IMPROVING CAUSE DETECTION SYSTEMS WITH ACTIVE LEARNING

ISAAC PERSING AND VINCENT NG

Abstract. Active learning has been successfully applied to many natural language processing tasks for obtaining annotated data in a cost-effective manner. We propose several extensions to an active learner that adopts the margin-based uncertainty sampling framework. Experimental results on a cause detection problem involving the classification of aviation safety reports demonstrate the effectiveness of our extensions.

1. Introduction

Automatic text classification is one of the most important applications in natural language processing (NLP). Supervised text classification systems, however, can be prohibitively expensive to train because a human annotator may have to read a large amount of text in order to label each training instance. In a typical system, a random sampling of documents is chosen for human annotation, but in many cases it is possible to reduce the training set annotation cost with active learning. In active learning, the learner is allowed to choose the instances to be labeled by a human annotator, potentially creating for itself an equally informative training set consisting of a smaller number of labeled instances.

In this paper, we study the application of active learning to *cause detection* using a new dataset involving the Aviation Safety Reporting System (ASRS), which collects voluntarily submitted reports about aviation safety incidents written by flight crews, attendants, controllers, and other related parties. Cause detection, or the determination of *why* an incident happened, is one of the central tasks in the automatic analysis of these reports. Aviation safety experts at NASA have identified 14 causes (also known as *shaping factors*, or simply *shapers*) that may contribute to an aviation safety incident. Hence, cause detection can be recast as a text classification task: given an incident report, determine which of a set of 14 shapers contributed to the incident described in the report.

It is worth mentioning that the accurate acquisition of a classifier for this cause detection task is complicated by several factors. First, the class distributions are *skewed*, with some shapers significantly outnumbering the others. Second, the task involves *multi-label categorization*: a report can be labeled with more than one category, as several shapers can contribute to the occurrence of an incident. Finally, the documents belong to the *same domain*. As a result, they tend to be more similar to each other with respect to word usage than topic-based text classification tasks, making the classes less easily separable.

The three properties mentioned above can pose significant challenges to cause detection, especially in an active learning setting, where classifiers are typically trained on only a small amount of labeled data.Unfortunately, these challenges remain relatively under-studied in existing work on active learning. For instance, though tackled extensively by using instance sampling and re-weighting methods to reduce class skewness, minority class prediction has primarily been studied in a passive learning setting (e.g., Morik et al. [11], Chawla et al. [4] Arbani et al. [1]). Relatively little work has attempted to address class skewness in the context of active learning (e.g., Ertekin et al. [6], Zhu & Hovy [20]). Similarly for multi-label categorization, which can complicate the learning process even when labeled data is abundant, let alone in an active learning setting. However, with a few exceptions (e.g., Brinker [2], Yang et al. [18]), the vast majority of existing work on active learning assumes that each instance can have a single label. Finally, virtually all active learning approaches

University of Texas at Dallas, persingq@hlt.utdallas.edu, vince@hlt.utdallas.edu.

to text classification have been evaluated on the topic-based text classification task, which is easier than cause detection, as discussed above.

We seek to improve an active learner for cause detection that adopts the margin-based uncertainty sampling framework. To address class imbalance and multi-label categorization, we not only investigate existing techniques, but also techniques that have not previously been applied in an active learning setting. In particular, while previous margin-based active learning methods characterize the informativeness of an unlabeled instance using only its distance from the separating hyperplane, we also take into account the information provided by a novel distance metric. In addition, though most previous work on active learning for text categorization is evaluated by plotting a learning curve against the number of labeled documents, works such as Haertel et al. [8] have pointed out that the performance of an active learning system can be highly dependant on the way annotation cost is measured. For that reason we additionally plot a curve against the number of *words* in the selected documents. This allows us to model the fact that longer documents take more effort to label than their short counterparts. Evaluation on 1,333 manually labeled incident reports demonstrate the effectiveness of our proposed extensions.

In the rest of the paper, we first present the 14 shapers, then explain how we preprocess and annotate the reports. After that, we review the standard margin-based active learning framework, and discuss baselines and our extensions to this framework. Finally, we present evaluation results, discuss related work, and conclude.

2. Shaping Factors

As mentioned in the introduction, the task of cause identification involves labeling an incident report with all the shaping factors that contributed to the occurrence of the incident. Table 1 lists the 14 shaping factors, as well as a description of each shaper taken verbatim from Posse et al. [12]. As we can see from Table 1, the descriptions of the shapers are not mutually exclusive. For instance, a lack of **Familiarity** (4) with equipment often implies a deficit in **Proficiency** (10) in its use, so the two shapers frequently co-occur. Similarly, tiredness, which is explicitely listed as one of the impairments covered under **Physical Factors** (7), often results from an extended **Duty Cycle** (3), and hence those two shapers frequently co-occur. These relationships are illustrated in Table 2, which shows the mutual dependence of each pair of shapers as measured by their mutual information in bits × 10^4. In addition, while some classes cover a specific and well-defined set of issues (e.g., **Illusion**), some encompass a relatively large range of situations. For instance, **Resource Deficiency** can include problems with equipment, charts, or even aviation personnel.

3. Dataset

We downloaded our corpus from the ASRS website[1]. The corpus consists of 140,599 incident reports collected during the period from January 1988 to December 2007. Each report is a free text narrative that describes not only why an incident happened, but also what happened, where it happened, how the reporter felt about the incident, the reporter's opinions of other people involved in the incident, and any other comments the reporter cared to include. In other words, a lot of information in the report is irrelevant to (and thus complicates) the task of cause identification.

3.1. Preprocessing.

Unlike newswire articles, at which many topic-based text classification tasks are targeted, the ASRS reports are informally written using various domain-specific abbreviations and acronyms, tend to contain poor grammar, and have capitalization information removed, as illustrated in the following sentence taken from one of the reports.

> HAD BEEN CLRED FOR APCH BY ZOA AND HAD BEEN HANDED OFF TO
> SANTA ROSA TWR.

[1] http://asrs.arc.nasa.gov/

Id	**Shaping Factor**	**Description**	**%**
1	**Attitude**	Any indication of unprofessional or antagonistic attitude by a controller or flight crew member, e.g., complacency or get-homeitis (in a hurry to get home).	2.4
2	**Communication Environment**	Interferences with communications in the cockpit such as noise, auditory interference, radio frequency congestion, or language barrier.	5.5
3	**Duty Cycle**	A strong indication of an unusual working period, e.g., a long day, flying very late at night, exceeding duty time regulations, having short and inadequate rest periods.	1.8
4	**Familiarity**	A lack of factual knowledge, such as new to or unfamiliar with company, airport, or aircraft.	3.2
5	**Illusion**	Bright lights that cause something to blend in, black hole, white out, sloping terrain, etc.	0.1
6	**Physical Environment**	Unusual physical conditions that could impair flying or make things difficult.	16.0
7	**Physical Factors**	Pilot ailment that could impair flying or make things more difficult, such as being tired, drugged, incapacitated, suffering from vertigo, illness, dizziness, hypoxia, nausea, loss of sight or hearing.	2.2
8	**Preoccupation**	A preoccupation, distraction, or division of attention that creates a deficit in performance, such as being preoccupied, busy (doing something else), or distracted.	6.7
9	**Pressure**	Psychological pressure, such as feeling intimidated, pressured, or being low on fuel.	1.8
10	**Proficiency**	A general deficit in capabilities, such as inexperience, lack of training, not qualified, or not current.	14.4
11	**Resource Deficiency**	Absence, insufficient number, or poor quality of a resource, such as overworked or unavailable controller, insufficient or out-of-date chart, malfunctioning or inoperative or missing equipment.	30.0
12	**Taskload**	Indicators of a heavy workload or many tasks at once, such as short-handed crew.	1.9
13	**Unexpected**	Something sudden and surprising that is not expected.	0.6
14	**Other**	Anything else that could be a shaper, such as shift change, passenger discomfort, or disorientation.	13.3

TABLE 1. Descriptions of shaping factor classes. The "%" column shows the percent of labels the shapers account for.

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	3622	3	13	12	2	22	6	12	9	0	44	2	5	2
2	3	5220	3	1	4	13	6	15	4	5	81	40	0	2
3	13	3	2008	3	1	8	389	5	13	3	30	1	8	6
4	12	1	3	3085	4	8	0	1	18	118	56	19	2	63
5	2	4	1	4	221	0	1	1	2	2	0	2	1	9
6	22	13	8	8	0	8035	10	1	0	37	35	0	5	48
7	6	6	389	0	1	10	2610	6	0	3	91	1	3	2
8	12	15	5	1	1	1	6	5524	1	24	239	177	2	33
9	9	4	13	18	2	0	0	1	2888	4	3	18	4	16
10	0	5	3	118	2	37	3	24	4	8131	264	0	5	160
11	44	81	30	56	0	35	91	239	3	264	9964	82	13	498
12	2	40	1	19	2	0	1	177	18	0	82	3067	1	4
13	5	0	8	2	1	5	3	2	4	5	13	1	2704	0
14	2	2	6	63	9	48	2	33	16	160	498	4	0	8015

TABLE 2. Mutual information in bits between shapers × 10^4.

This sentence is grammatically incorrect (due to the lack of a subject), and contains abbreviations such as CLRED, APCH, and TWR. This makes it difficult for a non-aviation expert to understand. To improve readability (and hence facilitate the annotation process), we preprocess each report as follows. First, we expand the abbreviations/acronyms with the help of an official list of acronyms/abbreviations and their expanded forms[2]. Second, though not as crucial as the first step, we heuristically restore the case of the words by relying on an English lexicon: if a word appears in the lexicon, we assume that it is not a proper name, and therefore convert it into lowercase. After preprocessing, the example sentence appears as

had been cleared for approach by ZOA and had been handed off to santa rosa tower.

Finally, to facilitate automatic analysis, we stem each word appearing in the reports.

1	P	N
P	6.4	2.3
N	2.3	89.1

2	P	N
P	10.0	2.8
N	2.8	84.5

3	P	N
P	1.6	1.0
N	1.0	96.5

4	P	N
P	4.2	0.9
N	0.9	94.1

5	P	N
P	0.2	0.0
N	0.0	99.8

6	P	N
P	15.4	3.5
N	3.5	77.6

7	P	N
P	4.1	0.7
N	0.7	94.5

8	P	N
P	7.4	4.0
N	4.0	84.6

9	P	N
P	6.4	0.8
N	0.8	92.0

10	P	N
P	13.9	6.7
N	6.7	72.7

11	P	N
P	23.1	10.1
N	10.1	56.8

12	P	N
P	5.7	1.5
N	1.5	91.4

13	P	N
P	4.4	2.5
N	2.5	90.6

14	P	N
P	14.8	6.6
N	6.6	72.0

S	1	2	3	4	5	6	7
F	74.0	78.4	62.7	83.2	100.0	81.5	85.4

S	8	9	10	11	12	13	14
F	64.9	88.9	67.5	69.7	79.7	63.8	69.2

Table 3. Annotator Agreement Per Class.

3.2. **Human Annotation.** Next, we randomly picked 1,333 preprocessed reports and had two graduate students not affiliated with this research annotate them with shaping factors. After a training session in which we explained to the annotators the definitions of the 14 shapers shown in Table 1, we had each annotator independently label a subset of the reports with shaping factors. To measure inter-annotator agreement, we compute Cohen's Kappa [3] from the two sets of annotations, obtaining a Kappa value of 0.72, which indicates fair agreement. This not only suggests the difficulty of the cause detection task, but also reveals the vagueness inherent in the definition of the 14 shapers.

[2] See `http://akama.arc.nasa.gov/ASRSDBOnline/pdf/ASRS_Decode.pdf`. In the very infrequently-occurring case where the same abbreviation or acronym may have more than expansion, we arbitrarily chose one of the possibilities.

Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Total	52	119	38	70	3	289	348	48	145	38	313	652	42	14
%	3.9	8.9	2.9	5.3	0.2	21.7	26.1	3.6	10.9	2.9	23.5	48.9	3.2	1.1

Table 4. Number of occurrences of each shaping factor in the dataset. The "Total" row shows the number of narratives labeled with each shaper and the "%" row shows the percentage of narratives tagged with each shaper in the 1,333 labeled narrative set.

x **(# Shapers)**	1	2	3	4	5	6
Percentage	53.6	33.2	10.3	2.7	0.2	0.1

Table 5. Percentage of documents with x labels.

Additional statistics on the annotated dataset can be found in Tables 3, 4, and 5. In Table 3, we further analyze annotator agreement on reports having two annotators. For each doubly-annotated report, we first assume its true labels are those applied by annotator 2 and score annotator 1's labels accordingly. We then assume annotator 1's labels are the true labels and score annotator 2's labels. So for example, the top left subtable means that for shaping factor 1 (Attitude), the two annotators agreed that 6.4% of the narratives were positive instances of Attitude, 89.1% of them were negative instances of Attitude, and disagreed on the remaining narratives. In the two long subtables at the bottom, we more directly compare the ease of identifying each of the 14 shapers by showing the F-measures corresponding to the above confusion matrices. So, for example, shaper 5 (Illusion) appears to be easy to identify, because the annotators agreed with respect to Illusion on all doubly-annotated narratives. As mentioned before, this high agreement rate may be attributed to the fact that Illusion covers a specific and well-defined set of issues. Shaper 11 (Resource Deficiency), however, appears harder for annotators agree on, possibly because of the broad range of unrelated situations it covers.

In Table 4, we show how frequently each shaping factor occurs in our 1,333 narrative dataset. This is expressed as both an absolute number of reports in the set having each shaper label, and as a percent of narratives in the set having each shaper as one of its labels. Notice that since some incidents are caused by several shaping factors, the percentages sum to more than 100%.

To get a better idea of how many reports have multiple labels, we categorize the reports according to the number of labels they contain in Table 5. As we can see, nearly half of the reports contain multiple labels.

4. Overview of Margin-Based Active Learning

The idea behind active learning is that a learner can reduce the annotation cost if it is allowed to choose which examples from an unlabeled pool to have manually annotated. The question that naturally follows is: how should an active learner select which examples should be labeled?

Although there are several popular frameworks for selecting active learning examples such as query-by-committee [15] or estimated error reduction [13], we will focus on margin-based uncertainty sampling. We chose uncertainty sampling [10] because it is commonly used. Throughout this paper we use support vector machine (SVM) classifiers due to their robust performance on many classification tasks, and it therefore makes sense to use margin-based uncertainty sampling rather than, for example, using entropy as the uncertainty measure. With margin based sampling, we can directly make use of our classifier's uncertainty about an unlabeled example when deciding which examples to request labels for. Following Schohn & Cohn [14] and Tong & Koller [17], we consider those examples falling closest to an SVM's decision boundary the most uncertain.

5. Baseline Approaches

In this section, we describe two baseline approaches to cause detection with active learning. Both baselines recast cause detection as a set of 14 binary classification problems, one for predicting each shaper. In the binary classification problem for predicting shaper s_i, we create one training instance from each document in the training set, labeling the instance as positive if the document has s_i as one of its labels, and negative otherwise. In essence, we are adopting a *one-versus-all* scheme for creating training instances.

We use the SVM learning algorithm as implemented in the SVM^{light} software package [9] for classifier training. To train and test the SVM classifiers, all words occurring in at least ten narratives in the ASRS dataset are employed as binary-valued features that indicate the presence or absence of a unigram. It is worth mentioning that our primary motivation for recasting the task as a set of binary classification problems is that this approach allows us to perform multi-label categorization in a simple and natural manner. The reason is that a document will receive s_i as its label as long as it is labeled as positive by c_i.

In our experiments, we conduct 5-fold cross validation. Specifically, for each experiment, we divide the 1,333 annotated reports into a test set of about 267 labeled reports and a pool of about 1066 potential active learning reports (henceforth the *unlabeled set*) from which all future active learning reports are drawn. As Algorithm 1 shows, an active learner begins with a training set T of 14 randomly selected documents from the unlabeled pool U. It iteratively requests a labeling of 14 documents from the unlabeled set, then removes the documents from the unlabeled set and adds them to the training set. The difference between systems lies in how reports are selected (line 4).

Algorithm 1: Active Learning Algorithm.

```
Input: $U$: A large pool of unlabeled reports.
1. $T \leftarrow$ 14 randomly selected reports from $U$;
2. Apply manually assigned labels to reports in $T$;
3. $U \leftarrow U - T$;
while $U \neq \emptyset$ do
    4. $H \leftarrow Select(T, U)$;
    5. Apply manually assigned labels to reports in $H$;
    6. $T \leftarrow T \cup H$;
    7. $U \leftarrow U - H$;
end
```

Random is commonly-used baseline in active learning experiments that selects documents from U to add to the training set randomly. The underlying learner is *passive*, as it is not permitted any choice in the documents that are annotated for training.

Before discussing our other baseline, recall that when it is applied to a test report an SVM^{light} classifier outputs a real number. If this number is greater than 0, the report should be labeled positive. Otherwise, it should be labeled negative. The absolute value of this number can be interpreted as the classifier's confidence about the report's predicted label. So for example, while a report which obtains a value of -0.01 and a report which obtains -3.00 should both be labeled negative, the classifier is much more confident about the label of the latter document than the former document.

Keeping this in mind, our **Margin** baseline selects reports to label in line 4 of Figure 1 in the following way. Using the labeled reports in T, it trains 14 binary SVM classifiers c_i, one for each shaper s_i. It then applies the classifiers to the reports in the unlabeled set, for each shaper s_i, choosing the report for which c_i returned the lowest score (in absolute value). Each time a report is selected for one shaper, we remove it from consideration when choosing reports for the remaining classes. In this way, we avoid the problem of possibly choosing fewer than 14 reports in cases where one report obtains the lowest score for multiple classifiers.

6. Active Learning Extensions

In this section, we describe four extensions to the active learning algorithm and a method with which they can be combined to form a better active learner.

Extension 1: Oversampling with BootOS

The problem of minority class prediction occurs frequently in natural language processing tasks. One of the aspects of this cause detection problem that makes it difficult its class skewness, with a few classes such as Resource Deficiency occurring very frequently and many minority classes occurring very infrequently. As shown in Table 4, 9 of the 14 shaping factors occur in fewer than 10% of the reports in the 1,333 document set. Undersampling and oversampling methods have been been successfully applied in supervised learning settings [7] [19] [4] to address the class imbalance problem. When applying active learning to word sense disambiguation, which also often suffers from class imbalance, Zhu & Hovy [20] showed that undersampling caused too many useful majority class examples to be removed in highly-skewed data, but oversampling using their **BootOS** method worked well. With the goal of understanding whether oversampling using **BootOS** can also work well for other tasks, we employ it as our first extension to the margin-based active learning framework for cause detection. More specifically, for active learners using the BootOS extension, we apply BootOS within the *Select* function to oversample the minority (usually positive) class for each of the 14 shapers in the training set T.

To do this, for each shaper we first identify the set X of minority (probably positive) class examples and the difference N in the sizes (in document count) between the positive and negative classes for this shaper in the current training set. We then iteratively cycle through each minority example x in X, using x to create an additional minority class example to add to the training set. We do this by combining each x with its one nearest neighbor until we have added 0.8 × N examples. To do this, we represent each example as a vector of its word features (where 1.0 indicates the presence of a word and 0.0 represents its absence). With this vector representation, we can find an example's nearest neighbor using the city block distance between the two vectors. We combine x with its nearest neighbor by taking the average of the two vectors. It should be noted that our decisions to combine each x with only its 1 nearest neighbor and to expand the minority class by 0.8 × N examples were based on the parameters used by Zhu & Hovy [20]. By training classifiers with these oversampled training sets, we hope that the margin-based uncertainty sampling extensions will select better active learning reports.

Extension 2: Overall Most Confident

Largely due to the imbalance between classes and the fact that some shapers cover a larger set of different situations than others, given any training set, it is likely that a classifier we can train for one shaper will be much better than a classifier we can train for another. Because we have access to all the information about documents in the unlabeled set except for their labels, one way we can compare two classifiers is by looking at how well they separate the reports in the unlabeled set. We hypothesize that a good SVM classifier's hyperplane would not pass through high density regions, whereas a poorer classifier's hyperplane would be more likely to pass through these regions. A poor classifier whose hyperplane passes through multiple high density regions therefore may have more unlabeled points which it cannot confidently classify than a good classifier not passing through many high density regions. This is the motivation behind our Overall Most Confident (OMC) extension. Like **Margin**, it trains 14 classifiers c_i, one for each shaper s_i. Unlike **Margin**, however, it assigns each document in the unlabeled set the smallest (absolute) value returned by any of the classifiers. It then selects the 14 reports that have been assigned the lowest confidence values. This allows the active learner to focus on improving the poorer classifiers.

It has been pointed out that the multi-label nature of some text classification tasks has implications for how active learning can be used [18]. Keeping this idea in mind, we can generalize the OMC extension to exploit the fact that some potential active learning reports may be useful for more than one of the binary shaper classification problems. By default, OMC assigns each unlabeled report the lowest confidence value any of the 14 classifiers gives it. So if, the default version of OMC assigns a report a value of x, that means that at least one of the binary classifiers assigned the report a certainty value of x or lower. What if, instead of assigning a report the lowest certainty value given it by any classifier, OMC instead assigned it the n-th lowest value? The interpretation of this value x would be that at least n of the binary classifiers assigned the report a certainty value of x or lower. Increasing n allows OMC to prefer reports that might be useful for a larger number of classifiers, but at the same time reduces the chance that a chosen point will be especially useful for any of them individually.

Extension 3: Explore All Words

One desirable property of a training set is that it should contain instances of all relevant features for the task being learned. Our Explore All Words (EAW) extension to active learning prefers to request labelings for reports containing many unseen words, since some of these words may be useful for cause detection. This idea is similar to those described by Druck et al. [5] and Sindhwani et al. [16] in that we are determining which features make a potential active learning document most desirable to label.

More generally, EAW can be said to prefer the documents that are least similar to those contained in the current training set. As each of our extensions to active learning (except for BootOS) needs to assign values to each report in the unlabeled set in order to determine which reports will be the most valuable for active learning, it would be useful to formalize EAW by creating a distance metric measuring the distance between a set of reports (the training set) and a report from the unlabeled set. To calculate this distance, we first represent each report as a vector of its unigram features, where $R_i[j] = 1$ only if report i contains feature j. We then represent the set of training reports R_T with another vector, where $R_T[j] = \max_{t \in T} R_t[j]$. Finally, we measure the distance between an unlabeled report vector R_i and a training set vector R_T as $Dist(R_T, R_i) = \sum_{j:R_i[j]>R_T[j]}(R_i[j] - R_T[j])$. This distance formula returns higher values when the unlabeled document R_i contains features not seen in the training set, allowing the EAW extension to prefer reports containing new features.

This extension has a number of obvious shortcomings. Among them is that our document representations do not account for the importance of each word in a document. To address this problem the **tf-idf** version of this extension represents each report with a tf-idf vector rather than a presence or absence vector as before. Hence, in the $Dist$ formula above, $R_i[j]$ is defined as the tf-idf value of term j in document i.

Another shortcoming is that it does not account for the importance of each word to the dataset. The document frequency **df** version of EAW additionally weights each term in the distance formula by its frequency in the original unlabeled set. Hence, the new distance formula is: $Dist(R_T, R_i) = \sum_{j:R_i[j]>R_T[j]} df(j) * (R_i[j] - R_T[j])$ Because we have defined two possible definitions of $R_i[j]$ and two possible distance functions using $R_i[j]$, this extension has four versions.

Extension 4: Document length

It may be possible to exploit our knowledge of the *length* of unlabeled reports to reduce annotation costs. Because reports associated with multiple shapers are on average slightly longer, the Long version of this method will assign each report its length in words and prefer larger values. If we are interested in reducing annotation cost as measured by length of annotated reports, however, the Short version of this method should be chosen. It also assigns reports their length in words, though it prefers the lower values.

Finally, note that these extensions do not have to be used in isolation. In order to combine the values each extension assigns to unlabeled reports, we have to perform three steps. First, we scale the values assigned by each system to the range of 0 to 1. Next, because OMC and Short prefer low values and EAW prefers high values, we transform the values assigned by OMC and Short by subtracting them from 1. Hence if the original OMC or Short value was near 0, the new value will be near 1. Finally, we assign each unlabeled report the sum of the values it was given by the different extensions. The multiple extension version of active learning selects the 14 reports for which this value is highest.

7. Evaluation

As is standard with active learning experiments, we report results in the form of learning curves. Each curve is plotted by computing the micro-averaged F-measure for different amounts of labeled data. This approach to reporting results is preferable to methods such as selecting one F-score and reporting the cost needed to obtain it, or selecting one cost and reporting the F-score obtained with this much annotation because any of these selections we made would be arbitrary, and different choices of annotation cost or F-score could potentially cause us to derive different conclusions. The micro-averaged F-measures we report are computed by aggregating over the 14 shapers as follows. Using the set of about 267 held out test reports, let tp_i be the number of test reports correctly labeled as positive by c_i; p_i be the total number of test reports labeled as positive by c_i; and n_i be the total number of test reports that belong to s_i according to the gold standard. Then,

$$\mathrm{P} = \frac{\sum_i tp_i}{\sum_i p_i}, \mathrm{R} = \frac{\sum_i tp_i}{\sum_i n_i}, \text{and } \mathrm{F} = \frac{2PR}{P+R}.$$

Since there is randomness involved in the selection of the first 14 documents, all results are averaged over three runs of 5-fold cross validation on the 1,333 annotated reports.

To evaluate our extensions to active learning, we begin by evaluating a full-fledged system that makes use of some version of all four extensions described in the previous section. In particular, we employ a full-fledged active learner that uses the Margin baseline along with BootOS, OMC-1, EAW-tfidf-df, and Short. To measure the contribution of each of these extensions to performance, we remove the extensions one-at-a-time in *reverse* order in which they were introduced in the last section and observe the effects.

Specifically, six of the eight figures below (1, 2, 5, 6, 7 and 8) correspond to (1) the two baselines, (2) several versions of the extension being examined, and (3) the system that remains after the extension's removal. To exemplify, Figure 1 shows results of the first experiment, in which Extension 4 is "examined". Hence, the figure contains (1) the two baselines, (2) the two versions of Extension 4 (i.e., Short and Long), and (3) EAW-tfidf-df, which is the system that remains after the removal of Extension 4.

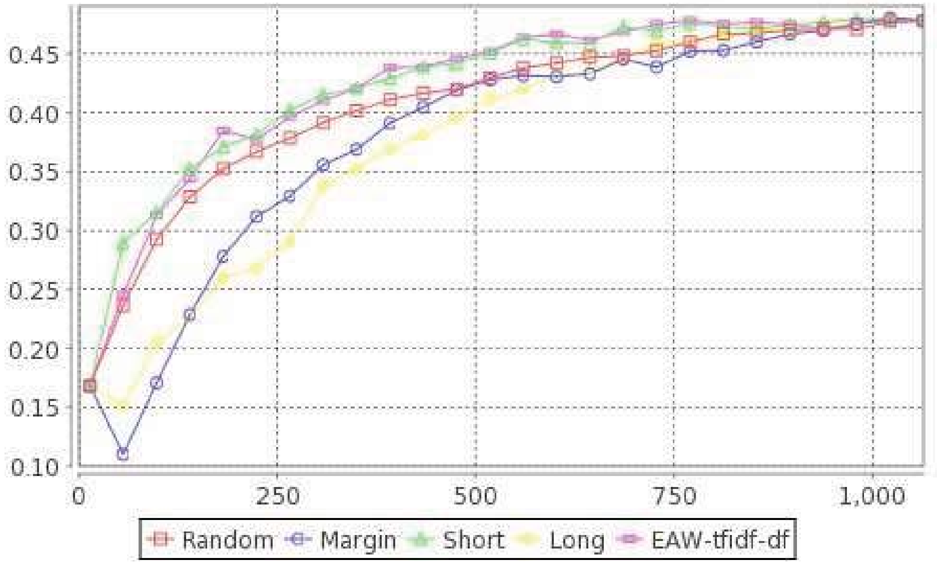

Figure 1. Length: F-measure against # of documents

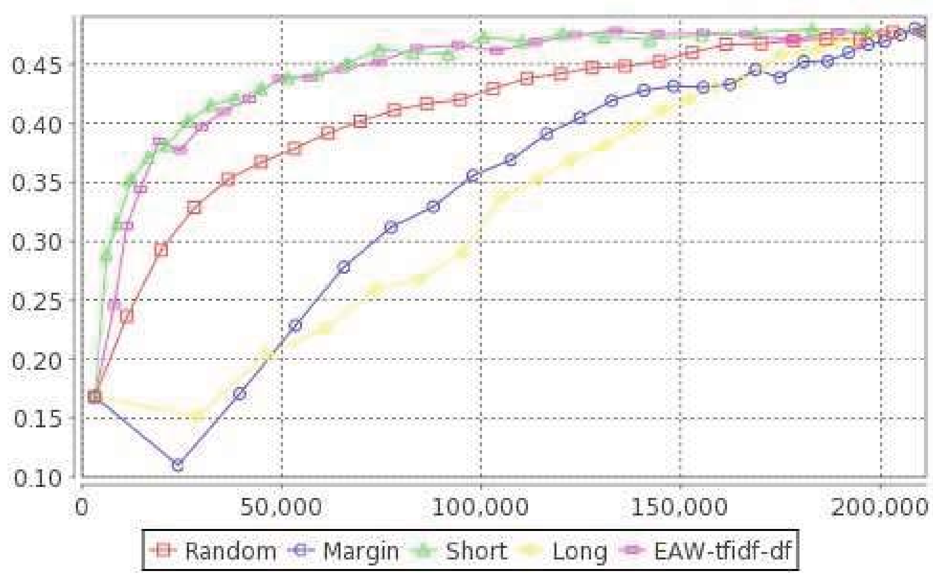

Figure 2. Length: F-measure against word count

Let us begin by examining Extension 4 (Document length). Figures 1 and 2 show results for the entire combined system using the two variations of the Length extension. That is, the Short and Long curves in these figures represent systems that make use of all three other extensions. As we can see, whether we measure annotation cost based on number of reports annotated (Figure 1) or number of words in annotated documents (Figure 2), the combined system using the Short version of this extension does not perform noticeably better than the EAW-tfidf-df system on which it is built. The fact that they perform comparably using both measurements and that the improvements over Random look much larger when measuring cost by word count suggest that EAW-tfidf-df has an inbuilt preference for short documents. This is understandable since it is easier for a word in a short document to have a high tf-idf value, and hence novel words in short documents contribute more in these versions' distance measures than novel words in long documents. Measuring annotation cost by word count, Figure 2 shows that with or without Short, the combined system can achieve results competitive with Random with less than half the annotation cost. Our speculation that Long might work well because of the correlation between document length and number of shapers is shown to be false in both graphs. The Long version hurts the performance of the underlying EAW-tfidf-df system. One possible explanation for this counter-intuitive result is that there are multiple reasons why a narrative might be long. While longer documents are on average associated with more shaping factors than short documents, some documents are long only because they contain excessive information irrelevant to cause detection, thereby making classifiers trained on them less effective.

Because the combined system using the short extension is the best performer overall, we would like to examine what it does in more detail. Figures 3 and 4 show the individual performances for each shaper classifier as measured by document count and word count for the combined system with the short extension. The first thing we notice when examining these graphs is the generally downward curve of the line for shaper 11 (Resource Deficiency). That the F-measure obtained by one classifier decreases as more training data is acquired seems at first counterintuitive. However, when we recall that SVM^{light} constructs a separating hyperplane that minimizes classification error rather than maximizing f-measure, it is not surprising that it would prefer a hyperplane resulting in high recall but low precision for the most frequent shaper when its potential accuracy is hampered by a small training set.

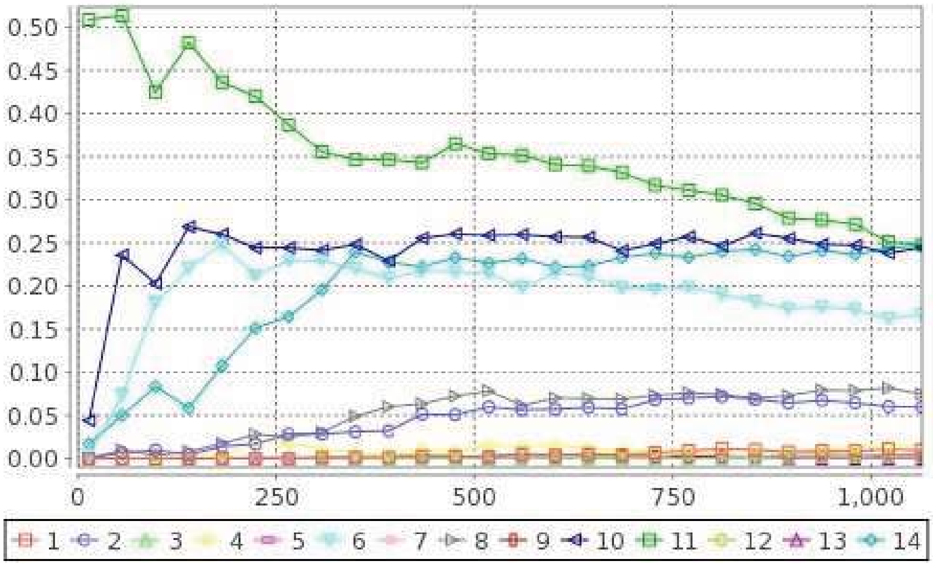

FIGURE 3. F-measure against # of documents per shaper for Short extension

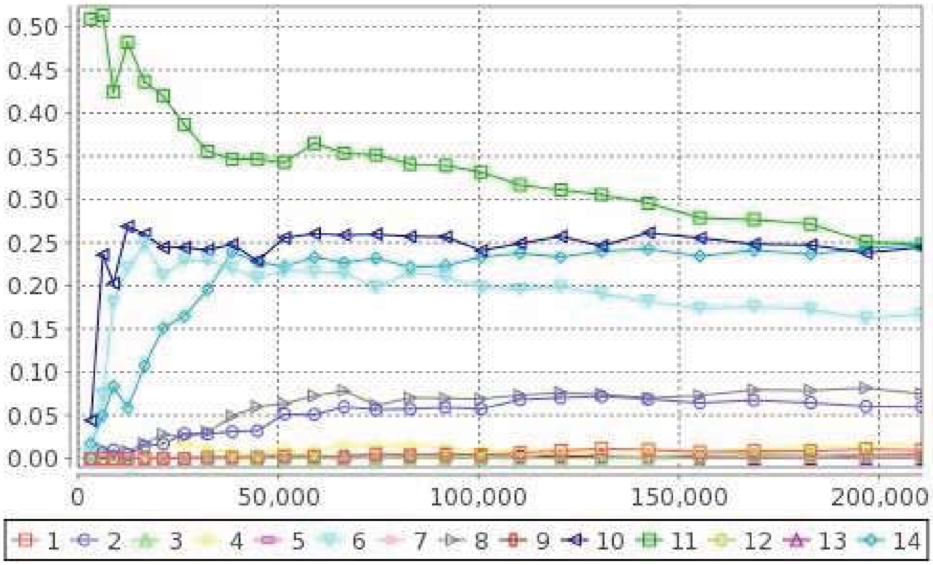

FIGURE 4. F-measure against word count per shaper for Short extension

This does not, however, mean that an error-minimizing SVM algorithm is an inappropriate choice for our sysems' component shaper classifiers. To avoid giving undue weight to the minority classes, the results we report for all of our systems are expressed in terms of micro f-measure. The micro-averaged f-measure formula shown at the beginning of the Evaluation section shows that it is possible for a system's performance to improve even when the performance of one of its component shaper classifiers drops. That is, the micro-averaged f-measure of a system is not merely the average of the f-measures of its component classifiers.

In general, however, these graphs show the unsurprising trend that the classifiers for the most frequent classes tend to do best, improving with increased training data, while minority classes improve very little. This suggests that unsupervised learning approaches or heuristic rule-based techniques might be most useful for minority shaper detection.

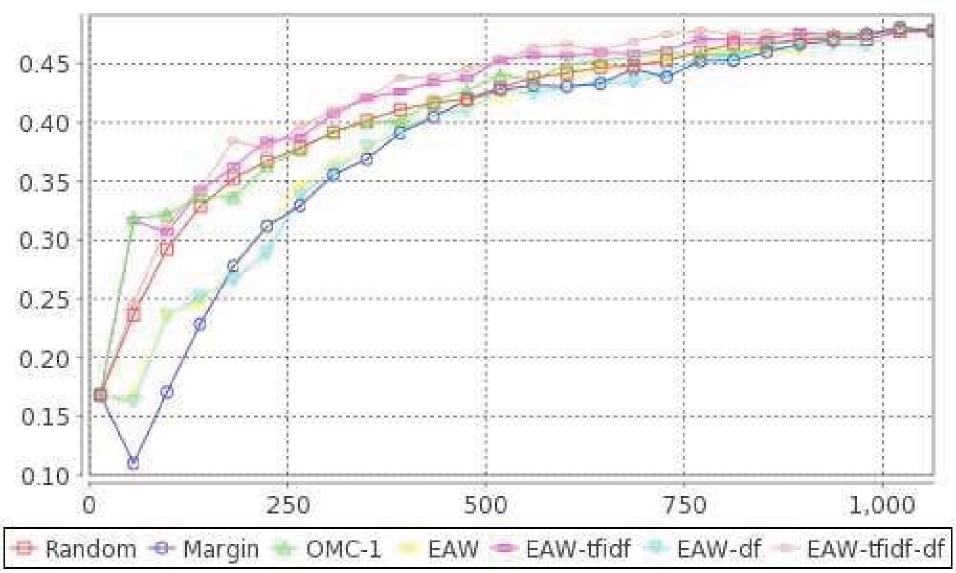

FIGURE 5. EAW: F-measure against # of documents

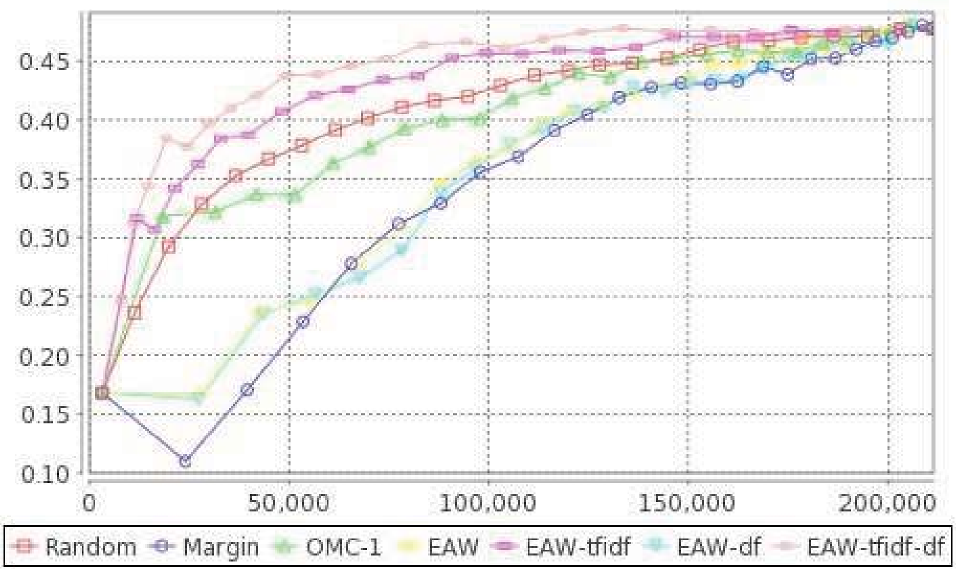

FIGURE 6. EAW: F-measure against word count

Next, we examine Extension 3 (EAW), which prefers reports containing words not yet seen in the training set. Figure 5 shows EAW and EAW-df, the versions which represent reports as binary presence or absence vectors, perform almost as poorly as the Margin baseline.This finding may be related to our discovery that active learners selecting long documents do more poorly than ones selecting short documents. Examining the two distance formulas used for this extension, we see that both make it possible for longer documents to obtain higher distance scores if binary presence or absense representations for the $R_i[j]$ and $R_T[j]$ terms are used. Using tf-idf values for these terms, however, can have the effect of scaling the document representations so that short documents are competitive with long documents.

EAW-tfidf and EAW-tfidf-df by contrast perform quite well. The fact that the tf-idf document representation is required to produce good results justifies our speculation in the previous section that it is not only important to label reports containing words that have not been seen before—it is also important that the new words appear frequently in the selected documents. Similarly, the fact that EAW-tfidf-df outperforms EAW-tfidf tells us that it is important to prefer reports containing words that figure prominently in the dataset over ones containing rarer words.

All these observations are mirrored in the results shown in Figure 6, where we show the same systems, but with annotation cost measured in word count. The fact that the differences we observed are more pronounced when cost is measured by word count is yet more evidence that actual annotation costs can be reduced using our best methods.

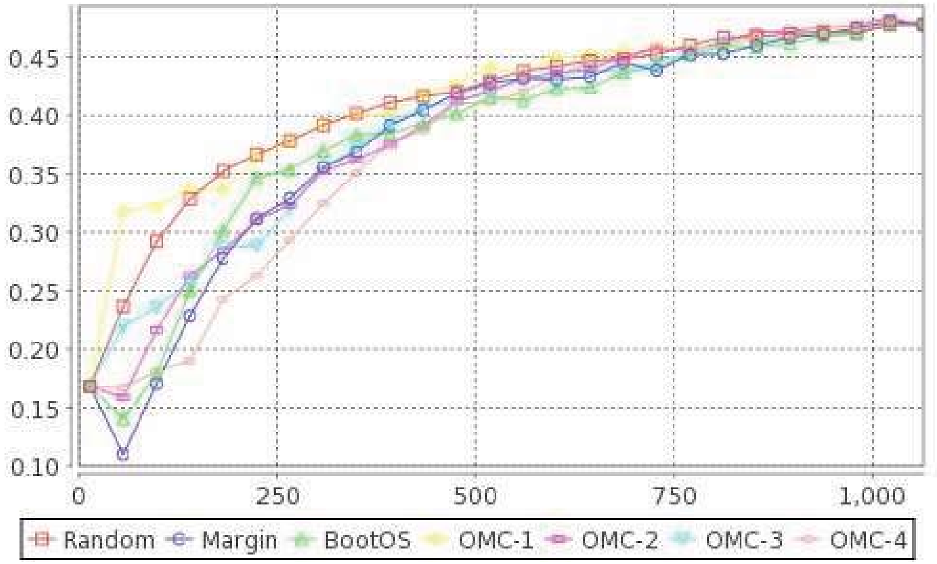

FIGURE 7. OMC & BootOS: F-measure against # of docs

The next extension we examine is Extension 2 (OMC), which prefers reports that are informative for weaker classifiers. In figures 7 and 8, we see that OMC-1 performs much better than BootOS, the system upon which it is built, and performs comparably to Random. This suggests that limiting ourselves to selecting one informative example for each class on each iteration gives our system a huge handicap. OMC-1 obtained a large improvement over BootOS alone by simply preferring reports that we expect to be informative for weaker classifiers rather than strictly limiting the system to one report per classifier per iteration. This intuitively makes sense because some of the binary classification tasks that make up the cause determination problem are much easier to build reasonable classifiers for by virtue of either dealing with more specific, well-defined sets of issues, or by simply being larger classes.

Despite also being permitted to select more examples for weaker classifiers, systems OMC-2, OMC-3, and OMC-4, which are also built on top of BootOS, perform poorly compared to the Random

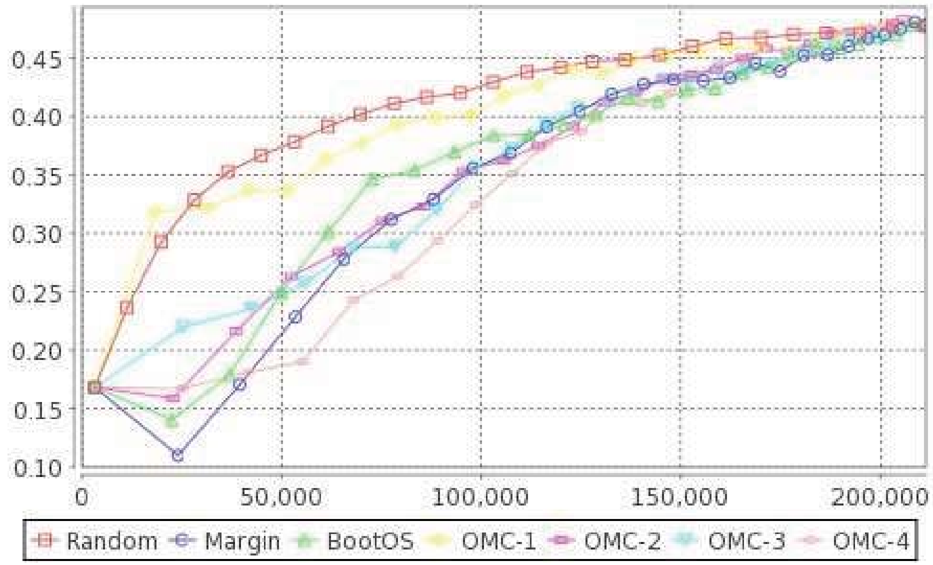

Figure 8. OMC & BootOS: F-measure against word count

baseline, and only the first two of the three compare favorably to even the Margin baseline. Recall that OMC-2, OMC-3, and OMC-4 prefer reports that lie close to 2, 3, or 4 hyperplanes respectively, and therefore should be informative for multiple classifiers. One factor we believe contributes to these systems' poor performance, which was described in the previous section, is that when we look for reports that are close to n hyperplanes, the reports we find tend to be less close to any individual hyperplane than are the reports we find when we search for examples that are close to $n - 1$ hyperplanes.

Finally, we examine BootOS, which is built directly atop the Margin baseline. Though the BootOS extension performs worse than the Random baseline, Figure 8 shows that this is mostly due to trying to choose one informative report for each classifier on each iteration. This also accounts for Margin's poor performance compared to Random. BootOS at least performs better than Margin, which is expected given previous research on BootOS (see Zhu & Hovy [20]).

8. Conclusions

We explored existing and new extensions to an active learner adopting the margin-based uncertainty sampling framework and evaluated them on cause determination. We discovered that, though its multi-label nature and data imbalance complicate active learning, by combining the existing and new extensions, we can build an active learning system that performs better than a random baseline. In particular, measuring annotation cost by training set word count, we showed that our system can reduce annotation cost for achieving reasonable f-scores by over 50%.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ICML*, pages 39–50, 2004.

[2] K. Brinker. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*. Springer, 2006.

[3] J. Carletta. Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.

[5] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 81–90, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[6] S. Ertekin, J. Huang, and C. L. Giles. Active learning for class imbalance problem. In *SIGIR*, pages 823–824, 2007.

[7] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[8] R. Haertel, E. Ringger, K. Seppi, J. Carroll, and M. Peter. Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 65–68, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[9] T. Joachims. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, pages 44–56. MIT Press, 1999.

[10] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, 1994.

[11] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *ICML*, pages 268–277, 1999.

[12] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke, and T. Ferryman. Extracting information from narratives: An application to aviation safety reports. In *Aerospace Conference, 2005 IEEE*, pages 3678–3690, March 2005.

[13] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448, 2001.

[14] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *ICML*, pages 839–846, 2000.

[15] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT*, pages 287–294, New York, NY, USA, 1992. ACM.

[16] V. Sindhwani, P. Melville, and R. D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 953–960, New York, NY, USA, 2009. ACM.

[17] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.

[18] B. Yang, J. T. Sun, T. Wang, and Z. Chen. Effective multi-label active learning for text classification. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 917–926, New York, NY, USA, 2009. ACM.

[19] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transations on Knowledge and Data Engineering*, 18(1):63–77, 2006.

[20] J. Zhu and E. Hovy. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *EMNLP-CoNLL*, pages 783–790, 2007.