Active Dual Supervision: Reducing the Cost of Annotating Examples and Features

Prem Melville
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
pmelvil@us.ibm.com

Vikas Sindhwani
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
vsindhw@us.ibm.com

Abstract

When faced with the task of building machine learning or NLP models, it is often worthwhile to turn to active learning to obtain human annotations at minimal costs. Traditional active learning schemes query a human for labels of intelligently chosen examples. However, human effort can also be expended in collecting alternative forms of annotations. For example, one may attempt to learn a text classifier by labeling class-indicating words, instead of, or in addition to, documents. Learning from two different kinds of supervision brings a new, unexplored dimension to the problem of active learning. In this paper, we demonstrate the value of such active dual supervision in the context of sentiment analysis. We show how interleaving queries for both documents and words significantly reduces human effort – more than what is possible through traditional one-dimensional active learning, or by passive combinations of supervisory inputs.

1 Introduction

As a canonical running example for the theme of this paper, consider the problem of *sentiment analysis* (Pang and Lee, 2008). Given a piece of text as input, the desired output is a *polarity score* that indicates whether this text expresses a positive or negative opinion towards a topic of interest. From a machine learning viewpoint, this problem may be posed as a typical binary text classification task. Sentiment, however, is often conveyed with subtle linguistic mechanisms such as sarcasm, negation and the use of highly domain-specific and contextual cues. This brings a multi-disciplinary flavor to the problem, drawing interest from both Natural Language Processing and Machine Learning communities.

Many methodologies proposed in these disciplines share a common limitation that their performance is bounded by the amount and quality of labeled data. However, they differ conceptually in the *type* of human effort they require. On one hand, supervised machine learning techniques require human effort in acquiring *labeled examples*, which requires reading documents and annotating them with their aggregate sentiment. On the other hand, dictionary-based NLP systems require human effort in collecting *labeled features*: for example, in the domain of movie reviews, words that evoke positive sentiment (e.g., “mesmerizing”, “thrilling” etc) may be labeled positive, while words that evoke negative sentiment (e.g., “boring”,“disappointing”) may be labeled negative. This kind of annotation requires a human to condense prior linguistic experience with a word into a sentiment label that reflects the net emotion that the word evokes.

We refer to the general setting of learning from both labels on examples and features as *dual supervision*. This setting arises more broadly in tasks where in addition to labeled documents, it is frequently possible to provide domain knowledge in the form of words, or phrases (Zaidan and Eisner, 2008) or even more sophisticated linguistic features, that associate strongly with a class. Recent work (Druck et al., 2008; Sindhwani and Melville, 2008) has demonstrated that the presence of word supervision can greatly reduce the number of labeled documents

required to build high quality text classifiers.

In general, these two sources of supervision are not mutually redundant, and have different annotation costs, human response quality, and degrees of utility towards learning a dual supervision model. This leads naturally to the problem of *active dual supervision*, or, how to optimally query a human oracle to simultaneously collect document *and* feature annotations, with the objective of building the highest quality model with the lowest cost. Much of the machine learning literature on active learning has focused on one-sided example-only annotation for classification problems. Less attention has been devoted to simultaneously acquiring alternative forms of supervisory domain knowledge, such as the kind routinely encountered in NLP. Our contribution may be viewed as a step in this direction.

2 Dual supervision

Most work in supervised learning has focused on learning from examples, each represented by a set of feature values and a class label. In dual supervision we consider an additional aspect, by way of labels of features, which convey prior knowledge on associations of features to particular classes. Since we deal only with text classification in this paper, all features represent term-frequencies of words, and as such we use *feature* and *word* interchangeably.

The active learning schemes we explore in this paper are broadly applicable to any learner that can support dual supervision, but here we focus on active learning for the Pooling Multinomials classifier (Melville et al., 2009) described below. In concurrent related work, we propose active dual supervision schemes for a class of graph-based and kernel-based dual supervision methods (Sindhwani et al., 2009).

2.1 Pooling Multinomials

The Pooling Multinomials classifier was introduced by Melville et al. (2009) as an approach to incorporate prior lexical knowledge into supervised learning for better sentiment detection. In the context of sentiment analysis, lexical knowledge is available in terms of the prior sentiment-polarity of words. From a dual supervision point of view, this knowledge can be seen as labeled features, since the lexicon effectively provides associations of a set of words with the positive or negative class.

Pooling Multinomials classifies unlabeled examples just as in multinomial Naïve Bayes classification (McCallum and Nigam, 1998), by predicting the class with the maximum likelihood, given by $argmax_{c_j} P(c_j) \prod_i P(w_i|c_j)$; where $P(c_j)$ is the prior probability of class c_j, and $P(w_i|c_j)$ is the probability of word w_i appearing in a document of class c_j. In the absence of background knowledge about the class distribution, we estimate the class priors $P(c_j)$ solely from the training data. However, unlike regular Naïve Bayes, the conditional probabilities $P(w_i|c_j)$ are computed using both the labeled examples and the labeled features.

Pooling distributions is a general approach for combining information from multiple sources or experts; where experts are typically represented in terms of probability distributions (Clemen and Winkler, 1999). Here, we only consider the special case of combining multinomial distributions from two sources – namely, the labeled examples and labeled features. The multinomial parameters of such models can be easily combined using the *linear opinion pool* (Clemen and Winkler, 1999), in which the aggregate probability is given by $P(w_i|c_j) = \alpha P_e(w_i|c_j) + (1 - \alpha)P_f(w_i|c_j)$; where $P_e(w_i|c_j)$ and $P_f(w_i|c_j)$ represent the probability assigned by using the example labels and feature labels respectively, and α is the weight for combining these distributions. The weight indicates a level of confidence in each source of information, and Melville et al. (2009) explore ways of automatically selecting this weight. However, in order to not confound our results with the choice of weight-selection mechanism, here we make the simplifying assumption that the two experts based on instance and feature labels are equally valuable, and as such set α to 0.5.

To learn a model from the labeled examples we compute conditionals $P_e(w_i|c_j)$ based on observed term frequencies, as in standard Naïve Bayes classification. In addition, for Pooling Multinomials we need to construct a multinomial model representing the labeled features in the background knowledge. For this, we assume that the feature-class associations provided by labeled features are implicitly arrived at by human experts by examining many positive and negative sentiment documents. So we

attempt to select the parameters $P_f(w_i|c_j)$ of the multinomial distributions that would generate such documents. The exact values of these conditionals are presented below. Their derivation is not directly pertinent to the subject of this paper, but can be found in (Melville et al., 2009).

Given:

$\mathcal{V}$ – the vocabulary, i.e., set of words in our domain
$\mathcal{P}$ – set of words labeled as positive
$\mathcal{N}$ – set of words labeled as negative
$\mathcal{U}$ – set of unknown words, i.e. $\mathcal{V} - (\mathcal{N} \cup \mathcal{P})$
m – size of vocabulary, i.e. $|\mathcal{V}|$
p – number of positive words, i.e. $|\mathcal{P}|$
n – number of negative words, i.e. $|\mathcal{N}|$

All words in the vocabulary can be divided into three categories – words with a positive label, negative label, and unknown label. We refer to the probability of any positive term appearing in a positive document simply as $P_f(w_+|+)$. Similarly, we refer to the probability of any negative term appearing in a negative document as $P_f(w_-|-)$; and the probability of an unknown word in a positive or negative context as $P_f(w_u|+)$ and $P_f(w_u|-)$ respectively. The generative model for labeled features can then be defined by:

$$P_f(w_+|+) = P_f(w_-|-) = \frac{1}{p+n}$$

$$P_f(w_+|-) = P_f(w_-|+) = \frac{1}{p+n} \times \frac{1}{r}$$

$$P_f(w_u|+) = \frac{n(1-1/r)}{(p+n)(m-p-n)}$$

$$P_f(w_u|-) = \frac{p(1-1/r)}{(p+n)(m-p-n)}$$

where, the *polarity level*, r, is a measure of how much more likely it is for a positive term to occur in a positive document compared to a negative term. The value of r is set to 100 in our experiments, as done in (Melville et al., 2009).

2.2 Learning from example vs. feature labels

Dual supervision makes it possible to learn from labeled examples and labeled features simultaneously; and, as in most supervised learning tasks, one would expect more labeled data of either form to lead to more accurate models. In this section we explore the influence of increased number of instance labels and feature labels independently, and also in tandem.

For these, and all subsequent experiments, we use 10-fold cross-validation on the publicly available data of movie reviews provided by Pang et al. (2002). This data consists of 1000 positive and 1000 negative reviews from the Internet Movie Database; where positive labels were assigned to reviews that had a rating above 3.5 stars and negative labels were assigned to ratings of 2 stars and below. We use a bag-of-words representation of reviews, where each review is represented by the term frequencies of the 5000 most frequent words across all reviews, excluding stop-words.

In order to study the effect of increasing number of labels we need to simulate a human oracle labeling data. In the case of examples this is straightforward, since all examples in the *Movies* dataset have labels. However, in the case of features, we do not have a gold-standard set of feature labels. So in order to simulate human responses to queries for feature labels, we construct a *feature oracle* in the following manner. The information gain of words with respect to the known true class labels in the dataset is computed using binary feature representations. Next, out of the 5000 total words, the top 1000 as ranked by information gain are assigned a label. This label is the class in which the word appears more frequently. The oracle returns a "dont know" response for the remaining words. Thus, this oracle simulates a human domain expert who is able to recognize and label the most relevant task-specific words, and also reject a word that falls below the relevance threshold. For instance, in sentiment classification, we would expect a "don't know" response for non-polar words.

We ran experiments beginning with a classifier provided with labels for 10 randomly selected instances and 10 randomly selected features. We then compare three schemes - Instances-then-features, Features-then-instances, and Passive Interleaving. As the name suggests, *Instances-then-features*, is provided labels for randomly selected instances until all instances have been labeled, and then switches to labeling features. Similarly, *Features-then-instances* acquires labels for randomly selected features first and then switches to getting instance labels. In *Passive Interleaving* we probabilistically switch be-

tween issuing queries for randomly chosen instance and feature labels. In particular, at each step we choose to query for an instance with probability 0.36, otherwise we query for a feature label. The instance-query rate of 0.36 is selected based on the ratio of available instances (1800) to available features (5000). The results of these learning curves are presented in Fig. 1. Note that the x-axis in the figure corresponds to the number of queries issued. As discussed earlier, in the case of features, the oracle may respond to a query with a class label or may issue a "don't know" response, indicating that no label is available. As such, the number of feature-queries on the x-axis does not correspond to the number of actual known feature labels. We would expect that on average 1 in 5 feature-label queries prompts a response from the feature oracle that results in a known feature label being provided.

At the end of the learning curves, each method has labels for all available instances and features; and as such, the last points of all three curves are identical. The results show that fixing the number of labeled features, and increasing the number of labeled instances steadily improves classification accuracy. This is what one would expect from traditional supervised learning curves. More interestingly, the results also indicate that we can fix the number of instances, and improve accuracy by labeling more features. Finally, results on Passive Interleaving show that, though both feature labels and example labels are beneficial by themselves, dual supervision which exploits the interaction of examples and features does in fact benefit from acquiring both types of labels concurrently.

For all results above, we are selecting instances and/or features to be labeled uniformly at random. Based on previous work in active learning one would expect that we can select instances to be labeled more efficiently, by having the learner decide which instances it is most likely to benefit from. The results in this section suggests that actively selecting features to be labeled may also be beneficial. Furthermore, the Passive Interleaving results suggest that an ideal active dual supervision scheme would actively select both instances and features for labeling. We begin by exploring active learning for feature labels in the next section, and then consider the simultaneous selection of instances and features in Sec. 4.

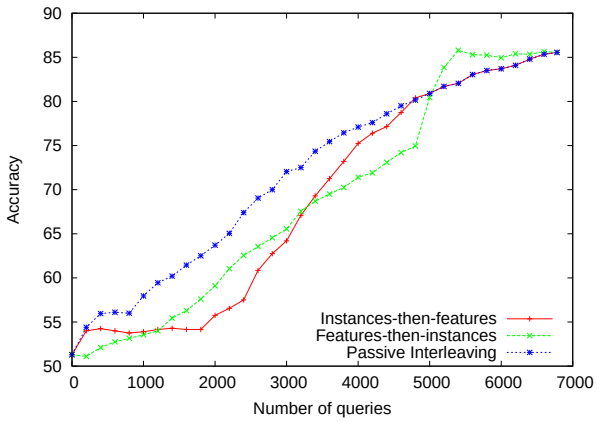

Figure 1: Comparing the effect of instance and feature label acquisition in dual supervision.

3 Acquiring feature labels

Traditional active learning has primarily focused on selecting unlabeled *instances* to be labeled. The dual-supervision setting now provides us with an additional dimension to active learning, where labels may also be acquired for features. In this section we look at the novel task of active learning applied only to feature-label acquisition. In Section 4 we study the more general task of active dual supervision, where both instance and feature labels may be acquired concurrently.

3.1 Feature uncertainty vs. certainty

A very common approach to active learning for instances is Uncertainty Sampling (Lewis and Catlett, 1994). In this approach we acquire labels for instances that the current model is most uncertain about. Uncertainty Sampling is founded on the heuristic that uncertain instances are close to the current classification boundary, and acquiring the correct labels for them are likely to help refine the location of this boundary. Despite its simplicity, Uncertainty Sampling is usually quite effective in practice; which raises the question of whether one can apply the same principle to feature-label acquisition. In this case, we want to select unlabeled features that the current model is most uncertain about.

Much like instance uncertainty, feature uncertainty can be measured in different ways, depending on the underlying method used for dual supervision. For instance, if the learner produces a lin-

ear classifier as in (Sindhwani and Melville, 2008), we could use the magnitude of the weights on the features as a measure of uncertainty – where lower weights indicate less certainty. Since Pooling Multinomials builds a multinomial Naïve Bayes model, we can directly use the model's conditional probabilities of each word (feature) given a class.

For ease of exposition we refer to the two classes in binary classification as *postive* (+) and *negative* (-), without loss of generality. Given the probabilities of word f belonging to the positive and negative class, $P(f|+)$ and $P(f|-)$, we can determine the uncertainty of a feature using the absolute value of the log-odds ratio, i.e.,

$$abs\left(\log\left(\frac{P(f|+)}{P(f|-)}\right)\right) \quad (1)$$

The smaller this value, the more uncertain the model is about the feature's class association. In every iteration of active learning we can select the features with the lowest certainty scores. We refer to this approach as *Feature Uncertainty*.

Though Uncertainty Sampling for features seems like an appealing notion, it may not lead to better models. If a classifier is uncertain about a feature, it may have insufficient information about this feature and may indeed benefit from learning its label. However, it is also quite likely that a feature has a low certainty score because it does not carry much discriminative information about the classes. In the context of sentiment detection, one would expect that neutral/non-polar words will appear to be uncertain words. For example, words such as "the" which are unlikely to help in discriminating between classes, are also likely to be considered the most uncertain. As we shortly report, on the movies dataset, Feature Uncertainty ends up wasting queries on such words ending up with performance inferior to random feature queries. What works significantly better is an alternative strategy which acquires labels for features in the descending order of the score in Eq 1. We refer to this approach as *Feature Certainty*.

3.2 Expected feature utility

The intuition underlying the feature certainty heuristic is that it serves to confirm or correct the orientation of model probabilities on different words during the active learning process. One can argue that feature certainty is also suboptimal in that queries may be wasted simply confirming confident predictions, which is of limited utility to the model. An alternative to using a certainty-based heuristic, is to directly estimate the expected value of acquiring each feature label. Such Expected Utility (Estimated Risk Minimization) approaches have been applied successfully to traditional active learning (Roy and McCallum, 2001), and to active feature-value acquisition (Melville et al., 2005). In this section we describe how this Expected Utility framework can be adapted for feature-label acquisition.

At every step of active learning for features, the next best feature to label is one that will result in the highest improvement in classifier performance. Since the true label of the unlabeled features are unknown prior to acquisition, it is necessary to estimate the potential impact of every feature query for all possible outcomes.[1] Hence, the decision-theoretic optimal policy is to ask for feature labels which, once incorporated into the data, will result in the highest increase in classification performance in *expectation*.

If f_j is the label of the j-th feature, and q_j is the query for this feature's label, then the Expected Utility of a feature query q_j can be computed as:

$$EU(q_j) = \sum_{k=1}^{K} P(f_j = c_k)\mathcal{U}(f_j = c_k) \quad (2)$$

Where $P(f_j = c_k)$ is the probability that f_j will be labeled with class c_k, and $\mathcal{U}(f_j = c_k)$ is the utility to the model of knowing that f_j has the label c_k. In practice, the true values of these two quantities are unknown, and the main challenge of any Expected Utility approach is to accurately estimate these quantities from the data currently available.

A direct way to estimate the utility of a feature label to classification, is to measure classification accuracy on the training set of a model built using this feature label. However, small changes in the model that may result from a acquiring a single additional feature label may not be reflected by a change in accuracy. As such, we use a more fine-grained measure of classifier performance, Log Gain, which is

[1] In the case of binary polarity classification, the possible outcomes are a *positive* or *negative* label for a queried feature.

computed as follows. For a model induced from a training set T, let $\hat{P}(c_k|x_i)$ be the probability estimated by the model that instance x_i belongs to class c_k; and $\mathbb{I}$ is an indicator function such that $\mathbb{I}(c_k, x_i) = 1$ if c_k is the correct class for x_i and $\mathbb{I}(c_k, x_i) = 0$, otherwise. Log Gain is then defined as:

$$LG(x_i) = -\sum_{k=1}^{K} \mathbb{I}(c_k)\hat{P}(c_k|x_i) \quad (3)$$

Then the utility of a classifier, $\mathcal{U}$, can be measured by summing the Log Gain for all instances in the training set T. A lower value of Log Gain indicates a better classifier performance. For a deeper discussion of this measure see (Saar-Tsechansky et al., 2008).

In Eq. 2, apart from the measure of utility, we also do not know the true probability distribution of labels for the feature under consideration. This too can be estimated from the training data, by seeing how frequently the word appears in documents of each class. In a multinomial Naïve Bayes model we already collect these statistics in order to determine the conditional probability of a class given a word, i.e. $P(f_j|c_k)$. We can use these probabilities to get an estimate of the feature label distribution, $\hat{P}(f_j = c_k) = \frac{P(f_j|c_k)}{\sum_{k=1}^{K} P(f_j|c_k)}$.

Given the estimated values of the feature-label distribution and the utility of a particular feature query outcome, we can now estimate the Expected Utility of each unknown feature, and select the features with the highest Expected Utility to modeling.

Though theoretically appealing, this approach is quite computationally intensive if applied to evaluate all unknown features. In the worst case it requires building and evaluating models for each possible outcome of each unlabeled feature query. If you have m features and K classes, this approach requires training $O(mK)$ classifiers. However, the complexity of the approach can be significantly alleviated by only applying Expected Utility evaluation to a sub-sample of all unlabeled features. Given the large number of features with no true class labels, selecting a sample of available features uniformly at random may be sub-optimal. Instead we select a sample of features based on Feature Certainty. In particular we select the top 100 unknown features that the current model is most certain about, and identify the features in this pool with the highest Expected Utility. We refer to this approach as *Expected Feature Utility*. We use Feature Certainty to sub-sample the available feature queries, since this approach is more likely to select features for which the label is known by the Oracle.

3.3 Active learning with feature labels

We ran experiments comparing the three different active learning approaches described above. In these, and all subsequent experiments, we begin with a model trained on 10 labeled features and 100 labeled instances, which were randomly selected. From our prior efforts of manually labeling such data, we find this to be a reasonable initial setting.

The experiments in this section focus only on the selection of *features* to be labeled. So, in each iteration of active learning we select the next 10 feature-label queries, based on Feature Uncertainty, Feature Certainty, or Expected Feature Utility. As a baseline, we also compare to the performance of a model that selects features uniformly at random. Our results are presented in Fig. 2.

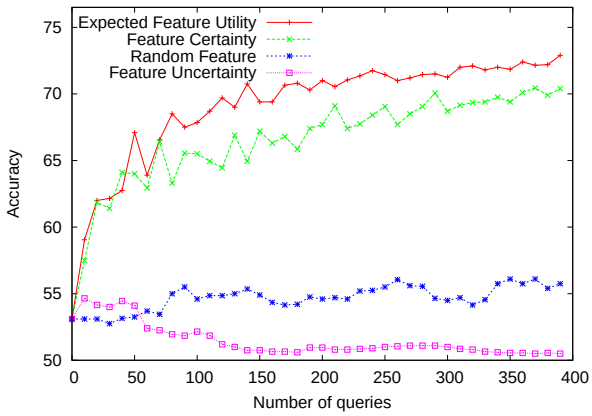

Figure 2: Comparing different active learning approaches for acquiring feature labels.

The results show that Feature Uncertainty, which is a direct analog of Uncertainty Sampling, actually performs worse than random sampling. Many uncertain features may actually not be very useful in discriminating between the classes, and selecting them can be systematically worse than selecting uniformly at random. However, the converse approach

of Feature Certainty does remarkably well. This may be because polarized words are better for learning, but it is also likely that querying for such words increases the likelihood of selecting one whose label is known to the oracle.

The results on Expected Feature Utility show that estimating the expected impact of potential labels for features does in fact perform much better than feature certainty. The results confirm that despite our crude estimations in Eq. 2, Expected Feature Utility is an effective approach to active learning of feature labels. Furthermore, we demonstrate that by applying the approach to only a small sub-sample of certain features, we are able to make this method computationally feasible to use in practice. Increasing the size of the sample of candidate feature queries is likely to improve performance, at the cost of increased time in selecting queries.

4 Active dual supervision

In the previous section we demonstrated that actively selecting informative features to be labeled is significantly better than random selection. In this section, we look at the complementary task of selecting instances to be labeled, and combined active learning for both forms of supervision.

Selecting unlabeled examples for learning has been a well-studied problem, and we use Uncertainty Sampling (Lewis and Catlett, 1994), which has been shown to be a computationally efficient and effective approach in the literature. In particular we select unlabeled examples to be labeled in order of decreasing uncertainty, where uncertainty is measured in terms of the margin, as done in (Melville and Mooney, 2004). The margin on an unlabeled example is defined as the absolute difference between the class probabilities predicted by the classifier for the given example, i.e., $|P(+|x) - P(-|x)|$. We refer to the selection of instances based on this uncertainty as Instance Uncertainty, in order to distinguish it from Feature Uncertainty.

We ran experiments as before, comparing selection of instances using Instance Uncertainty and selection of features using Expected Feature Utility. In addition, we also combine these to methods by interleaving feature and instance selection. In particular, we first order instances in decreasing order of uncertainty, and features in terms of decreasing Expected Feature Utility. We then probabilistically select instances or features from the top of these lists, where, as before, the probability of selecting an instance is 0.36. Recall that this probability corresponds to the ratio of available instances (1800) and features (5000). We refer to this approach as Active Interleaving, in contrast to Passive Interleaving, which we also present as a baseline. Recall that Passive Interleaving corresponds to probabilistically interleaving queries for randomly chosen, not actively chosen, examples and features. Our results are presented in Fig. 3.

We observe that, Instance Uncertainty performs better than Passive Interleaving, which in turn is better than random selection of only instances or features – as seen in Fig. 1. However, effectively selecting features labels, via Expected Feature Utility, does even better than actively selecting only instances. Finally, selecting instance and features simultaneously via Active Interleaving performs better than active learning of features or instances separately. Active Interleaving is indeed very effective, reaching an accuracy of 77% with only 500 queries, while Passive Interleaving requires more than 4000 queries to reach the same performance – as evidenced by Fig. 1

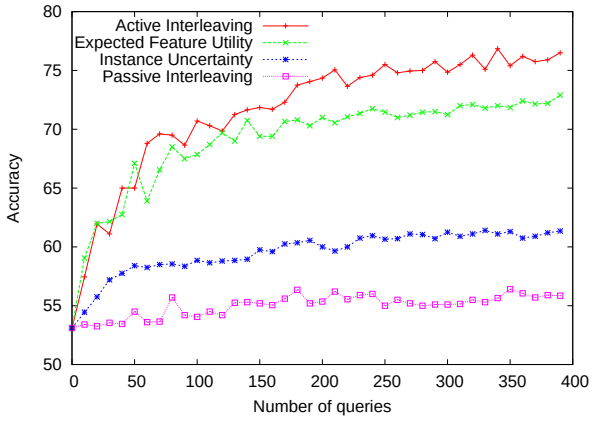

Figure 3: Comparing Active Interleaving to alternative label acquisition strategies.

5 Related work

Active learning in the context of dual supervision models is a new area of research with very little prior

work, to the best of our knowledge. Most prior work has focused on pooled-based active learning, where examples from an unlabeled pool are selected for labeling (Cohn et al., 1994; Tong and Koller, 2000). In contrast, active feature-value acquisition (Melville et al., 2005) and *budgeted learning* (Lizotte et al., 2003) focus on estimating the value of acquiring missing features, but do not deal with the task of learning from feature *labels*. In contrast, Raghavan and Allan (2007) and Raghavan et al. (2006) study the problem of *tandem learning* where they combine uncertainty sampling for instances along with co-occurence based interactive feature selection. Godbole et al. (2004) propose notions of feature uncertainty and incorporate the acquired feature labels, into learning by creating one-term mini-documents.

Learning from labeled examples and features via dual supervision, is itself a new area of research. Sindhwani et al. (2008) use a kernel-based framework to build dual supervision into co-clustering models. Sindhwani and Melville (2008) apply similar ideas for graph-based sentiment analysis. There have also been previous attempts at using only feature supervision, mostly along with unlabeled documents. Much of this work (Schapire et al., 2002; Wu and Srihari, 2004; Liu et al., 2004; Dayanik et al., 2006) has focused on using labeled features to generate *pseudo-labeled examples* that are then used with well-known models. In contrast, Druck et al. (2008) constrain the outputs of a multinomial logistic regression model to match certain reference distributions associated with labeled features.

6 Perspectives and future work

Though Active Interleaving is a very effective approach to active dual supervision, there is still a lot of room for improvement. Firstly, Active Interleaving relies on Uncertainty Sampling for the selection of instances. Though Uncertainty Sampling has the advantage of being fast and effective, there exist approaches that lead to better models with fewer examples – usually at the cost of computation time. One such method, estimating error reduction (Roy and McCallum, 2001), is a direct analog of Expected Feature Utility applied to instance selection. One would expect that an improvement in instance selection, should directly improve any method that combines instance and feature label selection. Secondly, Active Interleaving uses the simple approach of probabilistically choosing to select an instance or feature for each subsequent query. However, a more intelligent active scheme should be able to assess if an instance or feature would be more beneficial at each step. Furthermore, we do not currently consider the cost of acquiring labels. Presumably labeling a feature versus labeling an instance could incur very different costs – which could be monetary costs or time taken for each annotation. Fortunately, the Expected Utility method is very flexible, and allows us to address all these issues within a single framework. We can specifically estimate the expected utility of different forms of annotation, per unit cost. For instance, Provost et al. (2007) use such an approach to estimate the utility of acquiring class labels and feature values (not labels) per unit cost, within one unified framework. A similar method can be applied for a holistic approach to active dual supervision, where the Expected Utility of an instance or feature label query q, can be computed as $EU(q) = \sum_{k=1}^{K} P(q = c_k) \frac{\mathcal{U}(q=c_k)}{\omega_q}$; where ω_q is cost of the query q, and utility $\mathcal{U}$ can be computed as in Eq. 3. By evaluating instances and features on the same scale, and by measuring utility per unit cost of acquisition, such a framework should enable us to handle the trade-off between the costs and benefits of the different types of acquisitions. The primary challenge in the success of this approach is to *accurately* and *efficiently* estimate the different quantities in the equation above, using only the training data currently available. These are directions for future exploration.

7 Conclusions

This paper is a preliminary foray into active dual supervision. We have demonstrated that not only is combining example and feature labels beneficial for modeling, but that actively selecting the most informative examples and features for labeling can significantly reduce the burden of annotating such data. In future work, we would like to explore more effective solutions to the problem, and also to corroborate our results on a larger number of datasets and under different experimental settings.

References

R. T. Clemen and R. L. Winkler. 1999. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19:187–203.

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

Aynur Dayanik, David D. Lewis, David Madigan, Vladimir Menkov, and Alexander Genkin. 2006. Constructing informative prior distributions from domain knowledge in text classification. In *SIGIR*.

G. Druck, G. Mann, and A. McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*.

S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. 2004. Document classification through interactive supervision of document and term labels. In *PKDD*.

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of 11th Intl. Conf. on Machine Learning (ICML-94)*, pages 148–156, San Francisco, CA, July. Morgan Kaufmann.

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip Yu. 2004. Text classification by labeling words. In *AAAI*.

Dan Lizotte, Omid Madani, and Russell Greiner. 2003. Budgeted learning of naive-Bayes classifiers. In *UAI*.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Text Categorization*.

Prem Melville and Raymond J. Mooney. 2004. Diverse ensembles for active learning. In *Proc. of 21st Intl. Conf. on Machine Learning (ICML-2004)*, pages 584–591, Banff, Canada, July.

Prem Melville, Maytal Saar-Tsechansky, Foster Provost, and Raymond Mooney. 2005. An expected utility approach to active feature-value acquisition. In *ICDM*.

Prem Melville, Wojciech Gryc, and Richard Lawrence. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *KDD*.

Bo Pang and Lilian Lee. 2008. *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval: Vol. 2: No 1, pp 1-135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP*.

Foster Provost, Prem Melville, and Maytal Saar-Tsechansky. 2007. Data acquisition and cost-effective predictive modeling: Targeting offers for electronic commerce. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*.

Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *J. Mach. Learn. Res.*, 7:1655–1686.

H. Raghavan, O. Madani, and R. Jones. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *ICML*.

Maytal Saar-Tsechansky, Prem Melville, and Foster Provost. 2008. Active feature-value acquisition. In *Management Science*.

Robert E. Schapire, Marie Rochery, Mazin G. Rahim, and Narendra Gupta. 2002. Incorporating prior knowledge into boosting. In *ICML*.

Vikas Sindhwani and Prem Melville. 2008. Document-word co-regularization for semi-supervised sentiment analysis. In *ICDM*.

Vikas Sindhwani, Jianying Hu, and Alexandra Mojsilovic. 2008. Regularized co-clustering with dual supervision. In *NIPS*.

Vikas Sindhwani, Prem Melville, and Richard Lawrence. 2009. Uncertainty sampling and transductive experimental design for active dual supervision. In *ICML*.

Simon Tong and Daphne Koller. 2000. Support vector machine active learning with applications to text classification. In *Proc. of 17th Intl. Conf. on Machine Learning (ICML-2000)*.

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

O. F. Zaidan and J. Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *EMNLP*.