Learning from Multi-topic Web Documents for Contextual Advertisement

Yi Zhang
Microsoft adCenter Labs
yzhan@microsoft.com

Arun C. Surendran
Microsoft adCenter Labs
acsuren@microsoft.com

John C. Platt
Microsoft Research
jplatt@microsoft.com

Mukund Narasimhan
Microsoft Live Search
mukundn@microsoft.com

ABSTRACT

Contextual advertising on web pages has become very popular recently and it poses its own set of unique text mining challenges. Often advertisers wish to either target (or avoid) some specific content on web pages which may appear only in a small part of the page. Learning for these targeting tasks is difficult since most training pages are multi-topic and need expensive human labeling at the sub-document level for accurate training. In this paper we investigate ways to learn for sub-document classification when only page level labels are available - these labels only indicate if the relevant content exists in the given page or not. We propose the application of multiple-instance learning to this task to improve the effectiveness of traditional methods. We apply sub-document classification to two different problems in contextual advertising. One is "sensitive content detection" where the advertiser wants to avoid content relating to war, violence, pornography, etc. even if they occur only in a small part of a page. The second problem involves opinion mining from review sites - the advertiser wants to detect and avoid negative opinion about their product when positive, negative and neutral sentiments co-exist on a page. In both these scenarios we present experimental results to show that our proposed system is able to get good block level labeling for free and improve the performance of traditional learning methods.

Categories and Subject Descriptors

H.4 [**Information Systems**]: Information Systems Applications

General Terms

Algorithms

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

Keywords

sub-document classification, contextual advertising, sensitive content detection, opinion mining

1. INTRODUCTION

Contextual advertisement is a popular advertising paradigm where web page owners allow ad platforms (like Google, Yahoo, Microsoft, etc.) to place ads on their pages that match the content of their sites. Although this is an effective mechanism for advertisers to reach a large audience, it has its problems. Many of these problems arise due to the huge variety of content that can appear on a single web page (e.g. news sites, blogs, etc). Advertisers are very careful about their image and branding, hence they do not want to show their ads on pages with content like violence, pornography etc. (we call these "sensitive content") [6]. For example, General Motors may not want to show Chevy ads on any page with violence or accidents. More specifically, they want to avoid web pages that specifically refer to an accident involving a "Chevy" SUV. Often such content may occur only in a part of the page. For example news pages contain information on a range of topics; in them accidents or war may only be covered in a few lines. Advertisers are also careful about advertising on product review pages. People's opinion on a product may often be mixed - they may like some features of a product but dislike others. On blog review sites and discussion forums where many people express their opinions, again both positive and negative opinions are very common. Advertisers may not wish to advertise on pages which contain negative opinion about their products (or they may wish to specifically target pages which express only positive opinions). Thus it is important to detect and separate different types of opinions appearing on such mixed-content pages. Clearly in the above two applications there is a significant business need for sub-document methods in addition to document level methods i.e. we not only want to tell if a document has some targeted content in it, but we also want to label the parts of the document where the content is present.

Learning for sub-document classification is a unique practical challenge. The scenario is unlike standard text classification where each of the train and test documents is assumed to be about a single topic. The straighforward way to build such a sub-document classifier is to train on entire pages using page-level labels and test on individual blocks. This

method may work well if the target content dominates the positive training pages. But in real web application there are many problems. First, pages can contain unwanted parts like navigation panes, text advertisements, etc. Second, they may contain information on multiple topics. Methods that clean noisy pages may remove the unwanted parts, but may not be able to separate the individual topics in multi-topic pages. Often concepts that advertisers want to target can be broad e.g. the term "sensitive" can apply to many disparate concepts like war, pornography, natural disasters, accidents, etc., and collecting large amounts of broad coverage single-topic training data is difficult. In practice, to build accurate classifiers, we have to collect large amounts of training pages, pre-clean and hand-label the blocks [13]. This data labeling process is expensive and unscalable to many real-world concepts.

1.1 Our contribution

In this paper we investigate different methods to train sub-document classifiers when only page level labels are available. First we study the effectiveness of traditional methods for both document- and sub-document level classification to detect the presence of a desired content even when it appears only in a part of a page. Further we investigate whether the performance of traditional methods can be improved using multiple-instance learning (MIL) techniques. Specifically, we consider a particular example of multiple-instance learning called MILBoost. We show how the problems of sensitive content detection and opinion/sentiment classification for advertising can be considered as 2-class and multi-class versions of MILBoost, and how this approach can improve the performance of traditional classifiers. These are new applications of the MIL framework. In sentiment detection, we show that a Naive-Bayes based MILBoost detector performs as well as the best block detector trained with block-level labels. In short, with only page-level labels, MILBoost can produce a better page-level classifier than traditional methods and at the same time produce a competitive block-level detector, without block-level labeling in the training data.

The rest of the paper is organized as follows. In Section 2 we study the task of sensitive content detection. We show how this task easily fits in to the multiple-instance learning framework. In Section 3 we study the problem of opinion mining for advertising. We show that we can solve this problem by extending MILBoost to a multi-class scenario. Section 4 discusses some related work. In Section 5 we present computational experiments on these two tasks. We study the performance of the state of the art algorithms, and show how MILBoost can improve the performance of the traditional base classifiers. We discuss the results and present some screen-shots that demonstrate these algorithms in action.

2. SENSITIVE CONTENT DETECTION

In online advertising, advertisers often have a content blacklist and they do not want their ads to be shown on web pages that contain those sensitive contents. Usually, sensitive content categories include crime, war, disasters, terrorism, pornography, etc. To satisfy the advertisers' needs, it is important for an advertising platform to have tools that are capable of detecting those sensitive contents on a web page. As long as a web page contains such content blocks, it will be marked as sensitive and the ad display will be turned off. Note that in this paper, we do not differentiate between various sensitive categories (although we could, using the multi-class system we will derive later) but group them as one class labeled as "sensitive". Often, the available training web pages are labeled at the page-level, i.e. the labels only tell whether a page contains sensitive content somewhere in it or not.

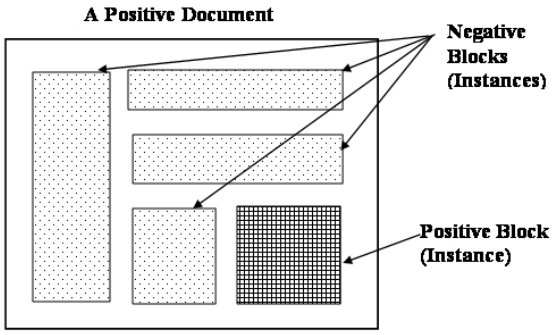

Figure 1: Illustration of a positive web page and its content blocks, one positive block suffices to label the whole page as positive

Figure 1 illustrates the above problem - a web page is divided into a number of content blocks based on its HTML structure. The page in the figure can be thought of as "sensitive" (i.e. labeled "positive") since it contains at least one block with senstive content; otherwise it would be labeled as "negative". Learning in this type of scenario is different from traditional cases where each page is devoted to one topic. If we use the entire page, we run the risk of learning everything on the page as "sensitive". To avoid this problem we need a classifier that can accurately identify the parts of the page that contain the targeted content, and only learn from those. Better still is a classifier that can integrate the two tasks of locating and learning - this type of learning is covered under the multiple-instance learning framework.

2.1 Multiple Instance Learning

Multiple Instance Learning (MIL) [4, 8] refers to a variation of supervised learning where labels of training data are incomplete. Unlike traditional methods where the label of each individual training instance is known, in MIL the labels are known only for groups of instances (also called "bags"). In our above sensitive content detection example, a web page can be considered as a "bag" while each block of text can be thought of as an instance inside this bag. In a 2-class scenario, a bag is labeled positive if at least one instance in that bag is positive, and a bag is labeled negative if all the instances in it are negative. There are no labels on the individual instances. The goal of our MIL algorithm is to produce a content detector at the sub-document (block) level without having the block labels in the training data. This can save significant amount of money and effort by avoiding labeling work at the sub-document level.

Why MILBoost: There are quite a few MIL learning methods available. Among them we choose MILBoost, which combines MIL approaches with ensemble approaches

[19]. The reasons are two-fold: First, the state of the art traditional algorithms use boosting (as we will see later) and we needed a framework to accurately measure the added effectiveness of the MIL framework. Comparing MIL alone against a boosted system will not accurately reveal this difference. But a system using ideas from MIL and boosting when compared to the baseline and boosted systems will tease out the effect of boosting and MIL separately. Secondly, MILBoost has been successfully applied to a similar problem - the problem of training a face detector to detect multiple faces in pictures [19] when only picture level labels are available.

3. SENTIMENT CLASSIFICATION AND DETECTION & OPINION MINING

In the previous scenario, we only had one target topic of interest ("sensitive" content). There are many applications where a user may be interested in a group of topics, and a page may contain one or more of these topics. The occurance of one of the topics may not preclude the occurance of others. Sentiment/opinion mining from review pages or blogs is one such application. It is common to label reviews as "positive" or "negative". However reviews are often not as polar or one sided as the label indicates. An overall negative review may sometimes contain some positive elements and vice versa. Blog review sites or discussion forums usually feature many people expressing varied opinions about the same product. These "mixed" opinions may act as noise during the training of traditional classification methods [13]. Clearly, this calls for a more granular (paragraph- or sentence-level) study of reviews. Once we have a system that can provide labels at a granular level, we can easily find out what aspects of a product that the reviewer likes or dislikes. Advertisers like such analysis since it allows them to stay away from pages that express negative reviews of their products. Alternatively, they may wish to target pages that only express positive reviews about their products. We will show that it is possible to address this problem using multiple instance learning.

3.1 Multi-target MILBoost Algorithm

Traditional MILBoost has been applied to only a 2-class case [19] . To apply MILBoost to the multi-topic detection task, it needs to be extended to a multiclass (or "multi-target") scenario. In this section we show how to derive the multi-target MILBoost algorithm. For example, in the sentiment detection task, the "positive" and "negative" opinions can be treated as the target classes and the "neutral" class as the null class in the MIL setup.

In a multi-target scenario, a bag is labeled as belonging to class k if it contains at least one instance of class k. **As a result, a bag can be multi-labeled since it may contain instances from more than two different target classes**. Figure 2 shows such an example for sentiment detection. There are both positive and negative statements on this review web page and according to the definition, this page shall be labeled both positive and negative. The way we deal with multi-labels is by creating duplicates of a bag with multiple labels and assigning a different label to each duplicate. Within each duplicate bag, MIL will eventually find the instances that match the label of the bag.

Before we derive the algorithm in detail, we would like to discuss the conceptual flow of the training algorithm (see

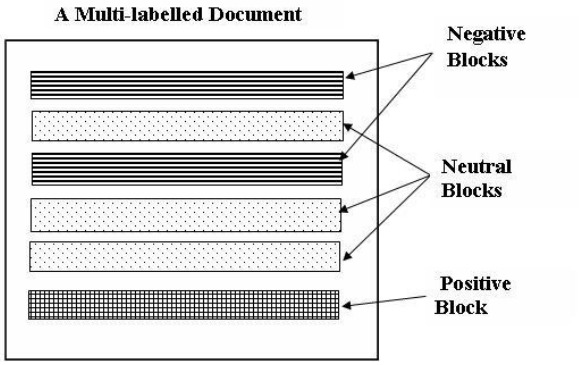

Figure 2: An Example of Multi-labeled Review Web Pages

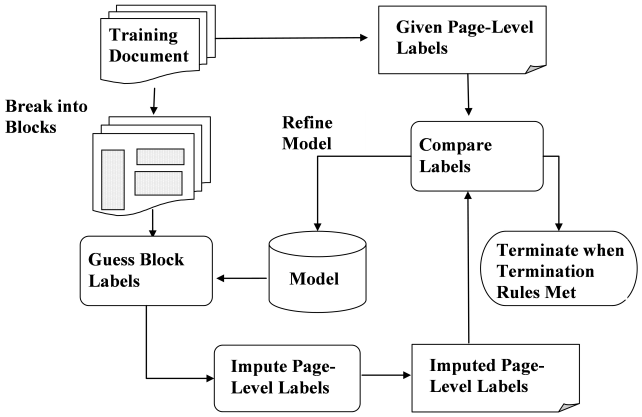

Figure 3: Conceptual Flow of Multiple Instance Learning

Figure 3). First we break the training pages into blocks and guess the block/instance level labels. Then we combine the instance labels to derive bag/page labels. We check if the imputed bag labels are consitent with the given training labels; if not, we adaptively adjust the probability of membership of the training instances until the imputed bag labels become consistent with the given labels. In MILBoost the weight of each instance changes in each iteration according the prediction made by an evolving boosting ensemble.

Initially, all instances get the same label as the bag label for training the first classifier. Subsequent classifiers are trained on reweighted instances based on the output of the existing weak classifiers. A detailed description of a 2-class MILBoost algorithm can be found in [19]. Here we derive the multi-class MILBoost algorithm.

Suppose we have $1 \dots K$ target classes and class 0 is the null class. For each instance x_{ij} of bag B_i, the probability that x_{ij} belongs to class k ($k \in \{1, 2, \dots, K\}$) is given by a softMax function,

$$P_{ijk} = \frac{e^{Y_{ijk}}}{\sum_{c=0}^{K} e^{Y_{ijc}}}$$

where

$$Y_{ijk} = \sum_t \lambda_t y^t_{ijk}$$

is the weighted sum of the output of each classifier in the ensemble with t steps. $\{y^t_{ijk}\}$ is the output score for class k from instance x_{ij} generated by the t^{th} classifier of the ensemble.

Referring to Figure 2, a bag is labeled as belonging to class k if it contains at least one instance of class k. If it contains no blocks with labels $1 \ldots K$, then it is labeled as neutral or the null class. Under this definition, the probability that a page has label k is the probability that at least one of its content block has label k. Given that the probability of each instance belonging to target class k, and assuming that the blocks are independent of each other, the probability that a bag belongs to any target class k ($k > 0$) is

$$P_{ik} = 1 - \prod_{j \in i}(1 - P_{ijk}).$$

This is the "noisy OR" model.

The probability that a page is neutral (or belongs to the null class 0) is the same as the probability that all the blocks in the page are neutral $P_{i0} = \prod_{j \in i} P_{ij0}$.

The log likelihood of all the training data can be given as

$$logLH = \sum_{k=1}^{K} \sum_{\{i|l_i=k\}} \log P_{ik} + \sum_{\{i|l_i=0\}} \log P_{i0} \qquad (1)$$

where l_i is the label of bag B_i.

According to the AnyBoost framework [10], the weight on each instance for next round of training is given as the derivative of the log likelihood function with respect to a change in the score of the instance. Therefore for the target classes,

$$w_{ij} = \frac{\partial logLH}{\partial Y^{l_i=k}_{ijk}} = \frac{1 - P^{l_i=k}_{ik}}{P^{l_i=k}_{ik}} P^{l_i=k}_{ijk}, \forall i \in \{i|l_i > 0\} \qquad (2)$$

For the null class,

$$w_{ij} = \frac{\partial logLH}{\partial Y_{ij0}} = 1 - P_{ij0}, \forall i \in \{i|l_i = 0\} \qquad (3)$$

Understanding the evolution of instance weights: To understand how the re-weighting works, let us look at the weight for a 2-class case [19]:

$$w_{ij} = \frac{\partial logLH}{\partial y_{ij}} = \frac{l_i - p_i}{p_i} p_{ij} \qquad (4)$$

The weight on each instance evolves in the following way to keep the learner focusing on the concepts that are not absorbed so far by the ensemble. First of all, observe that the overall weight is composed of two parts: bag weight $\frac{l_i - p_i}{p_i}$ and instance weight p_{ij}. For an instance in a negative bag, the bag weight is always -1, while the instance weight determines the magnitude of the weight. Generally, negative instances with a high p_{ij} will get a high weight (in magnitude) for next round of training, since they are more likely to cause misclassification at the bag level. For a positive bag, if it is correctly classified (p_i is high), the weight of all the instances in the bag will be reduced. Otherwise, instances with higher p_{ij} within the bag, which are potentially good candidates for "real positive" instances, will standout and get more attention in the next round of training.

Note that in multi-target MILBoost, the weight of each instance is always positive unlike the single-target (2-class) case. It no longer carries the class information by the sign of the weight, as in the single-target case. Similar to the single-target MILBoost, the weights on instances of a target class bag reduce as the ensemble prediction of the bag approaches the bag label. Otherwise, instances with high probability of being the target class will be singled out for next round of training. The weights on the negative instances are also as intuitive as the single-target case. In fact, it can be easily shown that the multi-target MILBoost scheme is consistent with the single-target case.

3.1.1 Combining weak classifiers

Once the $(t+1)th$ classifier is trained, the weight on the classifier λ_{t+1} can be obtained by a line search to maximize the log likelihood function.

3.1.2 Choice of classifier C^t

Just like in any ensemble learning scenario, we can choose a wide variety of base classifiers C^t. In our experiments we show results using Naive Bayes and decision trees. More details are given later in the paper.

A pseudo-code for 2-class MILBoost is shown in Figure 4.

```
Input: Training set T of N bags, each bag B_i with n_i
instances x_ij, bag label l_i ∈ {0,1}, base learner ℓ,
integer M (number of training rounds)
#Initialize weights
for i = 1 : N
  for j = 1 : n_i
    let w^0_ij = 2 * (l_i − 0.5);
  endfor
endfor
for t = 1 : M
  #Train base (weak) classifier with weighted instances
  C_t = ℓ(T, W^{t−1});
  #Combining weak classifiers - Line search for λ_t
  λ_t = argmax_λ logLH; #refer to (1)
  #Update instance weights using Anyboost
  for i = 1 : N
    for j = 1 : n_i
      #Compute instance probability
      let y_ij = Σ_{k=1}^{t} λ_k C_k(x_ij);
      let p_ij = 1 / (1+exp(−y_ij));
    endfor
    #Compute bag probability
    let p_i = 1 − Π_{j∈B_i} (1 − p_ij);
    #Update instance weights
    for j = 1 : n_i
      let w^t_ij = ((l_i − p_i) / p_i) p_ij;
    endfor
  endfor
endfor
Output: ensemble classifier {C_1, C_2, ··· , C_M},
classifier weights:{λ_1, λ_2, ··· , λ_M}
```

Figure 4: Pseudo code for MILBoost

3.1.3 Testing

To test the MILBoost model on a new page, the page is divided into blocks and the block level probabilities are computed using the classifier. The page level probabilities are obtained by combining the block level probabilities using noisy-OR. The block and page level labels are calcualted using thresholds on the probabilities.

4. RELATED WORK

Multiple instance learning has been applied to a wide range of problems such as drug activity prediction [4], image object detection [19, 22], text categorization [1], etc. Andrew et. al. have applied multiple instance learning combined with maximum-margin learning for text categorization [1]. Their work is focused on document-level classification instead of block detection. There are quite a few algorithms developed for multiple instance learning. Besides multiple instance boosting which is applied in this paper, other popular MIL algorithm include diverse density (DD) [9], EMDD [23], citation KNN [20], etc. All of the above algorithms are designed to solve the traditional two-class problem.

Xin et. al. studied the sensitive content detection problem [6]. The classifiers they built are at the page-level and are trained largely with single-topic web pages. There has not been much work in detecting and locating desired content in blocks. Pang et. al. proposed a minimum cut method to identify objective statements in movie reviews [13]. Oliver et. al. described an email tagging system that is able to identify certain actionable items inside an email [3]. However, both works require training data with block-level labels.

Others have studied methods to eliminate noisy parts of web pages like banners, advertisements etc. using style based pre-processing [21] or through summarization [17]. If the main content itself is multi-topic, these two approaches will not be useful. In contrast, our approach elegantly integrates both multi-topic disambiguation and noise removal with the learning step.

5. EXPERIMENTS

5.1 Sensitive Content Detection

In our first of experiments we show how traditional algorithms and their MILBoost-ed versions perform in sensitive content detection. The data set contains two thousand web pages [6] which are labeled at the page level by human annotators. The "sensitive" pages approximately cover war, crime, disaster and terrorism. The label for each web page is binary, either sensitive or nonsensitive. Simple HTML-tag based heuristics were used to split a web pages into content blocks (see Figure 6). Unfortunately, there is no labeling done at the text block level. Therefore, the evaluation has to be done at the web page level. For this specific task, it still makes sense since we only need to know whether a page contains sensitive content and we do not care much about whether all sensitive content blocks from a page are caught. The performance of our approach at the block level will be demonstrated in the next section on sentiment detection experiments.

Two popular base classifiers were used to build the MILBoost ensemble, decision trees [16] and Naive Bayes [11]. Area Under the ROC Curve (**AUC**) was used to evaluate the effectiveness of the various detectors because it reveals the full-spectrum performance when no specific triggering threshold is specified (An ROC curve plots true positive rate versus false positive rate with different classification thresholds, see figure 4. The area under the ROC curve is equivalent to the probability that a positive data point will be ranked above a negative point [5, 2]).

Table 1: Comparison between MILBoost based detectors and their corresponding boosted classifiers for Sensitive Content Detection by AUC. Decision Tree and Naive Bayes (NB) are the two base classifiers. All classifiers were trained with page-level labels only. Bolded numbers indicate that they are *statistically significant than the number in the row above them based on paired t-test at 95% significance level, using 10-fold cross validation.*

Algorithm	Decision Tree	NB
Base	0.555	0.693
Base+Boosting	**0.680**	**0.732**
Base+MILBoost	**0.736**	0.739

We evaluate the performance of various classifiers on the sensitive content task. We start with two different base classifiers - decision trees and Naive Bayes (**NB**). Then we build boosted versions of these classifiers, and also MILBoosted versions of them. All the classifiers are trained with page-level labels. The task is to classify pages into two classes - sensitive and nonsensitive. Both the MILBoost and the non-MILboost versions were run through 30 boosting iterations which end up with an ensemble of 30 classifiers. The depth of a decision tree is fixed at 5. Empirical evidence shows that those parameters appear to give good complexity-performance trade-off i.e. increasing the number of boosting iterations beyond 30 gave minimal improvement. The occurance frequency of word unigrams was used as features.

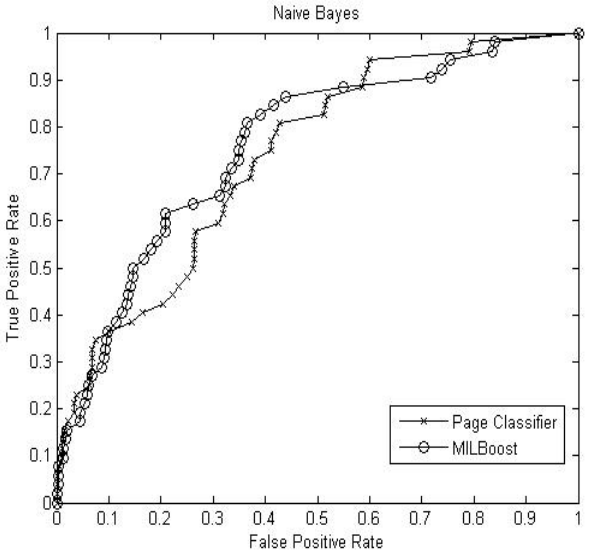

Figure 5: ROC Curve of Boosted Naive Bayes compared to it MILBoost version for sensitive content detection. The corresponding curves using decision trees shows similar performance

Table 1 shows the performance of the detectors trained with different algorithms. The numbers in the table are the average over five 10-fold cross validations. They clearly indicates the advantage of MILBoost detectors over both

the base classifiers and their traditional boosted versions. With a non-linear base classifier such as decision trees the AUC is 0.555; Boosting lifts this performance to 0.68 (a 22.5% improvement in performance). MILBoost further improves this performance by another 8.2% (AUC of 0.736 vs 0.680). Next, we show results using a robust base classifier like Naive Bayes - The boosted NB system gave a 5.6% improvement over the base classifier while the MILBoost version achieved almost the same performance as the boosted page-classifier (AUC of 0.739 vs 0.732). Figure 5 shows the ROC curves comparing the boosted- and the MILBoosted Naive Bayes systems. Althoug the AUC is about the same, the MILBoosted system is almost consistently better than the boosted page-classifier at the early part, where usually the operation point exists. This "early lift" brings practical advantage to the MILBoosted system. Overall, MILBoost framework substantially improves the counterpart page-classifier.

5.1.1 Naive Bayes vs Decision Trees

It is interesting to observe that Naive Bayes performed much better than decision trees in this task. We investigated this issue and found that the reason lies in the number of features the two algorithms use. The decision tree ensemble uses only about 700 keywords while NB theoretically uses the whole vocabulary, which is about 20,000. The bigger feature set enables NB to generalize better at the testing stage. If the feature set used by NB were limited to what decision tree is effectively using, its AUC would drop to 0.64.

A Sensitive Content Detection Demo: The MILBoost detector was used to build a system that is able to highlight sensitive content blocks on a web pages. Figure 6 is a screenshot of the demo. The shaded blocks are identified by the content detector as containing sensitive contents.

5.2 Sentiment Detection

For the previous task we did not have block-level labels, hence we were unable to evaluate it at the block level. To demonstrate block-level performance, we now show single-target MILBoost performance on a 2-class sentiment detection task.

5.2.1 Sentence Level Sentiment Detection

For this task we used the subjectivity dataset from the Cornell movie review data repository [12]. In this data set, 10000 "objective" and "subjective" sentences are labeled. These sentences were extracted from 3000 reviews, which are labeled at the review-level as well. Here a review is a "page" and a sentence is a "block". The MILBoost detector is trained with the review data only using page-level labels, and then evaluated at the sentence-level with sentence level labels. Again, decision trees and Naive Bayes are used as base classifiers. Traditional page-level classifiers using boosted NB and decision trees are also built as benchmark algorithms for comparison. In addition, a page-level classifier using support vector machines (**SVM**) [18, 15] is also trained. SVM is reported to have the best performance in sentiment detection [14]. (We did not build a MILBoost system with SVM as a base classifier becaues it is not straightforward to incorporate the instance weights into an SVM solver).

In addition, since we have sentence-level labels we also built classifiers that are *trained with sentence level labels*. This is the best case scenario, but an expensive proposition since labeling is an expensive task. We want to show that for this task, **we do almost as well as the classifiers that have sentence level labels**.

The results are shown in Table 2. The AUCs of the five algorithms on sentence-level sentiment detection are shown. The numbers are averaged over five 10-fold cross validations.

Table 2: Comparison between MILBoost detectors and their corresponding boosted base classifiers (NB and D.Tree), all trained using page-level labels only. For comparison, performance of the same base classifiers when trained with sentence-level labels are shown in the last row (labeled SENT). SVM results are given for comparison. The task is sentiment detection at the sentence-level. Numbers are AUC. Bolded number indicates statistically significant difference compared with the above row at 95% significance level

Algorithm	Boosted NB	Boosted D.TREE	SVM
Baseline	0.927	0.638	0.780
MILBoost	**0.950**	**0.690**	N/A
SENT	0.953	**0.866**	0.894

The first two lines in Table 2 compare algorithms using only page-level labels. The MILBoost system using NB base classifier achieves the highest AUC (0.950). This performance is comparable with the best sentence detector trained with sentence-level labels (line 3).

It turns out that decision tree is not a good classifier for sentiment detection at the sentence-level. Nonetheless, MILBoost improves the performance by about 10% (from 0.638 to 0.690) over boosted decision trees. The SVM did not do as well as the NB classifiers for sentence classification trained either with the page- or with the sentence-level label. The differences are statistically significant at 95% confidence level. At the page-level, all of them performed very well with AUC above 0.99. The numbers are omitted because of insignificance of the differences.

We can conclude from these results that traditional classifiers trained to work well on pages are not optimized for sentence level detection and MILBoost helps improve their performance.

5.2.2 Multi-class Sentiment Detection

The sentiment detection problem provides a good testbed for multi-target MILBoost. Regular sentiment detection is naturally a three-class problem with "positive", "negative" and "neutral" as class labels. As mentioned before, the "positive" and "negative" classes are the target classes and the "neutral" class is the null class in the MILBoost setup.

Again, we used the Cornell movie review data for the experiment. "Positive" and "negative" reviews are from Polarity dataset v2.0 and "neutral" reviews are from Subjectivity dataset v1.0 (source reviews). Since it is clear that NB performs better than decision trees in these tasks (see the previous results) we only built a multi-class MIL system based on Naives Bayes. The performance of MILBoost Naive Bayes,

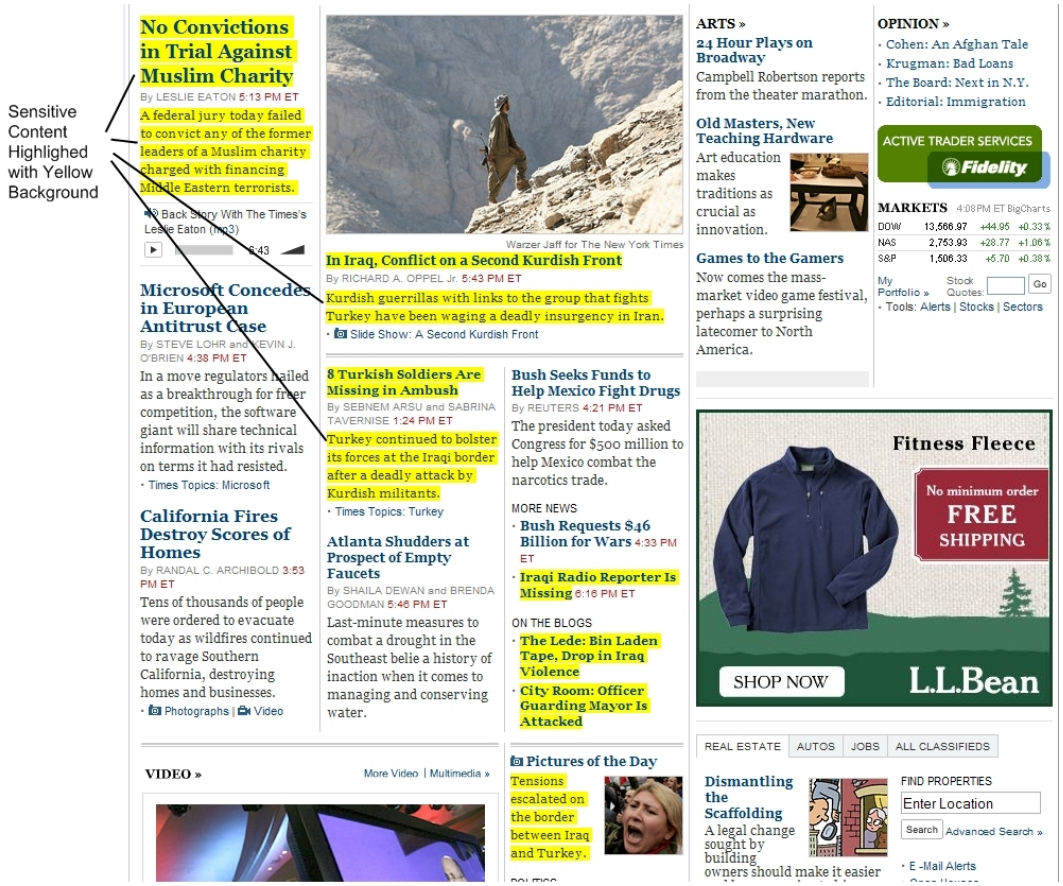

Figure 6: Screen Shot of the output of our MILBoost based sensitive content detection system, with the detected sensitive content (war, terror) highlighted in yellow.

boosted Naive Bayes and SVM for multi-class sentiment detection are compared in Table 3. Since this is a multi-class problem, we report classification accuracy instead of AUC. We can see that our MILBoost based system improves upon the boosted Naive Bayes classifier. The performance using SVM is comparable to the MILBoost system.

Unfortunately, we do not have sentence-level labels therefore the evaluation can only be done at the page-level.

Table 3: Boosted and MILBoosted Naive Bayes, and SVM in Multi-class Sentiment Detection. Bolded number indicates statistically significance compared with the above row.

Algorithm	Accuracy %
Boosted NB	84.8
MILBoost	**86.9**
SVM	87.0

We build a prototype that is able to run the sentiment detection on real movie review sites on the web to identify positive, negative and neutral statements in a movie review web page. Figure 7 shows a screenshot of the demo. The negative blocks are highlighted by dark shades, the negative blocks with light shades while neutral blocks are not highlighted.

5.2.3 *When does MIL improve on traditional methods? - An Analysis Experiment*

We hypothesized before that multiple-instance learning should improve learning of traditional techniques when the amount of mixed content is high. We designed an experiment to verify this. Our experiments were run on a car review dataset which contained 113,000 user reviews from MSN Autos. Each review page is made of the rating score and some review texts. The objective of the learning task to identify negative opinions in review texts. We want to show that as the amount of mixed content increases, MIL based approach can help traditional techniques improve.

Unlike our earlier experiments where the pages only had 0/1 labels, this data set had an overall review rating score from 0-10. We assume that if the rating score is 6 or below, there will be some negative opinions in the review text. This was also verified by reading a sample of the pages. We further split the negative reviews into two subsets, one with rating scores from 0 to 3 (later refered to as "data 0-3")and the other with ratings from 4 to 6 (later refered to as "data 4-6"). Presumably, the percentage of negative sentences in "data 0-3" will be much higher than that in "data 4-6". So if our hypothesis hold right, MIL based techniques should give a bigger boost in the latter data set.

We trained classifiers on each of the datasets "data 0-3" and "data 4-6". The positive training set for each of these

Figure 7: Screen shot of the MILBoost based multi-class sentiment detection system, with negative opinions highlighted in green (dark shade), and positive opinions highlighted in yellow (light shades). Neutral blocks are not highlighted.

experiments remained the same (reviews with rating of 7 or higher). We compare the performance of boosted and MILBoosted Naive Bayes on the two training sets. As we don't have sentence-level labels for this dataset, the evaluation is done at the page (review) level. The results summarized in Table 4 are averages over 10-fold cross validation.

Table 4: Experiment to show that MILBoost helps traditional classifiers in data sets when the ratio of mixed content is higher. Results are given on car review datasets with NB base classifier. The task is to identify negative opinions in reviews. Numbers are AUC. Bolded number indicates statistically significant improvement compared with the number in the same row.

Training Set	MILBoost NB	Boosted NB
data 0-3	0.769	0.763
data 4-6	**0.814**	0.789

From the results in Table 4, we observe that for "data 0-3" with strongly negative reviews, the MILBoost based system did not improve much over the regular boosted system. For "data 4-6" however (which has a larger percentage of mixed content) the MILBoost system gave statistically significant improvement over traditional classifiers of the same complexity. We can conclude that with good quality training data, MILBoost does not give much advantage over traditional methods. However, if the training data has a high ratio of mixed content, then MILBoost does provide significant advantages. (Readers will notice the the results on "data 0-3" are poorer than that on "data 4-6". This is because there are three times as many pages in "data 4-6" as in "data 0-3" and the entire class distribution is highly biased towards positive with positive to negative ratio of 5:1. If we combine the two data sets we perform even better with AUC of 0.818 for MILBoosted NB and 0.81 for boosted NB).

6. CONCLUSION AND FUTURE WORK

In this paper we explored sub-document classification for contextual advertisement applications where the desired content appears only in a small part of a multi-topic web doc-

ument. Specifically we addressed the problem of training such sub-document classifiers when only page level labels are available. We explored various traditional text mining techniques. We also explored the novel application of MILBoost to this problem. We showed that the MILBoost system is able to improve on the performance of the traditional classifiers in such tasks, especially when the percentage of mixed content is high. These systems provide good quality block level labels for free, leading to significant savings in time and cost on human labeling at the block level.

In this paper we did not use the spatial structure of the web pages in our systems. For example, the labels of adjacent content blocks may be correlated, in which case other combination schemes can be used in lieu of the noisy-OR [7, 19]. These can be used in conjuction with the hierarchical page structure ([21]) to improve the performance of the MIL framework. These are all potential directions for future work.

7. REFERENCES

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, 2002.

[2] K. Ataman, W. N. Street, and Y. Zhang. Learning to rank by maximizing AUC with linear programming. In *International Joint Conference on Neural Network*, 2006.

[3] S. Corston-Oliver, E. Ringger, M. Gamon, and R. Campbell. Integration of email and task lists. In *Proceedings of Fourth Conference on Email and Anti-Spam*, 2004.

[4] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[5] J. A. Hanley and B. J. McNeil. The meaning and the use of the area under a receiver operating characteristics (ROC) curve. *Radiology*, 143:29–36, 1982.

[6] X. Jin, Y. Li, and J. T. Teresa Mah. Sensitive webpage classification for content advertising. In *1st International Workshop on Data Mining and Audience Intelligence for Advertising (ADKDD'07)*, 2007.

[7] J. D. Keeler, D. E. Rumelhart, and W.-K. Leow. Integrated segmentation and recognition of hand-printed numerals. In *NIPS-3: Proceedings of the 1990 conference on Advances in neural information processing systems 3*, pages 557–563, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[8] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press, 1998.

[9] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[10] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. *Advances in Neural Information Processing Systems*, 12:512–518, 2000.

[11] M. Minsky and S. Papert. *Perceptrons: an introduction computational geomery*. MIT Press, 1969.

[12] B. Pang and L. Lee. Cornell movie review data repository, http://www.cs.cornell.edu/people/pabo/movie-review-data.

[13] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.

[14] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[15] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines, Technical Report 98-14, Microsoft Research, 1998.

[16] R. J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Manteo, CA, 1993.

[17] D. Shen, Z. Chen, Q. Yang, H.-J. Zeng, B. Zhang, Y. Lu, and W.-Y. Ma. Web-page classification through summarization. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 242–249, New York, NY, USA, 2004. ACM.

[18] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[19] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for objective detection. *Advances in Neural Information Processing Systems*, 18:1417–1426, 2006.

[20] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: A lazy learning approach. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1119–1126, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[21] L. Yi, B. Liu, and X. Li. Eliminating noisy information in web pages for data mining. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 296–305, New York, NY, USA, 2003. ACM.

[22] Q. Zhang, S. Goldman, W. Yu, and J. Fritts. Content-based image retrieval using multiple-instance learning. In *Proceedings of 19th International Conference on Machine Learning*, pages 682–689, 2002.

[23] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *In Proceedings of Neural Information Processing Systems 14*, 2001.