InterActive Feature Selection

Hema Raghavan HEMA@CS.UMASS.EDU
140 Governor's Drive
University of Massachusetts
Amherst, MA 01002, USA

Omid Madani MADANI@YAHOO-INC.COM
Yahoo! Inc.,
74 N Pasadena Ave, Pasadena
CA 91103, USA

Rosie Jones JONESR@YAHOO-INC.COM
Yahoo! Inc.,
74 N Pasadena Ave, Pasadena
CA 91103, USA

Editor:

Abstract

We execute a careful study of the effects of feature selection and human feedback on features in active learning settings. Our experiments on a variety of text categorization tasks indicate that there is significant potential in improving classifier performance by feature reweighting, beyond that achieved via selective sampling alone (standard active learning) if we have access to an *oracle* that can point to the important (most predictive) features. Consistent with previous findings, we find that feature selection based on the labeled training set has little effect. But our experiments on human subjects indicate that human feedback on feature relevance can identify a sufficient proportion (65%) of the most relevant features. Furthermore, these experiments show that feature labeling takes much less (about 1/5th) time than document labeling. We propose an algorithm that interleaves labeling features and documents which significantly accelerates active learning. Feature feedback can complement traditional active learning in applications like filtering, personalization, and recommendation.

1. Introduction

A major bottleneck in machine learning applications is the lack of sufficient labeled data for adequate classifier performance as manual labeling is often tedious and costly. Techniques such as active learning, semi-supervised learning, and transduction have been pursued with considerable success in reducing labeling requirements. In the standard active learning paradigm, learning proceeds sequentially, with the learning algorithm actively asking for the labels of instances from a teacher. The objective is to ask the teacher to label the most informative instances in order to reduce labeling costs and accelerate the learning. There has been very little work in supervised learning in which the user (teacher) is queried on something other than whole instances. In experiments in this paper we study the benefits and costs of feature feedback via humans on active learning. To this end we pick document classification Sebastiani (2002) as the learning problem of choice because it represents a case of supervised learning which traditionally relies on example documents as input for training and where users have sufficient prior knowledge on features which can be used to accelerate learning. For example, to find documents on the topic *cars* in traditional supervised learning the user would be required to provide sufficient examples of *cars* and *non-cars* documents. However, this is not the only way in which the information need of a user looking for documents on *cars* can be satisfied. In the *information retrieval* setting the user would be asked to issue a query, that is, state a few words (features) indicating her information need. Thereafter, feedback which may be at a term or at a document level may be incorporated. In fact, even in document classification, a user may use a keyword based search to locate the initial training examples. However, traditional supervised learning tends to ignore the prior knowledge that the user has, once a set of training examples have been obtained. In this work we try to find a marriage between approaches to incorporating user feedback from machine learning and information retrieval and show that active learning should be a dual process – at the term and at the document-level. This has applications in email filtering and news filtering where the user has some prior knowledge and a willingness to label some (as few as possible) documents in order to build a system that suits her needs. We show that humans have good intuition for important features in text classification tasks since features are typically words that are perceptible to the human and that this human prior knowledge can indeed accelerate learning.

In summary, our contributions are: (1) We demonstrate that access to a feature importance oracle can improve performance ($F1$) significantly. over uncertainty sampling with as few as 7 examples labeled. (2) We show that even naive users can provide feedback on features with about 60% accuracy of the oracle. (3) We show that the relative manual costs of labeling features is about 1/5th that of document feedback. We show a method of simultaneously soliciting class labels and feature (4) feedback that improves classifier performance significantly.

We describe the data, SVMs, active learning and performance metrics in Sec. 2 and show how feature selection using an oracle is useful to active learning in Sec. 3. In Sec. 4 we show that humans can indeed identify useful features and show how human-chosen features can be used to accelerate learning in Sec. 5. We relate our work to past work in Sec. 6 and outline directions for the future in section Sec. 7.

2. Experimental setup

Our test bed for this paper comes from three domains:
(1) The 10 most frequent classes from the Reuters-21578 corpus (12902 documents). (2) The 20-Newsgroups corpus (20000 documents from 20 Usenet newsgroups). (3) The first 10 topics from the TDT-2001 corpus (67111 documents in 3 languages from broadcast and news-wire sources).
For all three corpora we consider each topic as a *one versus all* classification problem. We also pick two binary classification problems viz., *Baseball vs Hockey* and *Automobiles vs Motorcycles* from the 20-Newsgroups corpus. In all we have 42 classification problems[1]. All the non-english stories in the TDT corpus were machine translated into English. As features we use words, bigrams and trigrams obtained after stopping and stemming with the Porter stemmer in the Rainbow Toolkit McCallum (1996)

We use linear support vector machines (SVMs) and uncertainty sampling for active learning Scholkopf and Smola (2002); Lewis and Catlett (1994). SVMs are the state of art in text categorization, and have been found to be fairly robust even in the presence of many redundant and irrelevant features Brank et al. (2002); Rose et al. (2002.). Uncertainty sampling Lewis and Catlett (1994) is a type of active learning in which the example that the user (teacher) is queried on is the unlabeled instance that the classifier is most uncertain about. When the classifier is an SVM, unlabeled instances closest to the margin are chosen as queries Tong and Koller (2002). The active learner may have access to all or a subset of the unlabeled instances. This subset is called the pool and we use a pool size of 500 in this paper. The newly labeled instance is added to the set of labeled instances and the classifier is retrained. The user is queried a total of T times.

The *Deficiency* metricBaram et al. (2003) quantifies the performance of the querying function for a given active learning algorithm. Originally deficiency was defined in terms of accuracy. Accuracy is a reasonable measure of performance when the positive class is a sizeable portion of the total. Since this is not the case for all the classification problems we have chosen, we modify the definition of deficiency, and define it in terms of the $F1$ measure (harmonic mean of precision and recall Rose et al. (2002.)). Using notation similar to the original paper Baram et al. (2003), let $\mathcal{U}$ be a random set of P labeled instances, $F1_t(RAND)$ be the average $F1$ achieved by an algorithm when it is trained on t randomly picked examples and $F1_t(ACT)$ be the average $F1$ obtained using t actively picked examples. Deficiency $\mathcal{D}$ is defined as:

$$\mathcal{D}_T = \frac{\sum_{t=init}^{T}(F1_M(RAND) - F1_t(ACT))}{\sum_{t=init}^{T}(F1_M(RAND) - F1_t(RAND))} \quad (1)$$

1. http://www.daviddlewis.com/resources/testcollections/reuters21578/, http://kdd.ics.uci.edu/da -tabases/20newsgroups/20newsgroups.html, http://www.ldc.upenn.edu/Projects/TDT3/

$F1_M(RAND)$ is the $F1$ obtained with a large number (M) of randomly picked examples. For this paper we take $M = 1000$ and $t = 2, 7...42$. When $t = 2$ we have one positive and one negative example. $F1_t(\bullet)$ is the average $F1$ computed over 10 trials. In addition to deficiency we report $F1_t$ for some values of t. Intuitively, if C_{act} is the curve obtained by plotting $F1_t(ACT)$, C_{rand} is the corresponding curve using random sampling and C_M is the straight line $F1_t = F1_M$ then deficiency is the ratio of the area between C_{act} and C_M and the area between C_{rand} and C_M. The lower the deficiency the better the active learning algorithm. We aim to minimize deficiency and maximize $F1$.

3. Oracle Feature Selection Experiments

The oracle in our experiments has access to the labels of all P documents in $\mathcal{U}$ and uses this information to return a list of the k most important features. We assume that the parameter k is input to the oracle. The oracle orders the k features in decreasing information gain order. Given a set of k features we can perform active learning as discussed in the previous section and plot C_{act} for each value of k.

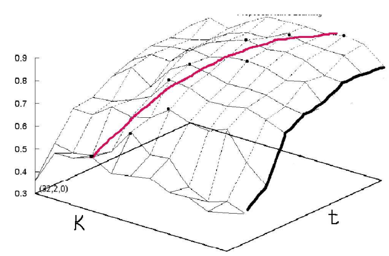

Figure 1: *Average $F1_t(ACT)$ for different values of k. k is the number of features and t is the number of documents.*

Figure 1 shows a plot of $F1_t(ACT)$ against number of features k and number of labeled training examples t, for the *Earnings* category in Reuters. The dark dots represent the maximum F_t for each value of t. The x, y and z axes denote k, t and $F1$ respectively. The number of labeled training examples t ranges from 2...42 in increments of 5. The number of features used for classification k has values from 32,64,128...33718 (all features). The dark band represents the case when all features are used. This method of learning in one dimension is representative of traditional active learning. Clearly when the number of documents is few, performance is better when there is a smaller number of features. As the number of documents increases the number of features needed to maintain high accuracy increases. From the figure it is obvious that we can get a big boost in accuracy by starting with fewer features and then increasing the complexity of the model as the number of labeled documents increase.

The second column shows the deficiency obtained using uncertainty sampling and all features. The third column indicates the average deficiency obtained using uncertainty sampling and a reduced subset of features. The average (over all classes) feature set size n at which this deficiency is attained is shown in column four. In the figure, m and p correspond to the average feature subset size at which $F1_7(ACT, k)$ and $F1_{22}(ACT, k)$ are maximized respectively. The last column shows $F1_{1000}(RAND)$. All 42 of our classification problems exhibit behavior as in figure 1. We report the average deficiency, $F1_7$ and $F1_{22}$ in order to illustrate this point. The second column shows the deficiency obtained using uncertainty sampling and all features. The third column indicates the average deficiency obtained using uncertainty sampling and a reduced subset of features. The average (over all classes) feature set size n at which this deficiency is attained is shown in column four. For each classification problem the best feature subset size is obtained as $argmin_k = \mathcal{D}(k)$. Column 5 shows the average $F1_7(ACT)$ when all features are used. Column 6 shows the average $F1_7(ACT)$ using a reduced feature subset. As for deficiency the best feature subset size for each classification problem is obtained as $argmax_k = \sum_i^{10} \frac{Ora\_IS(..k,7..i)}{10}$. Column 7 contains the average (again over all classes) feature subset size m for which this value of $F1_7(ACT)$ was obtained. Columns 7,8, and 9 show similar results for $F1_{22}(ACT)$ with the best feature subset size at $t = 22$ being denoted by k. The last column shows $F1_{1000}(RAND)$. In all cases cases n, m and p are less than the maximum number of features. Also, for 31 of 42 cases $m \leq p$, meaning that as t increases the complexity of the classifier also needs to increase. For 20-Newsgroups, for all classes we observe that deficiency, $F1_7$ and $F1_{22}$ are best at very small feature subset sizes. For Reuters and TDT there are classes for which a large number of features become important very early (examples: *trade, Bin Laden Indictment, NBA Labor disputes*).

Intuitively, with limited labeled data, there is little evidence to prefer one feature against another. Feature/dimension reduction (by the oracle) allows the learner to "focus" on dimensions that matter, rather than being "overwhelmed" with numerous dimensions right at the outset of learning. It improves example selection as the learner obtains examples to query that are most important for finding better weights on the features that matter. As the number of labeled examples increases, feature selection becomes less important, as the learning algorithm becomes more capable of finding the discriminating hyperplane (feature weights). We experimented with filter based methods for feature selection, which did not work very well (*i.e.*, tiny or no improvements). This is expected given such limited training set sizes (see Fig. 3), and is consistent with most previous findings Sebastiani (2002). Next we determine if humans can identify these *important features*.

4. Human Labeling

Consider our introductory example of a user who wants to find all documents that discuss *cars*. From a human perspective the words *car*, *auto* etc may be important features in documents discussing this topic. Given a large number of documents labeled as on-topic and off-topic, and given a classifier trained on these documents, the classifier may also find these features to be most relevant. With little labeled data (say 2 labeled examples) the classifier may not be able to determine the discriminating features. While in general in machine learning the source of labels is not important to us, in active learning scenarios in which we expect the labels to come from humans we have valid questions to pose: (1) Can humans label features as well as documents? (2) If the labels people provide are noisy through being inconsistent, can we learn well enough? (3) Are features that are important to the classifier perceptible to a human?

Our concern in this paper is asking people to give feedback on features, or word n-grams, as well as entire documents. We may expect this to be more efficient, since documents contain redundancy, and results from our oracle experiments indicate great potential. On the other hand, we also know that synthetic examples composed of a combination of real features can be difficult to label Baum and Lang (1992).

4.1 Experiments and Results

In order to answer the above questions we conducted the following experiment. We picked 5 classification problems which we thought were perceptible to the average person on the street and also represented the broad spectrum of problems from our set of 42 classification problems. We took the two binary classification problems and from the remaining 40 one-versus-all problems we chose three (*earnings*, *hurricane Mitch* and *talk.politics.mideast*). For a given classification problem we took the top 20 features as ranked by information gain on the entire labeled set. In this case we did not stem the data so that features remain as legitimate English words. We randomly mix these with features which are much lower in the ranked list. We show each user one feature at a time and give them two options – *relevant* and *not-relevant/don't know*. A feature is relevant if it helps discriminate the positive or the negative class. We measure the time it takes the user to label each feature. We do not show the user all the features as a list, though this may be easier, as lists provide some context and serve as a summary. Hence our method provides an upper bound on the time it takes a user to judge a feature. We compare this with the time it takes a user to judge a document. We measure the precision and recall of the user's ability to label features. We ask the user to first label the features and then documents, so that the feature labeling process receives no benefit due to the fact that the user has viewed relevant documents. In the learning process we have proposed, though, the user would be labeling documents

Class ↓	$\mathcal{D}_{42}(num\_feat)$			$F1_7(ACT, num\_feat)$			$F1_{22}(ACT, num\_feat)$			$F1_{1000}$
$num\_feat = \rightarrow$	@N	@n	n	@N	@m	m	@N	@p	p	(ACT,N)
Earnings	0.761	0.424	521	0.774	0.837	521	0.897	0.933	260	0.964
Acquisitions	0.4901	0.3476	1043	0.425	0.54	260	0.747	0.816	521	0.927
money-fx	0.509	0.488	260	0.189	0.322	521	0.399	0.498	260	0.65
crude	0.308	0.272	8344	0.293	0.584	65	0.725	0.798	16689	0.829
trade	0.268	0.268	33378	0.372	0.399	4172	0.592	0.633	8344	0.734
interest	0.461	0.461	33378	0.182	0.366	1043	0.384	0.384	33378	0.599
wheat	0.273	0.0106	65	0.233	0.61	32	0.612	0.717	130	0.645
corn	0.564	0.136	32	0.0659	0.421	32	0.26	0.469	32	0.569
money-supply	0.605	0.5755	16689	0.25	0.25	33378	0.348	0.389	16689	0.629
gold	0.163	0.1358	2086	0.362	0.576	130	0.718	0.771	8344	0.733
Reuters	0.421	**0.319**	9579.6	0.345	**0.446**	4015.4	0.569	**0.621**	8464.7	0.727
20-Newsgroups										
alt.atheism	0.741	0.513	67	0.046	0.197	33	0.148	0.259	33	0.45
comp.graphics	0.835	0.371	33	0.007	0.176	67	0.028	0.221	33	0.304
comp.os.ms-windows.misc	0.420	0.244	33	0.116	0.25	67	0.289	0.363	67	0.402
comp.sys.ibm.pc.hardware	0.736	0.530	33	0.031	0.135	67	0.094	0.175	134	0.359
comp.sys.mac.hardware	0.733	0.410	33	0.025	0.194	33	0.126	0.272	33	0.407
comp.windows.x	0.627	0.254	33	0.018	0.158	33	0.116	0.282	134	0.381
misc.forsale	0.529	-0.019	33	0.031	0.345	33	0.111	0.435	8608	0.387
rec.autos	0.423	0.105	33	0.096	0.361	33	0.308	0.405	33	0.429
rec.motorcycles	0.336	-0.172	33	0.067	0.524	67	0.214	0.709	33	0.519
rec.sport.baseball	0.584	0.405	33	0.035	0.200	33	0.098	0.346	67	0.513
rec.sport.hockey	0.407	0.289	33	0.089	0.331	33	0.35	0.476	67	0.641
sci.crypt	0.292	0.151	33	0.094	0.379	33	0.454	0.579	33	0.588
sci.electronics	0.932	0.432	33	0.008	0.086	67	0.025	0.144	67	0.256
sci.med	0.677	0.435	67	0.036	0.223	33	0.084	0.265	67	0.425
sci.space	0.499	0.352	67	0.068	0.26	67	0.397	0.465	33	0.55
soc.religion.christian	0.375	0.296	33	0.134	0.273	33	0.309	0.441	33	0.555
talk.politics.guns	0.667	0.359	33	0.078	0.306	33	0.13	0.337	33	0.464
talk.politics.mideast	0.334	0.161	33	0.15	0.382	67	0.49	0.623	67	0.637
talk.politics.misc	0.789	0.701	67	0.034	0.102	67	0.0906	0.123	33	0.37
talk.religion.misc	0.878	0.595	67	0.013	0.161	67	0.039	0.119	134	0.29
20-Newsgroups	0.602	**0.344**	41.5	0.072	**0.222**	48.3	0.21	**0.29**	487.1	0.446
TDT										
Camb gov. coal.	0.678	0.617	333	0.271	0.347	2669	0.262	0.446	5339	0.711
Hurricane Mitch	0.651	0.490	1334	0.0217	0.251	5339	0.399	0.621	5339	0.854
Pinochet Trial	0.318	0.2862	1334	0.673	0.673	85436	0.722	0.825	667	0.93
Chukwu Octuplets	0.754	0.649	41	0.105	0.357	667	0.219	0.328	667	0.747
Bin Laden Indictment	0.872	0.804	41	0.128	0.153	83	0.103	0.174	42718	0.68
NBA Labor Disputes	0.6455	0.6341	5339	0.21	0.261	5339	0.347	0.432	42718	0.825
Congolese Rebels	0.673	0.560	1334	0.174	0.354	667	0.301	0.511	2669	0.841
APEC Summit Meeting	0.797	0.716	2669	0.129	0.194	10679	0.198	0.298	1334	0.746
Anti-Doping	0.989	0.868	166	0.1	0.167	1334	0.082	0.246	2669	0.449
Car Bomb	0.981	0.937	166	0.0505	0.149	667	0.19	0.19	41	0.728
TDT	0.735	**0.656**	1275.7	0.186	**0.290**	11288	0.282	**0.407**	10416.1	0.751
Baseball versus Hockey	0.710	**0.447**	25	0.587	**0.701**	25	0.785	**0.828**	200	0.963
Auto versus Motorcycle	0.676	**0.321**	125	0.431	**0.724**	62	0.758	**0.860**	31	0.899

Figure 2: *Improvements in deficiency, $F1_7$ and $F1_{22}$ using an oracle to select the most important features. We show results for each metric at N (total number of features for a particular dataset) and at feature set sizes for which the scores are maximized (n, m and p for $\mathcal{D}_{42}$, F_7, and F_{22} respectively). Remember that the objective is to minimize deficiency and maximize $F1$. For each of the three metrics, figures in bold are statistically significant improvements over Uncertainty sampling using all features (the corresponding columns with feature set size of N). We see that with only 7 documents labeled ($F1_7$) the optimal number of features is smaller (48.3 on average for 20-Newsgroups), while with more documents labeled, (22 documents labeled for $F1_{22}$) the optimal number of features is larger (487.1 on average for 20-Newsgroups). When 1000 documents are labeled ($F1_{1000}$) using the entire feature set leads to better scores with the $F1$ measure. This suggests that our best active-learning algorithm would adjust the feature set size according to the number of training documents available.*

Class Problem	Prec. Hum.	Prec. @50	Rec. Hum.	Rec. @50	Avg. Time (secs) Feat.	Avg. Time (secs) Docs
Baseball..	0.42	0.3	0.7	0.3	2.83	12.6
Auto vs ...	0.54	0.25	0.81	0.25	3.56	19.84
Earnings	0.53	0.2	0.66	0.25	2.97	13
...mideast	0.68	0.35	0.55	0.35	2.38	12.93
...Mitch	0.716	0.65	0.56	0.65	2.38	13.19
Average	0.580	0.35	0.65	0.38	2.82	14.31

Figure 3: *Ability of users to identify important features. Precision and Recall against an oracle, of users (Hum.) and an active learner which has seen 50 documents(@50). Average labeling times for features and documents are also shown. All numbers are averaged over users.*

and features simultaneously, so the user would indeed be influenced by the documents he reads. Hence our method is more stringent than the real case. We could in practice ask users to highlight terms as they read documents. Experiments in this direction have been conducted in information retrieval Croft and Das (1990).

Our users were six graduate students and two employees of a company, none of whom were authors of this paper. Of the graduate students, five were in computer science and one from public health. All our users were familiar with the use of computers. Five users understood the problem of document classification but none had worked with these corpora. One of our users was not a native speaker of English. The topics were distributed randomly, and without considering user expertise, so that each user got an average of 2-3 topics. There were overlapping topics between users such that each topic was labeled by 2-3 users on average. A feedback form asking the users some questions about the difficulty of the task was handed out at the end.

We evaluated user feature labeling by calculating their average precision and recall at identifying the top 20 features as ranked by an oracle using information gain on the entire labeled set. Fig. 3 shows these results. For comparison we have also provided the precision and recall (against the same oracle ranking of top 20 features) obtained using 50 labeled examples (picked using uncertainty sampling) denoted by @50. Precision and Recall of the humans is high, supporting our hypothesis that features that a classifier finds to be relevant after seeing a large number of labeled instances are obvious to a human after seeing little or no labeled data (the latter case being true of our experiments). Additionally the Precision and Recall @50 is significantly lower than that of humans, indicating that a classifier like an SVM needs to see much more data before it can find the discriminatory features.

The last column of Fig. 3 shows time taken for labeling features and documents. On average humans require about 5 times longer to label documents than to label features. Note that features may be even easier to label if they are shown in context – as lists, with relevant passages etc. There are several other metrics and points of discussion such as user expertise, time taken to label relevant and non-relevant features and so on, which we reserve for the longer paper. One important consideration though, is that document length influences document labeling time. We found the two to be correlated by $r = 0.289$ which indicates a small increase in time for a large increase in length. The standard deviations for precision and recall are at 0.14 and 0.15 respectively. Different users vary significantly in precision, recall and the total number of features labeled relevant. From the post-labeling survey we are inclined to believe that this is due to individual caution exercised during the labeling process.

Some of the highlights of the post-labeling survey are as follows. On average users found the ease of labeling features to be 3.8 (where 0 is most difficult and 5 is very easy) and documents 4.2. In general users with poor prior knowledge found the feature labeling process very hard. The average expertise (5=expert) was 2.4, indicating that most users felt they had little domain knowledge for the tasks they were assigned. We now proceed to see how to use features labeled as relevant by our naive users in active learning.

5. A Human in the Loop

We saw in Sec. 3 that feature selection coupled with uncertainty sampling gives us big gains in performance when there are few labeled examples. In Sec. 4 we saw that humans can discern discriminative features with reasonable accuracy. We now describe our approach of applying term and document level feedback simultaneously in active learning.

5.1 Algorithm

Let documents be represented as vectors $X_i = x_{i1}...x_{i|F|}$, where $|F|$ is the total number of features. At each iteration the active learner not only queries the user on an uncertain document, but also presents a list of f features and asks the user to label features which she considers relevant. The features to be displayed to the user are the top f features obtained by ordering the features by information gain. To obtain the information gain values with t labeled instances we trained a classifier on these t labeled instances. Then to compute information gain, we used the 5 top ranked (farthest from the margin) documents from the unlabeled set in addition to the t labeled documents. Using the unlabeled data for term level feedback is very common in information retrieval and is called pseudo-relevance feedback Salton (1968).

The user labels some of the f features which he considers discriminative features. Let $\vec{s} = s_1...s_{|F|}$ be a vector containing weights of relevant features. If a feature number i that is presented to the user is labeled as relevant then we set $s_i = a$, otherwise $s_i = b$, where a and b are parameters of the system. The vector $\vec{s}$ is noisier than the real case because in addition to mistakes made by the user we lose out on those features that the user might have considered relevant, had he been presented that feature when we were collecting relevance judgments for features. In a real life scenario this might correspond to the lazy user who labels few features as relevant and leaves some features unlabeled in addition to making mistakes. If a user had labeled a feature as relevant in some past iteration we don't show the user that feature again.

We incorporate the vector $\vec{s}$ as follows. For each X_i in the labeled and unlabeled sets we multiply x_{ij} by s_j to get X'_{ij}. In other words we scale all relevant features by a and non-relevant features by b. We set $a = 10$ and $b = 1$. [2]

By scaling the important features by a we are forcing the classifier to assign higher weights to these features. We demonstrate this with the following example. Consider a linear SVM, $|F| = 2$ and 2 data points $X_1 = (1, 2)$ and $X_2 = (2, 1)$ with labels $+1$ and -1 respectively. An SVM trained on this input learns a classifier with $w = (-0.599, +0.599)$. Thus both features are equally discriminative. If feature 1 is considered more discriminative by a user, then by our method $X'_1 = (10, 2)$ and $X'_2 = (20, 1)$ and $w' = (0.043, -0.0043)$, thus assigning higher weight to f_1. Now, this is a "soft" version of the feature selection mechanism of Sec. 3. But in that case the Oracle knew the ideal set of features and we look upon that set of experiments as a special case where $b = 0$. We expect that human labels are noisy and we do not want to zero-out potentially relevant features.

5.2 Experiments and Results

To make our experiments repeatable (to compute average performance and for convenience) we simulate user interaction as follows. For each classification problem we maintain a list of features that a user might have considered relevant had he been presented that feature. For these lists we used the judgments obtained in Sec. 4. Thus for each of the 5 classification problems we had 2-3 such lists, one per user who judged that topic. For the 10 TDT topics we have topic descriptions as provided by the LDC. These topic descriptions contain names of people, places and organizations that are key players in this topic in addition to other keywords. We used the words in these topic descriptions to be equal to the list of relevant features. Now, given these lists we can perform the simulated HIL (*Human in the Loop*) experiments for 15 classification problems. At each iteration f features are shown to the user. If the feature exists in the list of relevant features, we set the corresponding bit in $\vec{s}$ and proceed with the active learning as in Sec. 5.1. Fig. 4 shows the performance of

2. We picked our algorithm's parameters based on a quick test on 3 topics (baseball, earnings, and acquisitions) using the oracle features of Sec. 3.

the HIL experiments. Like before we report deficiency, $F1_7$ and $F1_{22}$. As a baseline we also report results for the case when the top 20 features as obtained by the information gain oracle are input to the simulated HIL experiments (this represents what a user with 100% precision and recall would obtain by our method). The Oracle is (as expected) much better than plain Uncertainty sampling, on all 3 measures, reinforcing our faith in the algorithm of Sec. 5.1. The performance of the *HIL* experiments is almost as good as the Oracle, indicating that user input (although noisy) can help improve performance significantly. The plot on the right is of $F1_t(HIL)$ for *hurricane Mitch*. As a comparison $F1_t(ACT)$ is shown. The HIL values are much higher than for uncertainty sampling.

We also observed that relevant features were usually spotted in very early iterations. For the *Auto vs Motorcycles* problem, the user has been asked to label 75% (averaged over multiple iterations and multiple users) of the oracle features at some point or the other. The most informative words (as determined by the Oracle) – *car* and *bike* are asked to the user in very early iterations. The label for *car* is always (100% of the times) asked, and 70% of the time the label for this word is asked to the user in the first iteration itself. This is closely followed by the word *bike* which the user is queried on within the first 5 iterations 80% of the time. Most relevant features are asked within 10 iterations which makes us believe that we can stop feature level feedback in 10 iterations or so. When to stop asking questions on both features and documents and switch entirely to documents remains an area for future work.

Dataset	$\mathcal{D}_{42}$			$F1_7$			$F1_{22}$		
	Unc	Ora	HIL	Unc	Ora	HIL	Unc	Ora	HIL
Baseball	0.71	0.41	0.46	0.49	0.63	0.60	0.63	0.79	0.70
Earnings	0.90	0.64	0.64	0.61	0.79	0.73	0.80	0.85	0.86
Auto vs Motor	0.82	0.33	0.60	0.35	0.62	0.60	0.71	0.83	0.73
Hurr. Mitch	0.89	0.38	0.38	0.04	0.46	0.60	0.08	0.63	0.58
mideast	0.49	0.28	0.28	0.14	0.28	0.29	0.32	0.49	0.49
TDT (avg)	0.86	0.77	0.89	0.09	0.21	0.24	0.18	0.32	0.22

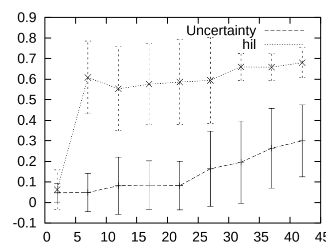

Figure 4: *Improvement in deficiency due to human feature selection. The graph on the right shows Human Feature Selection for* Hurricane Mitch *with the x-axis being the number of labeled documents and y-axis $F1(HIL)$; the difference between these two curves is summarized by the deficiency score. The $F1_7$ and $F1_{22}$ scores show the points on the two curves where 7 and 22 documents have been labeled with active learning. The difference between no feature feedback (Unc) and human-labeled features (HIL) is greatest with few documents labeled, but persists up to 42 documents labeled.*

6. Related Work

Our work is related to a number of areas including query learning, active learning, use of (prior) knowledge and feature selection in machine learning, term-relevance feedback in information retrieval, and human-computer interaction, from which we can cite only a few.

Our proposed method is an instance of query learning and an extension of standard ("pool-based") active learning which focuses on selective sampling of instances (from a pool of unlabeled data) alone Cohn et al. (1994). Although query learning can be very powerful in theory Angluin (1992), arbitrary queries may be difficult to answer in practice Baum and Lang (1992), hence the popularity of pool-based methods, and the motivation for studying the effectiveness and ease of predictive feature identification by humans in our application area. That human prior knowledge can accelerate learning has been investigated by Pazzani and Kibler (1992), but our work differs in techniques (they use prior knowledge to generate horn-clause rules) and applications. Beineke et al. (2004) uses human prior knowledge of co-occurence of words to improve classification of product reviews. None of these works however consider the use of prior knowledge in the active learning setting. Our work is unique in the field of active learning because we consider the case of querying a user on something other than instances and probably the work of Godbole et al. (2004) comes closest to this. Our study of the human factors (such as quality of feedback and costs) is also a major differentiating theme between our work from previous work in incorporating prior knowledge which did not address this issue, or might have assumed experts in machine learning taking a role in training the system Schapire et al. (2002); Wu and Srihari (2004); Godbole et al. (2004). We only assume knowledge about the topic of interest. Our algorithmic techniques and the studied modes of interaction differ and are worth further comparison.

In both Wu and Srihari (2004); Schapire et al. (2002), prior knowledge is given at the outset which leads to a "soft" labeling of the labeled or unlabeled data that is incorporated into training via modified boosting or SVM training. However, in our scheme the user is labeling documents and features simultaneously. We expect that our proposed interactive mode has an advantage over requesting prior knowledge from the outset, as it may be easier for the user to identify/recall relevant features while labeling documents in the collection and being presented with candidate features. The work of Godbole et al. (2004) puts more emphasis on system issues and focuses on multi-class training rather than a careful analysis of effects of feature selection and human efficacy. Their proposed method is attractive in that it treats features as single term documents that can be labeled by humans, but they also study labeling features before documents (and only in an "oracle" setting, *i.e.*, not using actual human annotators), and do not observe much improvements using their particular method over standard active learning in the single domain (Reuters) they test on.

7. Conclusions and Future Work

We proved experimentally that for learning with few labeled examples good feature selection is extremely useful. As the number of examples increases, the vocabulary (feature set size) of the system also needs to increase. A teacher, who is not knowledgeable in machine learning, can help accelerate training the system in this early stage, by pointing out potentially important features or words. We did experiments showing how the complexity of the best classifier increases with increase in the number of labeled instances. We also conducted a user study to see how well naive users performed as compared to a feature oracle. We used our users' outputs in realistic *human in the loop* experiments and found significant increase in performance.

This paper points to two main tracks for further exploration. The first question that needs to be tackled is – what is the minimal set of questions that the active learner needs to ask, and how to incorporate the feedback, to learn as quickly as possible. The second aspect then is how to translate what the learner needs to know, into a question that the teacher can understand. In our case, the learner asked the teacher labels on word features and documents, both of which required little effort on the part of the teacher to understand what was being asked of him. Our subjects did indeed find labeling words without context a little hard, and suggested that context might have helped. We intend to conduct an exhaustive user study, to see what users can perceive easily, and to incorporate these into learning algorithms.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor. We would also like to thank our users who willingly labeled data.

References

D. Angluin. Comp. learning theory: survey and selected bibl. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 351–369, 1992.

Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. In *Proc. of ICML-2003*, pages 19–26, 2003. URL `citeseer.ist.psu.edu/luz03online.html`.

E. B. Baum and K. Lang. Query learning can work poorly when human oracle is used. In *Intern Joint Conf in Neural Netwroks*, 1992.

Philip Beineke, Trevor Hastie, and Shivakumar Vaithyanathan. The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 263–270, Barcelona, Spain, July 2004.

Janez Brank, Marko Grobelnik, Natasa Milic-Frayling, and Dunja Mladenic. Feature selection using linear support vector machines. Technical report, Microsoft Research, 2002.

D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

W. B. Croft and R. Das. Experiments with query acquisition and use in document retrieval systems. In *SIGIR '90*, 1990.

Shantanu Godbole, Abhay Harpale, Sunita Sarawagi, and Soumen Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD*, pages 185–196, 2004.

David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of ICML-94*, pages 148–156, 1994. URL `citeseer.ist.psu.edu/135290.html`.

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

M. J. Pazzani and D. Kibler. The role of prior knowledge in inductive learning. *Machine Learning, 9 , 54-97.*, 9, 1992.

Tony G. Rose, Mark Stevenson, and Miles Whitehead. The reuters corpus vol. 1 - from yesterday's news to tomorrow's language resources. In *Inter. Conf. on Lang. Resources and Evaluation*, 2002.

G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill, 1968.

R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002. URL `citeseer.ist.psu.edu/schapire02incorporating.html`.

Scholkopf and Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surverys*, 2002.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002. ISSN 1533-7928.

Xiaoyun Wu and Rohini Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proc. of KDD*, 2004.