (NB) and 3-Class SVM (3CSVM) as well as the ranked list obtained by using Rank-SVM. The ranking for NB is a descending order of probability score output for *relevant* class for the articles and 3CSVM ranking is based on descending order of the distance of the article feature vectors from the separating hyperplane.

To perform this experiment, we used the set of articles and their pairwise preferences given by the user, collected over a span of 60 days. Out of 60 days, we take out random 20 days as test set. For each day in the test set, we obtained a ranked list of articles for that day by training Rank-SVM over pairwise preferences for randomly chosen 40 days. Then, we repeated this procedure 5 times by choosing random 40 days for training Rank-SVM each time and report the average AP score for each test day corresponding to these 5 runs in Table 1. The AP scores are obtained over top 30 articles for each day in the test set. For ordering the articles in the test set using NB and 3CSVM, we sorted the documents according to the score given by the classifiers<sup>5</sup>. Finally, we compared the AP scores for different ranked lists given by NB, 3CSVM and Rank-SVM which were calculated as follows: For any test date,

$$AP = \frac{\sum_{k=1}^n P(k) \times rel(k)}{\text{number of relevant documents}} \quad (3)$$

where,  $k$  is the rank in the sequence of *relevant* articles,  $n$  is the number of *relevant* articles (30 for our experiments),  $P(k)$  is the precision at cut-off  $k$  in the list, and  $rel(k)$  is 1 if the article at rank  $k$  is a *relevant* article, zero otherwise. In Table 1 we present the corresponding results. It can be clearly seen that Rank-SVM gives significantly better AP scores for 19 days compared to NB and for 15 days compared to 3CSVM. Further, the average of AP scores over all 20 days in test set is greater for Rank-SVM and shows a 29% improvement over NB and about 16% improvement over 3CSVM. We also performed a paired t-test over these AP scores to find out if the improvement in the performance of Rank-SVM is statistically significant(overall as well as over individual days). To accomplish this, we used the AP values of 5 random runs of Rank-SVM to compute the p-value using paired t-test. We also report the p-values of *individual* days in Table 1 and highlight the values where the performance improvement was found to be statistically significant. Moreover, it was also observed that the *overall* performance improvement of Rank-SVM compared to NB and 3CSVM is statistically significant with a p-value of 1.168e-06 and 6.316e-05 respectively. Furthermore, performance improvement of 3CSVM compared to NB is statistically significant as well with a p-value of 0.0232.

## 6. DEPLOYMENT

FoodSIS is developed as part of a research project between NEA and IBM. It is currently in its second version and has been deployed within NEA on a research infrastructure. NEA users are using the

<sup>5</sup>However, since NB and 3CSVM do not need pairwise preferences for training, the number of data points used for training of these classifiers was significantly larger than used for Rank-SVM.

Day	Rank-SVM	NB		3CSVM	
	AP	AP	p-value	AP	p-value
1	0.929	0.595	<b>8.97E-06</b>	0.518	<b>3.89E-06</b>
2	0.921	0.566	<b>6.14E-05</b>	0.743	<b>0.00089</b>
3	0.909	0.665	<b>3.49E-05</b>	0.647	<b>2.66E-05</b>
4	0.914	0.84	<b>0.00067</b>	0.917	0.60608
5	0.902	0.73	<b>4.07E-05</b>	0.816	<b>0.0006</b>
6	0.953	0.904	<b>6.23E-05</b>	0.901	<b>4.91E-05</b>
7	0.916	0.831	<b>0.0007</b>	0.751	<b>5.24E-05</b>
8	0.839	0.705	<b>0.00305</b>	0.642	<b>0.00072</b>
9	0.882	0.889	0.8681	0.928	0.999
10	0.901	0.817	<b>0.00081</b>	0.816	<b>0.00079</b>
11	0.903	0.842	<b>0.00721</b>	0.784	<b>0.00064</b>
12	0.947	0.632	<b>5.62E-07</b>	0.765	<b>5.00E-06</b>
13	0.832	0.68	<b>0.00048</b>	0.654	<b>0.00026</b>
14	0.577	0.452	<b>0.00082</b>	0.459	<b>0.00101</b>
15	0.846	0.721	<b>0.00013</b>	0.675	<b>3.90E-05</b>
16	0.723	0.394	<b>8.36E-05</b>	0.749	0.827
17	0.632	0.159	<b>0.01192</b>	0.583	0.3663
18	0.861	0.517	<b>9.24E-06</b>	0.725	<b>0.00035</b>
19	1.0	0.835	<b>1.49E-11</b>	0.831	<b>1.35E-11</b>
20	0.864	0.646	<b>0.00035</b>	0.934	0.981
AVG	<b>0.863</b>	0.671		0.742	

**Table 1: Comparison of Average Precision(AP) scores for rankings based on different approaches**

web interface for past 6 months and have helped us gather substantial amount of labeled data. FoodSIS is now by default the starting point for searches related to food safety within NEA. However, it is still evolving and we are adding more data sources, and more features to enhance its capability. With the current list of data sources that are crawled, we get an average of around 1000 articles a day. Note that this includes articles only from the identified websites and news from sections relevant to food safety. It is clearly impossible for the users to go through such a large number of articles per day. Employing recent advances in machine learning to classify and rank the crawled articles results in presenting a significantly smaller number of highly relevant articles to the users leading to significant improvement in the newsletter generation process efficiency. This section highlights some of the challenges of the deployment and an illustration of how the system was used to generate food safety newsletters. We also discuss the impact of the newsletters for NEA.

### 6.1 Challenges

Development of FoodSIS exposed us to many interesting characteristics of a real world machine learning task. The foremost being availability of labeled data. In real life, labeled data for training supervised approaches is either absent or is available but very small. We had access to very small number of articles of the positive class as seed data. Another typical situation is the lack of negatively labeled historical data. This is in large part because the users identify and store the positive examples but rarely bother to keep the negative examples especially for historically labeled data. The following 3 design choices helped us handle these challenges and provided a substantial performance lift.

1. Web application: One important decision was to build an overall architecture for FoodSIS with the web application as a key way to interact with the users. We focused on build-



	Fortnight 1	Fortnight 2	Fortnight 3	Fortnight 4
MAP	0.58	0.699	0.722	0.862

**Table 2: Performance improvements with increase in labeled data**

ing the web interface such that users find it useful from the beginning. In version 1, when the machine learning was not mature, users had access to all the articles (we did not filter any articles based on classification output) and they could use the keyword search feature if they wanted to query for specific terms. This allowed FoodSIS to be used as a custom search engine in absence of a high performing machine learning model. Easy use of the web interface also allowed them to provide label for each article with very little burden. It is well known that supervised approaches improve with significant increase in training data size and we observed that though such improvement was not so visible over a short time period, it can be clearly observed over longer period (in our case, training data size increases with time). Table 2 shows improvements in retrieval of top 30 relevant articles over a period of 8 weeks. It reports the MAP (MAP is Mean Average Precision i.e. AP (refer to equation 3) averaged over a time period) for top 30 ranks per day where the average is taken over 2 weeks, so we have 4 different MAP scores for the 8 week period. Observe the increasing trend in MAP scores over the 4 fortnights. Over this period, the classification method did not change and the only change was increase in the amount of training data on a weekly basis.

- Exploiting unlabeled and UNIVERSUM datasets: The available labeled data was insignificant to get a reasonable classification performance and hence we explored several options. The first was to understand and exploit additional structural information in the data sources. The first source of structural information was semantic organization of many of the news sources which clearly organize articles under different sections (such as sports, entertainments, health etc). This provides a very clear categorization of the documents into two categories: (i) A set of documents from sections such as sports, entertainment and so forth that are clearly irrelevant for food safety related articles. This corpus of documents we can label as the UNIVERSUM. (ii) The remaining documents from the sections of the news source such as health, food, local news which might contain potential relevant documents but are unlabeled. Classification techniques which exploit the availability of unlabeled and UNIVERSUM datasets helped us achieve better classification.
- Extracting pairwise preferences: The labeling done by users via the web interface not only provides us labels associated with individual articles, but also allows us to extract pairwise preferences as discussed in section 5. These preferences are partial since there is no extra effort involved from users in explicitly providing a correct total order of articles. As experimental evaluation demonstrates, using these preferences to learn a ranking model based on a state of the art technique results in a significant performance gain. As we reported in section 5, the use of pairwise comparisons improved the ranking engine by upto 16%.

It can be clearly seen that though we started with a very small labeled dataset, but careful design of the system and process allowed

us to make the machine learning approach a viable and preferable alternative to information gathering. Another important challenge is to educate the users regarding the difference between the use of machine learning based customized search over a simple information retrieval based search. After the system was made functional, we conducted workshops to highlight this difference and the promise of continual improvement of a machine learning based technique.

## 6.2 Newsletter Generation

As discussed in sections 1 and 2, a team in the environmental health department of NEA generates a weekly newsletter on food safety from information posted on the web. The newsletter is circulated to the entire department. We give an example of a newsletter that was generated in the recent past and describe how FoodSIS contributed to increasing the efficiency of the process, thereby saving time and effort. This example is a real newsletter and it contains summaries of the following 3 articles on topic "mobile food vendors": (i) Food trucks cleanliness, (ii) Food trucks vs restaurants, (iii) Late night mobile food vendors.

These articles were ranked at 1, 5 and 14 on FoodSIS in that order on the day they were published. To illustrate how such a newsletter would have been generated without the use of our system, we perform search on Google under section "News" and select the exact date for each of the above 3 articles. We created 2 disjunctive queries (Google limits a search query to only 32 words) from the 19 keyword phrases provided to us by the NEA users and use those for search. Under such search setting, the first article on "food trucks cleanliness" appeared 10<sup>th</sup> in the search results and the other 2 did not even appear in the top 50. This example illustrates that FoodSIS succeeded in its objective of providing relevant information with minimum effort for the user<sup>6</sup>.

On close observation, it can be seen that the newsletter has a central theme: "mobile food vendors" and the articles span over a period of 2.5 weeks. As part of the enhancements to the web interface, users had emphasized a need for an ability to filter articles based on whether they had been labeled as relevant, not relevant or had been left as unlabeled by the users. The drop down "Show Label" on the web interface provides such a filter. Such simple functionality provided through the web interface can facilitate browsing of articles labeled as relevant in the past and make it easier for the users to include such articles on related topics but from different time periods. As we mention in section 3, all the features of the web interface have been designed with input from the users and considering the newsletter generation process.

## 6.3 Impact to NEA's Operations:

The newsletters keep NEA abreast of the recent developments in topics related to food safety. They can use that information to improve their operational processes. For example, considering the newsletter on "mobile food vendors" that we discussed above, it helps NEA to understand how mobile food vendors are managed in other countries. The number of mobile food vans is on increase in Singapore and NEA wants to constantly improve its processes and regulations to handle these changes. Easy access to information about how they are handled in other countries gives them ideas that can be considered for improving NEA's processes. Newsletters about food recalls in neighboring countries act as alerts and

<sup>6</sup>We acknowledge that such comparison with Google search is simplistic and biased since FoodSIS might have influenced the user's choice of articles for the newsletter. But the difference in the effectiveness of the methods is significant to conclude that FoodSIS was a better tool to help with this example newsletter generation.

they can assess the impact of those recalls on Singapore. Other areas of impact of the newsletters include adopting the technology advances, policy, regulation improvements and capability development programs.

## 7. CONCLUSION

Many groups in an enterprise have the need for keeping up to date with news and information about their domain and they rely on the World Wide Web as a primary source of information. For example, the supply chain management organization is constantly monitoring the web for news related to various echelons of the supply network to understand potential disruptions due to weather on their logistics, impact of political, labor and natural disaster events on productivity and service levels. Similarly the marketing divisions monitor the web (and social media) for sentiment or buzz around product release or marketing campaigns. The need to build custom search engines for different enterprise functions is now a universal need.

In this paper, we have outlined our experience and a framework (leveraging machine learning) for building such systems that are tailored for a department's specific function needs. There are several operational challenges (from availability of labeled data, to leveraging interactive learning) that need to be addressed. We believe additional work and effort in streamlining the use of these techniques and the training of document classifiers would help in the adoption of these techniques broadly in daily enterprise operations.

## 8. ACKNOWLEDGMENTS

We would like to thank Christine Lee and Jo-Ann Lim from NEA, Singapore for labeling the articles by actively using FoodSIS. We also appreciate the valuable suggestions given by them and Christopher Goh, Atikah Rahman, Lee San Tay, Yoke Leng Toh and Kaiyun Chio from NEA, Singapore for improving the user interface. We thank Vikas Sindhwani from IBM Research for useful and insightful discussions and pointers on semi-supervised learning methods.

## 9. REFERENCES

- [1] Last accessed on 20 Feb 2014, Available at: <http://www.foodprocessing-technology.com/features/feature87576/>.
- [2] Last accessed on 20 Feb 2014, Available at: <http://www.taipeitimes.com/News/front/archives/2013/06/02/2003563752>.
- [3] Last accessed on 20 Feb 2014, Available at: <http://www.todayonline.com/singapore/ikea-singapore-recalls-rhubarb-crumble-pie>.
- [4] General information related to microbiological risks in food, world health organization. Last accessed on 20 Feb 2014, Available at: <http://www.who.int/foodsafety/micro/general/en/>.
- [5] Health promotion board, press releases 2012. Last accessed on 20 Feb 2014, Available at: [http://www.hpb.gov.sg/uploadedFiles/HPB\\_Online/News\\_and\\_Events/News/2012/Media%20Release%20-%20Healthier%20CoffeeShop%20Initiative%20Final.pdf](http://www.hpb.gov.sg/uploadedFiles/HPB_Online/News_and_Events/News/2012/Media%20Release%20-%20Healthier%20CoffeeShop%20Initiative%20Final.pdf).
- [6] Estimates of foodborne illness in the united states, us centers for disease control and prevention. 2011. Last accessed on 20 Feb 2014, Available at: <http://www.cdc.gov/foodborneburden/index.html>.
- [7] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, June 2007.
- [8] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 19–26, San Francisco, 2001.
- [9] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, 1998.
- [10] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- [11] Olivier Chapelle, Vikas Sindhwani, and Sathya S. Keerthi. Optimization techniques for semi-supervised support vector machines. *J. Mach. Learn. Res.*, 9:203–233, June 2008.
- [12] Vijil Chenthamarakshan, Prem Melville, Vikas Sindhwani, and Richard Lawrence. Concept labeling: Building text classifiers with minimal supervision. In *International Joint Conference on Artificial Intelligence*, 2011.
- [13] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 595–602, 2008.
- [14] Toh HY., Hishamuddin P., and Tay J. Epidemiology and control of food poisoning outbreak in singapore, 2009-2011. 38(3):62–71.
- [15] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 200–209, 1999.
- [16] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, 2002.
- [17] Ramesh Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 64–71, 2004.
- [18] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May 2000.
- [19] Burr Settles. Active learning literature survey. Technical report, 2010.
- [20] Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [21] Haiqin Yang, Shenghuo Zhu, Irwin King, and Michael R. Lyu. Can irrelevant data help semi-supervised learning, why and how? In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 937–946, 2011.
- [22] Dan Zhang, Jingdong Wang, Fei Wang, and Changshui Zhang. Semi-supervised classification with universum. In *SDM*, pages 323–333, 2008.
- [23] Ethan Zhang and Yi Zhang. Average precision. In LING LIU and M.TAMER AÛSU, editors, *Encyclopedia of Database Systems*, pages 192–193. Springer US, 2009.
- [24] Tong Zhang. The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1191–1198, 2000.