# Assessing Benefit from Feature Feedback in Active Learning for Text Classification

**Shilpa Arora**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
shilpaa@cs.cmu.edu

**Eric Nyberg**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
ehn@cs.cmu.edu

## Abstract

Feature feedback is an alternative to instance labeling when seeking supervision from human experts. Combination of instance and feature feedback has been shown to reduce the total annotation cost for supervised learning. However, learning problems may not benefit equally from feature feedback. It is well understood that the benefit from feature feedback reduces as the amount of training data increases. We show that other characteristics such as domain, instance granularity, feature space, instance selection strategy and proportion of relevant text, have a significant effect on benefit from feature feedback. We estimate the maximum benefit feature feedback may provide; our estimate does not depend on how the feedback is solicited and incorporated into the model. We extend the complexity measures proposed in the literature and propose some new ones to categorize learning problems, and find that they are strong indicators of the benefit from feature feedback.

## 1 Introduction

Linear classifiers model the response as a weighted linear combination of the features in input instances. A supervised approach to learning a linear classifier involves learning the weights for the features from labeled data. A large number of labeled instances may be needed to determine the class association of the features and learn accurate weights for them. Alternatively, the user may directly label the features. For example, for a sentiment classification task, the user may label features, such as words or phrases, as expressing positive or negative sentiment. Prior work (Raghavan et al., 2006; Zaidan et al., 2007) has demonstrated that users are able to reliably provide useful feedback on features.

*Direct feedback* on a list of features (Raghavan et al., 2006; Druck et al., 2008) is limited to simple features like unigrams. However, unigrams are limited in the linguistic phenomena they can capture. Structured features such as dependency relations, paths in syntactic parse trees, etc., are often needed for learning the target concept (Pradhan et al., 2004; Joshi and Rosé, 2009). It is not clear how direct feature feedback can be extended straightforwardly to structured features, as they are difficult to present visually for feedback and may require special expertise to comprehend. An alternative approach is to seek *indirect feedback* on structured features (Arora and Nyberg, 2009) by asking the user to highlight spans of text, called *rationales*, that support the instance label (Zaidan et al., 2007). For example, when classifying the sentiment of a movie review, rationales are spans of text in the review that support the sentiment label for the review.

Assuming a fixed cost per unit of work, it might be cheaper to ask the user to label a few features, i.e. identify relevant features and their class association, than to label several instances. Prior work (Raghavan et al., 2006; Druck et al., 2008; Druck et al., 2009; Zaidan et al., 2007) has shown that a combination of instance and feature labeling can be used to reduce the total annotation cost required to learn the target concept. However, the benefit from feature feedback may vary across learning problems. If we can estimate the benefit from feature feedback for a



*Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 106–114,
Portland, Oregon, USA, 23–24 June 2011. 

given problem, we can minimize the total annotation cost for achieving the desired performance by selecting the optimal annotation strategy (feature feedback or not) at every stage in learning. In this paper, we present the ground work for this research problem by analyzing how benefit from feature feedback varies across different learning problems and what characteristics of a learning problem have a significant effect on benefit from feature feedback.

We define a *learning problem* ($P = \{D, G, F, L, I, S\}$) as a tuple of the domain ($D$), instance granularity ($G$), feature representation ($F$), labeled data units ($L$), amount of irrelevant text ($I$) and instance selection strategy ($S$).

With enough labeled data, we may not benefit from feature feedback. Benefit from feature feedback also depends on the *features* used to represent the instances. If the feature space is large, we may need several labeled instances to identify the relevant features, while relatively fewer labeled features may help us quickly find these relevant features. Apart from the feature space size, it also matters what types of features are used. When hand crafted features from a domain expert are used (Pradhan et al., 2004) we expect to gain less from feature feedback as most of the features will be relevant. On the other hand, when features are extracted automatically as patterns in annotation graphs (Arora et al., 2010) feature feedback can help to identify relevant features from the large feature space.

In active learning, instances to be labeled are selectively sampled in each iteration. Benefit from feature feedback will depend on the instances that were used to train the model in each iteration. In the case of indirect feature feedback through rationales or direct feature feedback in context, instances selected will also determine what features receive feedback. Hence, *instance selection strategy* should affect the benefit from feature feedback.

In text classification, an instance may contain a large amount of text, and even a simple unigram representation will generate a lot of features. Often only a part of the text is relevant for the classification task. For example, in movie reviews, often the reviewers talk about the plot and characters in addition to providing their opinion about the movie. Often this extra information is not relevant to the classification task and bloats the feature space without adding many useful features. With feature feedback, we hope to filter out some of this noise and improve the model. Thus, the *amount of irrelevant information* in the instance should play an important role in determining the benefit from feature feedback. We expect to see less of such noise when the text instance is more concise. For example, a movie review snippet (about a sentence length) tends to have less irrelevant text than a full movie review (several sentences). In addition to analyzing document instances with varying amount of noise, we also compare the benefit from feature feedback for problems with different *granularity*. Granularity for a learning problem is defined based on the average amount of text in its instances.

Benefit from feature feedback will also depend on how feedback is solicited from the user and how it is incorporated back into the model. Independently from these factors, we estimate the maximum possible benefit and analyze how it varies across problems. Next we describe measures proposed in the literature and propose some new ones for categorizing learning problems. We then discuss our experimental setup and analysis.

## 2 Related Work

There has been little work on categorizing learning problems and how benefit from feature feedback varies with them. To the best of our knowledge there is only one work in this area by Raghavan et al. (2007). They categorize problems in terms of their *feature complexity*. Feature complexity is defined in terms of the minimum number of features required to learn a good classifier (close to maximum performance). If the concept can be described by a weighted combination of a few well-selected features, it is considered to be of low complexity.

In this estimate of complexity, an assumption is made that the best performance is achieved when the learner has access to all available features and not for any subset of the features. This is a reasonable assumption for text classification problems with robust learners like SVMs together with appropriate regularization and sufficient training data.

Instead of evaluating all possible combinations of features to determine the minimum number of features required to achieve close to the best perfor-

mance, feature complexity is estimated using an intelligent ranking of the features. This ranking is based on their discriminative ability determined using a large amount of labeled data (referred to as *oracle*) and a feature selection criterion such as Information Gain (Rijsbergen, 1979). It is intuitive that the rate of learning, i.e., the rate at which performance improves as we add more features to the model, is also associated with problem complexity. Raghavan et al. (2007) define the *feature learning convergence profile* ($p_{fl}$) as the area under the feature learning curve (performance vs. number of features used in training), given by:

$$p_{fl} = \frac{\sum_{t=1}^{log_2 N} F1(M, 2^t)}{log_2 N \times F1(M, N)} \quad (1)$$

where $F1(M, 2^t)$ is the F1 score on the test data when using all $M$ instances for training with top ranked $2^t$ features. The features are added at an exponentially increasing interval to emphasize the relative increase in feature space size. The three feature complexity measures proposed by Raghavan et al. (2007) are the following: 1) *Feature size complexity ($N_f$)*: Logarithm (base 2) of the number of features needed to achieve $95\%$ of the best performance (when all instances are available), 2) *Feature profile complexity ($F_{pc}$)*, given by $F_{pc} = 1 - p_{fl}$, and 3) *Combined feature complexity ($C_f$)* , $C_f = F_{pc} * n_f$, incorporates both the learning profile and the number of features required.

In order to evaluate the benefit from feature feedback, Raghavan et al. (2007) use their tandem learning approach of interleaving instance and feature feedback (Raghavan et al., 2006), referred to as interactive feature selection ($ifs$). The features are labeled as 'relevant' (feature discriminates well among the classes), or 'non-relevant/don't know'. The labeled features are incorporated into learning by scaling the value of the relevant features by a constant factor in all instances.

Raghavan et al. (2007) measure the benefit from feature feedback as the gain in the learning speed with feature feedback. The learning speed measures the rate of performance improvement with increasing amount of supervision. It is defined in terms of the convergence profile similar to feature learning convergence profile in Equation 1, except in terms of the number of labeled units instead of the number of features. A labeled unit is either a labeled instance or an equivalent set of labeled features with the same annotation time. The benefit from feature feedback is then measured as the difference in the convergence profile with interactive feature selection ($p_{ifs}$) and with labeled instances only ($p_{al}$).

Raghavan et al. (2007) analysed $9$ corpora and $358$ binary classification tasks. Most of these corpora, such as Reuters (Lewis, 1995), 20-newsgroup (Lang, 1995), etc., have topic-based category labels. For all classification tasks, they used simple and fixed feature space containing only unigram features (n-gram features were added where it seemed to improve performance). They observed a negative correlation ($r = -0.65$) between the benefit from feature feedback and combined feature complexity ($C_f$), i.e., feature feedback accelerates active learning by an amount that is inversely proportional to the feature complexity of the problem. If a concept can be expressed using a few well-selected features from a large feature space, we stand to benefit from feature feedback as few labeled features can provide this information. On the other hand, if learning a concept requires all or most of the features in the feature space, there is little knowledge that feature feedback can provide.

## 3 Estimating Maximum Benefit & Additional Measures

In this section, we highlight some limitations of the prior work that we address in this work.

Raghavan et al. (2007) only varied the domain among different problems they analyzed, i.e, only the variable $D$ in our problem definition ($P = \{D, G, F, L, I, S\}$). However, as motivated in the introduction, other characteristics are also important when categorizing learning problems and it is not clear if we will observe similar results on problems that differ in these additional characteristics. In this work, we apply their measures to problems that differ in these characteristics in addition to the domain.

Analysis in Raghavan et al. (2007) is specific to their approach for incorporating feature feedback into the model, which may not work well for all domains and datasets as also mentioned in their work (Section 6.1). It is not clear how their results can be

extended to alternate approaches for seeking and incorporating feature feedback. Thus, in this work we analyze the maximum benefit a given problem can get from feature feedback independent of the feedback solicitation and incorporation approach.

Raghavan et al. (2007) analyze benefit from feature feedback at a fixed training data size of 42 labeled units. However, the difference between learning problems may vary with the amount of labeled data. Some problems may benefit significantly from feature feedback even at relatively larger amount of labeled data. On the other hand, with very large training set, the benefit from feature feedback can be expected to be small and not significant for all problems and all problems will look similar. Thus, we evaluate the benefit from feature feedback at different amount of labeled data.

Raghavan et al. (2007) evaluate benefit from feature feedback in terms of the gain in learning speed. However, the learning rate does not tell us how much improvement we get in performance at a given stage in learning. In fact, even if at every point in the learning curve performance with feature feedback was lower than performance without feature feedback, the rate of convergence to the corresponding maximum performance may still be higher when using feature feedback. Thus, in this work, in addition to evaluating the improvement in the learning speed, we also evaluate the improvement in the absolute performance at a given stage in learning.

### 3.1 Determining the Maximum Benefit

Annotating instances with or without feature feedback may require different annotation time. It is only fair to compare different annotation strategies at same annotation cost. Raghavan et al. (2006) found that on average labeling an instance takes the same amount of time as direct feedback on 5 features. Zaidan et al. (2007) found that on average it takes twice as much time to annotate an instance with rationales than to annotate one without rationales. In our analysis, we focus on feedback on features in context of the instance they occur in, i.e., indirect feature feedback through rationales or direct feedback on features that occur in the instance being labeled. Thus, based on the findings in Zaidan et al. (2007), we assume that on average annotating an instance with feature feedback takes twice as much time as annotating an instance without feature feedback. We define a currency for annotation cost as *Annotation cost Units (AUs)*. For an annotation budget of $a$ AUs, we compare two annotation strategies of annotating $a$ instances without feature feedback or $\frac{a}{2}$ instances with feature feedback.

In this work, we only focus on using feature feedback as an alternative to labeled data, i.e., to provide evidence about features in terms of their relevance and class association. Thus, the best feature feedback can do is provide as much evidence about features as evidence from a large amount of labeled data (oracle). Let $F1(k, N_m)$ be the F1 score of a model trained with features that occur in $m$ training instances ($N_m$) and evidence for these features from $k$ instances ($k \geq m$). For an annotation budget of $a$ AUs, we define the maximum improvement in performance with feature feedback ($IP_a$) as the difference in performance with feature feedback from oracle on $\frac{a}{2}$ training instances and performance with $a$ training instances without feature feedback.

$$IP_a = F1(o, N_{\frac{a}{2}}) - F1(a, N_a) \tag{2}$$

where $o$ is the number of instances in the oracle dataset ($o >> a$). We also compare annotation strategies in terms of the learning rate similar to Raghavan et al. (2007), except that we estimate and compare the maximum improvement in the learning rate. For an annotation budget of $a$ AUs, we define the maximum improvement in learning rate from 0 to $a$ AUs ($ILR_{0-a}$) as follows.

$$ILR_{0-a} = {p_{cp}}^{wFF} - {p_{cp}}^{woFF} \tag{3}$$

where ${p_{cp}}^{wFF}$ and ${p_{cp}}^{woFF}$ are the convergence profiles with and without feature feedback at same annotation cost, calculated as follows.

$${p_{cp}}^{wFF} = \frac{\sum_{t=1}^{log_2 \frac{a}{2}} F1(o, N_{2^t})}{log_2 \frac{a}{2} \times F1(o, N_{\frac{a}{2}})} \tag{4}$$

$${p_{cp}}^{woFF} = \frac{\sum_{t=2}^{log_2 a} F1(2^t, N_{2^t})}{(log_2 a - 1) \times F1(a, N_a)} \tag{5}$$

where $2^t$ denotes the training data size in iteration $t$. Like Raghavan et al. (2007), we use exponentially increasing intervals to emphasize the relative increase in the training data size, since adding a few

labeled instances earlier in learning will give us significantly more improvement in performance than adding the same number of instances later on.

### 3.2 Additional Metrics

The feature complexity measures require an 'oracle', simulated using a large amount of labeled data, which is often not available. Thus, we need measures that do not require an oracle.

Benefit from feature feedback will depend on the uncertainty of the model on its predictions, since it suggests uncertainty on the features and hence scope for benefit from feature feedback. We use the probability of the predicted label from the model as an estimate of the model's uncertainty. We evaluate how benefit from feature feedback varies with summary statistics such as mean, median and maximum probability from the model on labels for instances in a held out dataset.

## 4 Experiments, Results and Observations

In this section, we describe the details of our experimental setup followed by the results.

### 4.1 Data

We analyzed three datasets: 1) Movie reviews with rationale annotations by Zaidan et al. (2007), where the task is to classify the sentiment (positive/negative) of a review, 2) Movie review snippets from Rotten Tomatoes (Pang and Lee., 2005), and 3) WebKB dataset with the task of classifying whether or not a webpage is a faculty member's homepage. Raghavan et al. (2007) found that the webpage classification task has low feature complexity and benefited the most from feature feedback. We compare our results on this task and the sentiment classification task on the movie review datasets.

### 4.2 Experimental Setup

Table 1 describes the different variables and their possible values in our experiments. We make a logical distinction for granularity based on whether an instance in the problem is a document (several sentences) or a sentence. Labeled data is composed of instances and their class labels with or without feature feedback. As discussed in Section 3.1, instances with feature feedback take on average twice as much time to annotate as instances without feature feedback. Thus, we measure the labeled data in terms of the number of annotation cost units which may mean different number of labeled instances based on the annotation strategy. We used two feature configurations of "unigram only" and "unigram+dependency triples". The unigram and dependency annotations are derived from the Stanford Dependency Parser (Klein and Manning, 2003).

Rationales by definition are spans of text in a review that convey the sentiment of the reviewer and hence are the part of the document most relevant for the classification task. In order to vary the amount of irrelevant text, we vary the amount of text (measured in terms of the number of characters) around the rationales that is included in the instance representation. We call this the *slack* around rationales. When using the rationales with or without the slack, only features that overlap with the rationales (and the slack, if used) are used to represent the instance. Since we only have rationales for the movie review documents, we only studied the effect of varying the amount of irrelevant text on this dataset.

| **Variable** | **Possible Values** |
|---|---|
| Domain ($D$) | {Movie Review classification (MR), Webpage classification (WebKB)} |
| Instance Granularity ($G$) | {document (doc), sentence (sent)} |
| Feature Space ($F$) | {unigram only (u), unigram+dependency (u+d)} |
| Labeled Data (#AUs) ($L$) | {64, 128, 256, 512, 1024} |
| Irrelevant Text ($I$) | {0, 200, 400, 600, $\infty$ } |
| Instance Selection Strategy ($S$)) | {deterministic (deter), uncertainty (uncert)} |

Table 1: Experiment space for analysis of learning problems ($P = \{D, G, F, L, I, S\}$)

For all our experiments, we used Support Vector Machines (SVMs) with linear kernel for learning (libSVM (Chang and Lin, 2001) in Minorthird (Cohen, 2004)). For identifying the discriminative features we used the information gain score. For all datasets we used 1800 total examples with equal number of positive and negative examples. We

held out 10% of the data for estimating model's uncertainty as explained in Section 3.2. The results we present are averaged over 10 cross validation folds on the remaining 90% of the data (1620 instances). In a cross validation fold, 10% data is used for testing (162 instances) and all of the remaining 1458 instances are used as the 'oracle' for calculating the feature complexity measures and estimating the maximum benefit from feature feedback as discussed in Sections 2 and 3.1 respectively. The training data size is varied from 64 to 1024 instances (from the total of 1458 instances for training in a fold), based on the annotation cost budget. Instances with their label are added to the training set either in the original order they existed in the dataset, i.e. no selective sampling (deterministic), or in the decreasing order of current model's uncertainty on them. Uncertainty sampling in SVMs (Tong and Koller, 2000) selects the instances closest to the decision boundary since the model is expected to be most uncertain about these instances. In each slice of the data, we ensured that there is equal distribution of the positive and negative class. SVMs do not yield probabilistic output but a decision boundary, a common practice is to fit the decision values from SVMs to a sigmoid curve to estimate the probability of the predicted class (Platt, 1999).

### 4.3 Results and Analysis

To determine the effect of various factors on benefit from feature feedback, we did an ANOVA analysis with Generalized Linear Model using a 95% confidence interval. The top part of Table 2 shows the average $F1$ score for the two annotation strategies at same annotation cost. As can be seen, with feature feedback, we get a significant improvement in performance.

Next we analyze the significance of the effect of various problem characteristics discussed above on benefit from feature feedback in terms of improvement in performance ($IP$) at given annotation cost and improvement in learning rate ($ILR$). Improvement in learning rate is calculated by comparing the learning profile for the two annotation strategies with increasing amount of labeled data, up to the maximum annotation cost of 1024 $AUs$.

As can be seen from the second part of Table 2, most of the factors have a significant effect on benefit from feature feedback. The benefit is significantly higher for the webpage classification task than the sentiment classification task in the movie review domain. We found that average feature complexity for the webpage classification task ($N_f = 3.07$) to be lower than average feature complexity for the sentiment classification task ($N_f = 5.18$) for 1024 training examples. Lower feature complexity suggests that the webpage classification concept can be expressed with few keywords such as *professor*, *faculty*, etc., and with feature feedback we can quickly identify these features. Sentiment on the other hand can be expressed in a variety of ways which explains the high feature complexity.

The benefit is more for document granularity than sentence granularity, which is intuitive as feature space is substantially larger for documents and we expect to gain more from the user's feedback on which features are important. This difference is significant for improvement in the learning rate and marginally significant for improvement in performance. Note that here we are comparing documents (with or without rationale slack) and sentences. However, documents with low rationale slack should have similar amount of noise as a sentence. Also, a significant difference between domains suggests that documents in WebKB domain might be quite different from those in Movie Review domain. This may explain the marginal significant difference between benefit for documents and sentences. To understand the effect of granularity alone, we compared the benefit from feature feedback for documents (without removing any noise) and sentences in movie review domain only and we found that this difference in also not significant. Thus, contrary to our intuition, sentences and documents seem to benefit equally from feature feedback.

The benefit is more when the feature space is larger and more diverse, i.e., when dependency features are used in addition to unigram features. We found that on average adding dependency features to unigram features increases the feature space by a factor of 10. With larger feature space, feature feedback can help to identify a few relevant features. As can also be seen, feature feedback is more helpful when there is more irrelevant text, i.e., there is noise that feature feedback can help to filter out. Unlike improvement in performance, the improve-

ment in learning rate does not decrease monotonically as the amount of rationale slack decreases. This supports our belief that improvement in performance does not necessarily imply improvement in the learning rate. We saw similar result when comparing benefit from feature feedback at different instance granularity. Improvement in learning rate for problems with different granularity was statistically significant but improvement in performance was not significant. Thus, both metrics should be used when evaluating the benefit from feature feedback.

We also observe that when training examples are selectively sampled as the most uncertain instances, we gain more from feature feedback than without selective sampling. This is intuitive as instances the model is uncertain about are likely to contain features it is uncertain about and hence the model should benefit from feedback on features in these instances. Next we evaluate how well the complexity measures proposed in Raghavan et al. (2007) correlate with improvement in performance and improvement in learning rate.

| $Var.$ | $Values$ | $AvgF1$ | Group | | |
|---|---|---|---|---|---|
| Strat. | wFF | 78.2 | A | | |
| | woFF | 68.2 | B | | |
| $Var.$ | $Values$ | $Avg_{IP}$ | $Grp_{IP}$ | $Avg_{ILR}$ | $Grp_{ILR}$ |
| D | WebKB | 11.9 | A | 0.32 | A |
| | MR | 8.0 | B | 0.20 | B |
| G | Doc | 10.9 | A | 0.30 | A |
| | Sent | 9.0 | A | 0.22 | B |
| F | u+d | 12.1 | A | 0.30 | A |
| | u | 7.8 | B | 0.22 | B |
| I | $\infty$ | 12.8 | A | 0.34 | A |
| | 600 | 11.2 | A B | 0.23 | B |
| | 400 | 11.1 | A B | 0.26 | A B |
| | 200 | 9.8 | B | 0.26 | A B |
| | 0 | 4.8 | C | 0.21 | B |
| S | Uncer. | 12.7 | A | 0.32 | A |
| | Deter. | 7.1 | B | 0.20 | B |

Table 2: Effect of variables defined in Table 1 on benefit from feature feedback. $Avg_{IP}$ is the average increase in performance ($F1$) and $Avg_{ILR}$ is the average increase in the learning rate. Different letters in $Grp_{IP}$ and $Grp_{ILR}$ indicate significantly different results.

For a given problem with an annotation cost budget of $a$ AUs, we calculate the benefit from feature feedback by comparing the performance with feature feedback on $\frac{a}{2}$ instances and the performance without feature feedback on $a$ instances as described in Section 3.1. The feature complexity measures are calculated using $\frac{a}{2}$ instances, since it should be the characteristics of these $\frac{a}{2}$ training instances that determine whether we would benefit from feature feedback on these $\frac{a}{2}$ instances or from labeling new $\frac{a}{2}$ instances. As can be seen from Table 3, the correlation of feature complexity measures with both measures of benefit from feature feedback is strong, negative and significant. This suggests that problems with low feature complexity, i.e. concepts that can be expressed with few well-selected features, benefit more from feature feedback.

It is intuitive that the benefit from feature feedback decreases as amount of labeled data increases. We found a significant negative correlation ($-0.574$) between annotation budget (number of $AUs$) and improvement in performance with feature feedback. However, note that this correlation is not very strong, which supports our belief that factors other than the amount of labeled data affect benefit from feature feedback.

| **Measure** | **R**($IP$) | **R**($ILR$) |
|---|---|---|
| $N_f$ | -0.625 | -0.615 |
| $F_{pc}$ | -0.575 | -0.735 |
| $C_f$ | -0.603 | -0.629 |

Table 3: Correlation coefficient (R) for feature size complexity ($N_f$), feature profile complexity ($F_{pc}$) and combined feature complexity ($C_f$) with improvement in performance ($IP$) and improvement in learning rate ($ILR$). All results are statistically significant ($p < 0.05$)

Feature complexity measures require an 'oracle' simulated using a large amount of labeled data which is not available for real annotation tasks. In Section 3.2, we proposed measures based on model's uncertainty that do not require an oracle. We calculate the mean, maximum and median of the probability scores from the learned model on instances in the held out dateset. We found a significant but low negative correlation of these measures with improvement in performance with feature feedback ($maxProb = -0.384$, $meanProb = -0.256$, $medianProb = -0.242$). This may seem counter-intuitive. However, note that when the training data is very small, the model might be quite certain about

its prediction even when it is wrong and feature feedback may help by correcting the model's beliefs. We observed that these probability measures have only medium and significant positive correlation (around $0.5$) with training datasize. Also, the held out dataset we used may not be representative of the whole set and using a larger dataset may give us more accurate estimate of the model's uncertainty. There are also other ways to measure the model's uncertainty, for example, in SVMs the distance of an instance from the decision boundary gives us an estimate of the model's uncertainty about that instance. We plan to explore additional measures for model's uncertainty in the future.

## 5 Conclusion and Future Work

In this work, we analyze how the benefit from feature feedback varies with different problem characteristics and how measures for categorizing learning problems correlate with benefit from feature feedback. We define a problem instance as a tuple of domain, instance granularity, feature representation, labeled data, amount of irrelevant text and selective sampling strategy.

We compare the two annotation strategies, with and without feature feedback, in terms of both improvement in performance at a given stage in learning and improvement in learning rate. Instead of evaluating the benefit from feature feedback using a specific feedback incorporation approach, we estimate and compare how the maximum benefit from feature feedback varies across different learning problems. This tells us what is the best feature feedback can do for a given learning problem.

We find a strong and significant correlation between feature complexity measures and the two measures of maximum benefit from feature feedback. However, these measures require an 'oracle', simulated using a large amount of labeled data which is not available in real world annotation tasks. We present measures based on the uncertainty of the model on its prediction that do not require an oracle. The proposed measures have a low but significant correlation with benefit from feature feedback. In our current work, we are exploring other measures of uncertainty of the model. It is intuitive that a metric that measures the uncertainty of the model on parameter estimates should correlate strongly with benefit from feature feedback. Variance in parameter estimates is one measure of uncertainty. The Bootstrap or Jacknife method (Efron and Tibshirani, 1994) of resampling from the training data is one way of estimating variance in parameter estimates that we are exploring.

So far only a linear relationship of various measures with benefit from feature feedback has been considered. However, some of these relationships may not be linear or a combination of several measures together may be stronger indicators of the benefit from feature feedback. We plan to do further analysis in this direction in the future.

We only considered one selective sampling strategy based on model's uncertainty which we found to provide more benefit from feature feedback. In the future, we plan to explore other selective sampling strategies. For example, density-based sampling (Donmez and Carbonell, 2008) selects the instances that are representative of clusters of similar instances, and may facilitate more effective feedback on a diverse set of features.

In this work, feature feedback was simulated using an oracle. Feedback from the users, however, might be less accurate. Our next step will be to analyze how the benefit from feature feedback varies as the quality of feature feedback varies.

Our eventual goal is to estimate the benefit from feature feedback for a given problem so that the right annotation strategy can be selected for a given learning problem at a given stage in learning and the total annotation cost for learning the target concept can be minimized. Note that in addition to the characteristics of the labeled data analyzed so far, expected benefit from feature feedback will also depend on the properties of the data to be labeled next for the two annotation strategies - with or without feature feedback.

## Acknowledgments

We thank Carolyn P. Rosé, Omid Madani, Hema Raghavan, Jaime Carbonell, Pinar Donmez and Chih-Jen Lin for helpful discussions, and the reviewers for their feedback. This work is supported by DARPA's Machine Reading program under contract FA8750-09-C-0172.

## References


Shilpa Arora and Eric Nyberg. 2009. Interactive annotation learning with indirect feature voting. In *Proceedings of NAACL-HLT 2009 (Student Research Workshop)*.

Shilpa Arora, Elijah Mayfield, Carolyn Penstein Rosé, and Eric Nyberg. 2010. Sentiment classification using automatically extracted subgraph features. In *Proceedings of the Workshop on Emotion in Text at NAACL*.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

William W. Cohen. 2004. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data.

Pinar Donmez and Jaime G. Carbonell. 2008. Paired Sampling in Density-Sensitive Active Learning. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*.

Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 595–602, New York, NY, USA. ACM.

Gregory Druck, Burr Settles, and Andrew McCallum. 2009. Active learning by labeling features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

B. Efron and R.J. Tibshirani. 1994. *An introduction to the bootstrap*. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, New York.

Mahesh Joshi and Carolyn Penstein Rosé. 2009. Generalizing dependency features for opinion mining. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 313–316, Morristown, NJ, USA. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

K. Lang. 1995. NewsWeeder: Learning to filter netnews. In *12th International Conference on Machine Learning (ICML95)*, pages 331–339.

D. Lewis. 1995. The reuters-21578 text categorization test collection.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association of Computational Linguistics (HLT/NAACL)*.

Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, 7:1655–1686.

Hema Raghavan, Omid Madani, and Rosie Jones. 2007. When will feature feedback help? quantifying the complexity of classification problems. In *IJCAI Workshop on Human in the Loop Computing*.

C. J. Van Rijsbergen. 1979. *Information Retrieval*. Butterworths, London, 2 edition.

Simon Tong and Daphne Koller. 2000. Support vector machine active learning with applications to text classification. In *JOURNAL OF MACHINE LEARNING RESEARCH*, pages 999–1006.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 260–267, Rochester, NY, April.