Active Learning with Rationales for Text Classification

Manali Sharma, Di Zhuang and **Mustafa Bilgic**
Department of Computer Science
Illinois Institute of Technology
Chicago, IL USA
{msharm11, dzhuang3}@hawk.iit.edu and mbilgic@iit.edu

Abstract

We present a simple and yet effective approach that can incorporate rationales elicited from annotators into the training of any off-the-shelf classifier. We show that our simple approach is effective for multinomial naïve Bayes, logistic regression, and support vector machines. We additionally present an active learning method tailored specifically for the learning with rationales framework.

1 Introduction

Annotating documents for supervised learning is a tedious, laborious, and time consuming task for humans. Given huge amounts of unlabeled documents, it is impractical for annotators to go over each document and provide a label. To reduce the annotation time and effort, various approaches such as semi-supervised learning (Chapelle et al., 2006) that utilizes both labeled and unlabeled data, and active learning (Settles, 2012) that carefully chooses instances for annotation have been developed. To further minimize the human effort, recent work looked at eliciting domain knowledge, such as rationales and feature annotations, from the annotators instead of just the labels of documents.

One of the bottlenecks in eliciting domain knowledge from annotators is that the traditional supervised learning approaches cannot readily handle the elicited rich feedback. To address this issue, many methods have been developed that are classifier-specific. Examples include knowledge-based neural networks (e.g., (Towell and Shavlik, 1994), (Girosi and Chan, 1995), (Towell et al., 1990)), knowledge-based support vector machines (Fung et al., 2002), pooling multinomial naïve Bayes (Melville and Sindhwani, 2009), incorporating constraints into the training of naïve Bayes (Stumpf et al., 2007), and converting rationales and feature annotations into constraints for support vector machines (e.g., (Small et al., 2011) and (Zaidan et al., 2007)). Being classifier-specific limits their applicability when one wants to test a different classifier for his/her domain, necessitating an approach that can be utilized by several off-the-shelf classifiers.

In this paper we present a simple and yet effective approach that can incorporate the elicited rationales in the form of feature annotations into the training of any off-the-shelf classifier. We empirically show that it is effective at incorporating rationales into the learning of naïve Bayes, logistic regression, and support vector machines using four text categorization datasets. We further discuss a novel active learning strategy specifically geared towards the learning with rationales framework and empirically show that it improves over traditional active learning.

The rest of the paper is organized as follows. In Section 2, we provide a brief background on eliciting rationales in the context of active learning. In Section 3, we describe our approach for incorporating rationales into the training of classifiers and compare learning without rationales and learning with rationales. In Section 4, we present an active learning method using the learning with rationales framework and present relevant results. Finally, we discuss limitations and future work in Section 5, related work in Section 6, and conclude in Section 7.

2 Background

In this section, we provide a brief background on data annotation with rationales in the context of active learning and introduce the notation to be used throughout the paper.

Let $\mathcal{D}$ be a set of document-label pairs $\langle x, y \rangle$, where the label (value of y) is known only for a small subset $\mathcal{L} \subset \mathcal{D}$ of the documents: $\mathcal{L} = \{\langle x, y \rangle\}$ and the rest $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$ consists of the unlabeled documents: $\mathcal{U} = \{\langle x, ? \rangle\}$. We assume that each document x^i is represented as a vector of features (most commonly as a bag-of-words model with a dictionary of predefined set of phrases, which can be unigrams, bigrams, etc.): $x^i \triangleq \{f^i_1, f^i_2, \cdots, f^i_n\}$. Each feature f^i_j represents the binary presence (or absence), frequency, or tf-idf representation of the word/phrase j in document x^i. Each label $y \in \mathcal{Y}$ is discrete-valued variable $\mathcal{Y} \triangleq \{y_1, y_2, \cdots, y_l\}$.

Typical greedy active learning algorithms iteratively select a document $\langle x, ? \rangle \in \mathcal{U}$, query a labeler for its label y, and incorporate the new document $\langle x, y \rangle$ into its training set $\mathcal{L}$. This process continues until a stopping criterion is met, usually until a given budget, B, is exhausted.

In the learning with rationales framework, in addition to querying for label y^i of a document x^i, the active learner asks the labeler to provide a rationale, $R(x^i)$ for the chosen label. The rationale in its most general form consists of a subset of the terms that are present in x^i: $R(x^i) = \{f^i_k : j \in x^i\}$. Note that there might be cases where the labeler cannot pinpoint any phrase as a rationale, in which case $R(x^i)$ is allowed to be ϕ. Algorithm 1 formally describes the active learning process that elicits rationales from the labeler.

The goal of eliciting rationales is to improve the learning efficiency by incorporating domain knowledge. However, it is not trivial to integrate domain knowledge into state-of-the-art classifiers, such as logistic regression and support vector machines.

Next, we describe our approach for incorporating rationales into the learning process.

3 Learning with Rationales

In this section we first provide the formulation of our approach to incorporate rationales into learning and then present the results to compare *learning without rationales* (Lw/oR) and *learning with rationales* (LwR) on four datasets. We evaluate our approach using multinomial naïve Bayes, logistic regression, and support vector machines classifiers.

Algorithm 1 Active Learning with Rationales

```
Input: $\mathcal{U}$ - unlabeled documents, $\mathcal{L}$ - labeled documents, $\theta$ - underlying classification model, $B$ - budget
repeat
    $x^* = \underset{x \in \mathcal{U}}{\operatorname{argmax}}\, utility(x|\theta)$
    request label and rationale for this label
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle x^*, y^*, R(x^*) \rangle\}$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle x^* \rangle\}$
    Train $\theta$ on $\mathcal{L}$
until Budget $B$ is exhausted; e.g., $|\mathcal{L}| = B$
```

3.1 Training a Classifier Using Labels and Rationales

Like most previous work, we assume that the rationales, i.e. the phrases, returned by the labeler already exist in the dictionary of the vectorizer. Hence, rationales correspond to features in our vector representation. It is possible that the labeler returns a phrase that is currently not in the dictionary; for example, the labeler might return a phrase that consists of three words whereas the representation has single words and bi-grams only. In that case, the representation can be enriched by creating and adding a new feature that represents the phrase returned by the labeler.

Our simple approach works as follows: we modify the features of the annotated document $\langle x^*, y^*, R(x^*) \rangle$ to emphasize the rationale(s) and de-emphasize the remaining phrases in that document. We simply multiply the features corresponding to phrase(s) that are returned as rationale(s) by weight r and we multiply the remaining features in the document by weight o, where $r > o$, and r and o are hyper-parameters. The modified document becomes:

$$x^i_{\prime} = \langle r \times f^i_j, \forall f^i_j \in R(x^i); o \times f^i_j, \forall f^i_j \notin R(x^i), \rangle \tag{1}$$

Note that the rationales are tied to their documents for which they were provided as rationales. One phrase might be a rationale for the label of one document and yet it might not be the rationale for the label of another document. Hence, the feature weightings are done at the document level, rather than globally. To illustrate this concept, we provide an example dataset below with three documents. In these documents, the words that are returned as rationales are underlined.

Document 1: This is a great movie.

Document 2: The plot was great, but the performance of the actors was terrible. Avoid it.

Document 3: I've seen this at an outdoor cinema; great atmosphere. The movie was terrific.

As these examples illustrate, the word "great" appears in all three documents, but it is marked as a rationale only for Document 1. Hence, we do not weight the rationales globally; rather, we modify only the labeled document using its particular rationale. Table 1 illustrates both the Lw/oR and LwR representations for these documents.

Table 1: The Lw/oR binary representation (top) and its LwR transformation (bottom) for Documents 1, 2, and 3. Stop words are removed. LwR multiplies the rationales with r and other features with o.

	great	movie	plot	performance	actor	terrible	avoid	outdoor	cinema	atmosphere	terrific
Lw/oR Representation (binary)											
D1	1	1									
D2	1		1	1	1	1	1				
D3	1	1						1	1	1	1
LwR Transformation of the binary Lw/oR repr.											
D1	r	o									
D2	o		o	o	o	r	r				
D3	o	o						o	o	o	r

This approach is simple, intuitive, and classifier-agnostic. As we will show later, it is quite effective empirically as well. To gain a theoretical understanding of this approach, consider the work on regularization: the aim is to build a sparse/simple model that can capture the most important features of the training data and thus have large weights for important features and small/zero weights for irrelevant features. For example, consider the gradient for w_j of feature f_j for logistic regression with l_2 regularization (assuming y is binary with 0/1):

$$\nabla w_j = C \times \sum_{x^l \in \mathcal{L}} f_j^l \times (y^l - P(y = 1|x^l)) - w_j \quad (2)$$

where C is the complexity parameter that balances between fit to the data and the model complexity.

With our rationales framework, the gradient for w_j will be:

$$\begin{aligned} &\nabla w_j = \\ &C \times \left(\sum_{x^l \in \mathcal{L}: f_j^l \in R(x^l)} r \times f_j^l \times (y^l - P(y^l = 1|x^l)) \right. \\ &\quad \left. + \sum_{x^l \in \mathcal{L}: f_j^l \notin R(x^l)} o \times f_j^l \times (y^l - P(y^l = 1|x^l)) \right) \\ &\quad - w_j \qquad (3) \end{aligned}$$

In the above equation, a feature f_j contributes more to the gradient of its weight w_j when a document in which it is marked as a rationale is misclassified. When f_j appears in another document x^k but is not a rationale, it's contribution to the gradient is muted by o. And hence, when $r > o$, this framework implicitly provides more granular (per instance-feature combination) regularization by placing a higher importance on the contribution of the rationales versus non-rationales in each document.[1]

Note that in our framework the rationales are tied to their own documents; that is, we do not weight rationales and non-rationales globally. In addition to providing more granular regularization, this approach has the benefit of allowing different rationales to contribute differently to the objective function of the trained classifier. For example, consider the case where the number of documents in which one word f_j (e.g., "excellent") is marked as a rationale is much more than the number of documents where another word f_k (e.g., "good") is marked as

[1] The justification for our approach is similar for support vector machines. The idea is also similar for multinomial naïve Bayes with Dirichlet priors α_j. For a fixed Dirichlet prior with $\langle \alpha_1, \alpha_2, \cdots, \alpha_n \rangle$ setting, when $o < 1$ for a feature f_j, its counts are smoothed more.

a rationale. Then, the first sum in equation 3 will range over more documents for the gradient of w_j compared to the gradient of w_k, giving more importance to w_j than to w_k. In the traditional feature annotation work, this can be achieved only if the labeler can rank the features; even then, it is often very difficult, if not impossible, for the labelers to determine how much more important one feature is compared to another.

3.2 Experiments Comparing Lw/oR to LwR

In this section we first describe the settings, datasets, and classifiers used for our experiments and how we simulated a human labeler to provide rationales. Then, we present the results comparing the learning curves achieved with *learning without rationales* (Lw/oR) and *learning with rationales* (LwR).

3.2.1 Methodology

For this study, we used four text classification datasets. The IMDB dataset consists of 25K movie reviews (Maas et al., 2011). The SRAA[2] dataset consists of 48K documents that discuss either auto or aviation. Nova is a text classification dataset used in active learning challenge (Guyon, 2011) and contains 12K documents. WvsH is a 20 Newsgroups[3] dataset in which we use the Windows vs. hardware categories, and it contains 1176 documents.

To make sure our approach works across representations, we experimented with both binary and tf-idf representations for these text datasets. We evaluated our strategy using multinomial naïve Bayes, logistic regression, and support vector machines, as these are strong classifiers for text classification. We used the scikit-learn (Pedregosa et al., 2011) implementation of these classifiers with their default parameter settings for our experiments.

To compare various strategies, we used learning curves. The initially labeled dataset was bootstrapped using 10 documents by picking 5 random documents from each class. A budget (B) of 200 documents was used in our experiments, because most of the learning curves flatten out after about 200 documents. We evaluated all the strategies using AUC (Area Under an ROC Curve) measure. The code to repeat our experiments is available at Github `http://www.cs.iit.edu/~ml/code/`.

While incorporating the rationales into learning, we set the weights for rationales and the remaining features of a document as 1 and 0.01 respectively (i.e. $r = 1$ and $o = 0.01$). That is, we did not overemphasize the features corresponding to rationales but rather de-emphasized the remaining features in the document. These weights worked reasonably well for all four datasets, across all three classifiers, and for both binary and tf-idf data representations.

Obviously, these are not necessarily the best weight settings one can achieve; the optimal settings for r and o depend on many factors, such as the extent of the knowledge of the labeler (i.e., how many words a labeler can recognize), how noisy the labeler is, and how much labeled data we have in our training set. Ideally, one should have $r >> o$ when the labeled data is small and r should be closer to o when the labeled data is large; a more practical approach would be to tune for these parameters (e.g., cross-validation) at each step of the learning curve. However, in our experiments, we fixed r and o and we found that most settings where $r > o$ worked quite well.

3.2.2 Simulating the Human Expert

Like most literature on feature labeling, we constructed an artificial labeler to simulate a human labeler. Every time a document is annotated, we asked the artificial labeler to mark a word as a rationale for that document's label. We allowed the labeler to return any one (and not necessarily the top one) of the positive words as a rationale for a positive document and any one of the negative words as a rationale for a negative document. If the labeler did not recognize any of the words as positive (negative) in a positive (negative) document, we let the labeler return nothing as the rationale. To make this as practical as possible in a real-world setting, we constructed the artificial labeler to recognize only the most apparent words in the documents. For generating rationales, we chose only the positive (negative) features that had the highest χ^2 (chi-squared) statistic in at least 5% of the positive (negative) documents. This resulted in an overly-conservative labeler that recognized only a tiny subset of the words. For example,

[2] http://people.cs.umass.edu/ mccallum/data.html

[3] http://qwone.com/ jason/20Newsgroups/

the artificial labeler knew about only 49 words (23 for one class and 26 for the other class) for IMDB, 67 words (32 for one class and 35 for the other class) for SRAA, 95 words (42 for one class and 53 for the other class) for WvsH, and 111 words (31 for one class and 80 for the other class) for the Nova dataset.

To determine whether the rationales selected by this artificial labeler are meaningful, we printed out the actual words used as rationales, and we ourselves verified that these words are human-recognizable words that could be naturally provided as rationales for classification. For example, the positive terms for the IMDB dataset included “great”, “excellent”, and “wonderful” and the negative terms included “worst”, “bad”, and “waste.”

3.2.3 Results

Next, we compare Lw/oR to LwR. Figure 1 presents the learning curves for random sampling on four text classification datasets with binary and tf-idf representations and using multinomial naïve Bayes, logistic regression, and support vector machines. Figure 1 shows that even though the artificial labeler knew only about a tiny subset of the vocabulary, and returned any *one* word, rather than the top word or all the words, as rationale, LwR still drastically outperformed Lw/oR across all datasets, classifiers, and representations. This shows that our method for incorporating rationales into the learning process is empirically effective.

We used the default complexity parameters for logistic regression and support vector machines and used Laplace smoothing for multinomial naïve Bayes. In our rationale framework, most features were non-rationales, and hence in Equation 3, most features appeared in the second summation term, with $o = 0.01$. We tested whether the improvements that LwR provide over Lw/oR are simply due to implicit higher regularization for most of the features with $o = 0.01$, and hence experimented with equation 2 (which is Lw/oR) using $C = 0.01$. We observed that setting $C = 0.01$ and indiscriminately regularizing all the terms did not improve Lw/oR, further providing experimental evidence that the improvements provided by LwR are not due to just higher regularization, but they are due to a more fine-grained regularization, as explained in Section 3.1.

Even though LwR provides huge benefits, providing both a label and a rationale is expected to take more time of the labeler than simply providing a label. However, the improvements of LwR over Lw/oR is so huge that it might be worth spending the extra time in providing rationales. For example, in order to achieve a target AUC of 0.95 for SRAA dataset (using tf-idf representation with MNB classifier), Lw/oR required labeling 656 documents, whereas LwR required annotating a mere 29 documents, which is 22.6 times reduction in the number of documents. As another example, in order to achieve a target AUC of 0.8 for WvsH dataset (using binary representation with SVM classifier), Lw/oR required labeling 113 documents, whereas LwR achieved this target with only 13 documents.

(Zaidan et al., 2007) conducted user studies and showed that providing 5 to 11 rationales and a class label per document takes roughly twice the time of providing only the label for the document. (Raghavan et al., 2006) also conducted user studies and showed that labeling instances takes five times more time than labeling features. We worked with simulated user and showed that a document that is annotated with a label and a single rationale can be worth as many as 22 documents that are annotated with only a label and thus these results suggest that LwR, compared to Lw/oR, can lead to significant time savings for the annotator.

4 Active Learning with Rationales

So far we have seen that LwR provides drastic improvements over Lw/oR. Both these strategies selected documents randomly for labeling. Active learning (Settles, 2012) aims to carefully choose instances for labeling to improve over random sampling. Many successful active learning approaches have been developed for instance labeling (e.g. (Lewis and Gale, 1994), (Seung et al., 1992), (Roy and McCallum, 2001)), feature labeling (e.g. (Druck et al., 2009)), and rotating between instance and feature labeling (e.g. (Raghavan and Allan, 2007), (Druck et al., 2009), (Attenberg et al., 2010), (Melville and Sindhwani, 2009)). In this section, we introduce an active learning strategy that can utilize the learning with rationales framework.

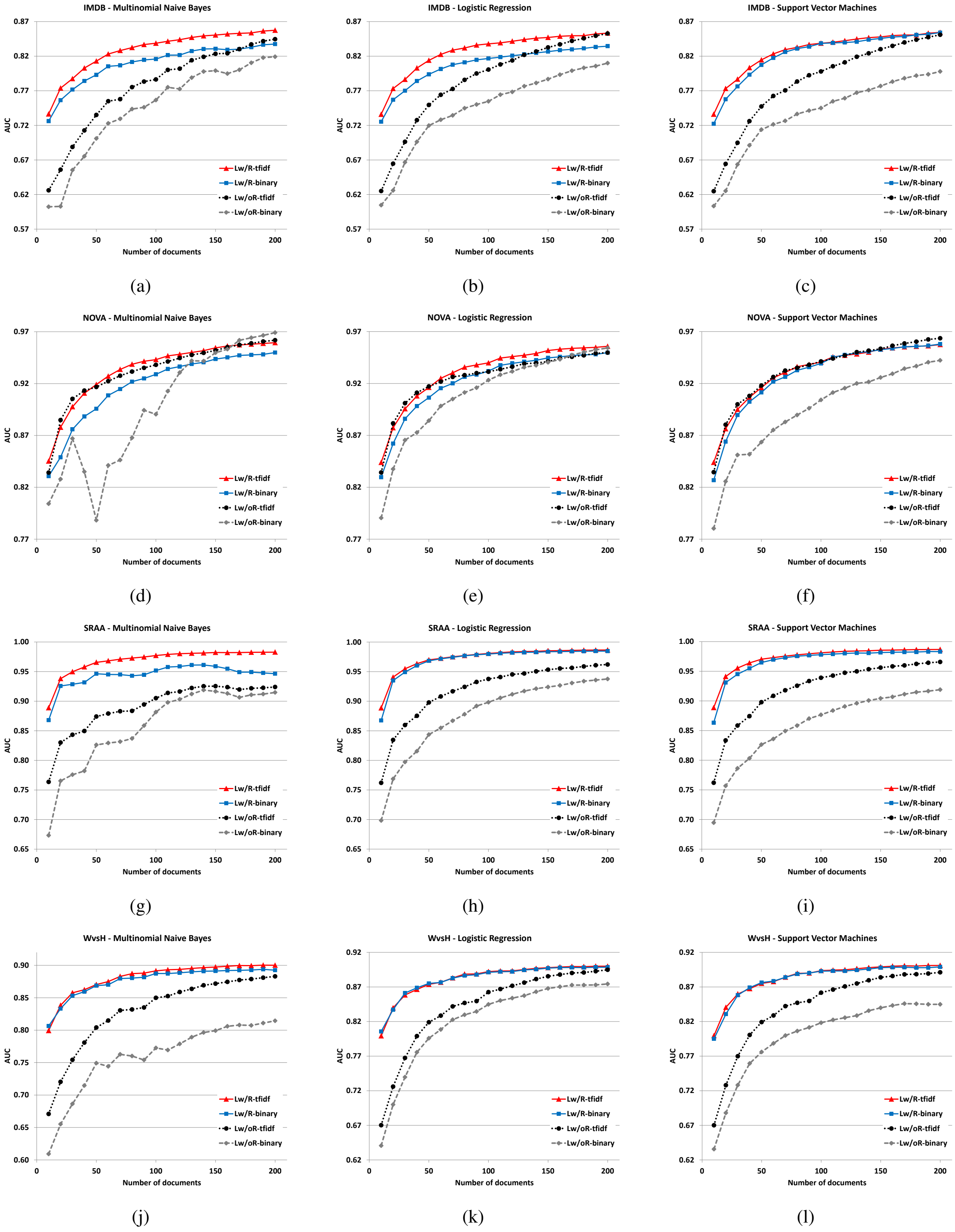


Figure 1: Comparison of Lw/oR with LwR. LwR provides drastic improvements over Lw/oR for all datasets with binary and tf-idf representations and using all three classifiers.

4.1 Active Learning to Select Documents based on Rationales

Arguably, one of the most successful active learning strategies for text categorization is uncertainty sampling, which was first introduced by (Lewis and Catlett, 1994) for probabilistic classifiers and later formalized for support vector machines (Tong and Koller, 2001). The idea is to label instances for which the underlying classifier is uncertain, i.e., the instances that are close to the decision boundary of the model. It has been successfully applied to text classification tasks in numerous publications, including (Zhu and Hovy, 2007), (Sindhwani et al., 2009), and (Segal et al., 2006).

We adapt uncertainty sampling for the learning with rationales framework. To put simply, when the underlying model is uncertain about an unlabeled document, we look whether the unlabeled document contains words/phrases that were returned as rationales for any of the existing labeled documents. More formally, let R^+ denote the union of all the rationales returned for the positive documents so far. Similarly, let R^- denote the union of all the rationales returned for the negative documents so far. An unlabeled document can be of these three types:

1. Type1: has no words in common with R^+ and R^-.
2. Type2: has word(s) in common with either R^+ or R^- but not both.
3. Type3: has at least one word in common with R^+ and at least one word in common with R^-.

One would imagine that labeling each of the type1, type2, and type3 documents has its own advantage. Labeling type1 documents has the potential to elicit new domain knowledge, i.e., terms that were not provided as a rationale for any of the existing labeled documents. It also carries the risk of containing little to no useful information for the classifier (e.g., a neutral review). For type2 documents, even though the document shares a word that was returned as a rationale for another document, the classifier is still uncertain about the document either because that word is not weighted high enough by the classifier and/or there are other words that pull the classification decision in the other direction, making the classifier uncertain. Type3 documents contain conflicting words/phrases and are potentially harder cases, however, they also have the potential to resolve the conflicts for the classifier.

Building on our previous work (Sharma and Bilgic, 2013) we devised an active learning approach, where given uncertain documents, the active learner prefers instances of type3 over type1 and type2. We call this strategy as *uncertain-prefer-conflict* (UNC-PC) because type3 documents carry conflicting words (with respect to rationales) whereas type1 and type2 do not. The difference between this approach and our previous work (Sharma and Bilgic, 2013) is that in (Sharma and Bilgic, 2013), we selected uncertain instances based on model's perceived conflict whereas in this work, we are selecting documents based on conflict caused by the domain knowledge provided by the labeler. Next, we compare the vanilla uncertainty sampling (UNC) and UNC-PC strategies using LwR to see if using uncertain documents of type3 could improve active learning.

4.2 Active Learning with Rationales Experiments

We used the same four text datasets and evaluated our method UNC-PC using multinomial naïve Bayes, logistic regression, and support vector machines. For the active learning strategies, we used a bootstrap of 10 random documents, and labeled five documents at each round of active learning. We used a budget of 200 documents for all methods. UNC simply picks the top five uncertain documents, whereas UNC-PC looks at top 20 uncertain documents and picks five uncertain documents giving preference to the conflicting cases (type 3) over the non-conflicting cases (type1 and type2). We repeated each experiment 10 times starting with a different bootstrap at each trial and report the average results.

In Figure 2 we show the learning curves comparing UNC-PC with UNC for multinomial naïve Bayes. (Logistic regression and SVM curves are omitted due to space.) Since the results for LwR using tf-idf representation are better than the results using the binary representation, we compared UNC-PC to UNC for LwR using only the tf-idf representation. We see that for multinomial naïve

Bayes, UNC-PC improves over traditional uncertainty, UNC, on two datasets, and hurts performance on one dataset. Next, we discuss the significance results for all classifiers.

Table 2 shows the paired t-test results comparing the learning curves of UNC-PC with the learning curves of UNC at each step of the active learning (i.e, if the average of one learning curve is significantly better or worse than the average of the learning curve of the other). If UNC-PC has a higher average AUC than UNC with a t-test significance level of 0.05 or better, it is a Win (W), if it has significantly lower performance, it is a Loss (L), and if the difference is not statistically significant, the result is a Tie (T).

Using multinomial naïve Bayes, UNC-PC wins over UNC for two of the datasets (IMDB and WvsH), does not cause any significant changes for Nova (ties all the time), and loses for SRAA. Using logistic regression, UNC-PC wins for two datasets (Nova and SRAA), ties for WvsH and loses for IMDB. Using support vector machines, UNC-PC wins for three datasets (Nova, SRAA, and WvsH) and loses for IMDB. The t-test results show that UNC-PC often improves learning over UNC.

Table 2: Significant W/T/L counts for UNC-PC versus UNC. UNC-PC improves over UNC significantly for all three classifiers and most of the datasets.

UNC baseline	**MNB**	**LR**	**SVM**
UNC-PC	**2**/1/1	**2**/1/1	**3**/0/1

5 Limitations and Future Work

A limitation of our work is that we simulated the labeler in our experiments. Even though we simulated the labeler in a very conservative way (that is, our simulated labeler knows only a few most apparent words) and asked the simulated labeler to provide any one (rather than the top) rationale, a user study is needed to i) experiment with potentially noisy labelers, and ii) measure how much actual time saving LwR provides over Lw/oR.

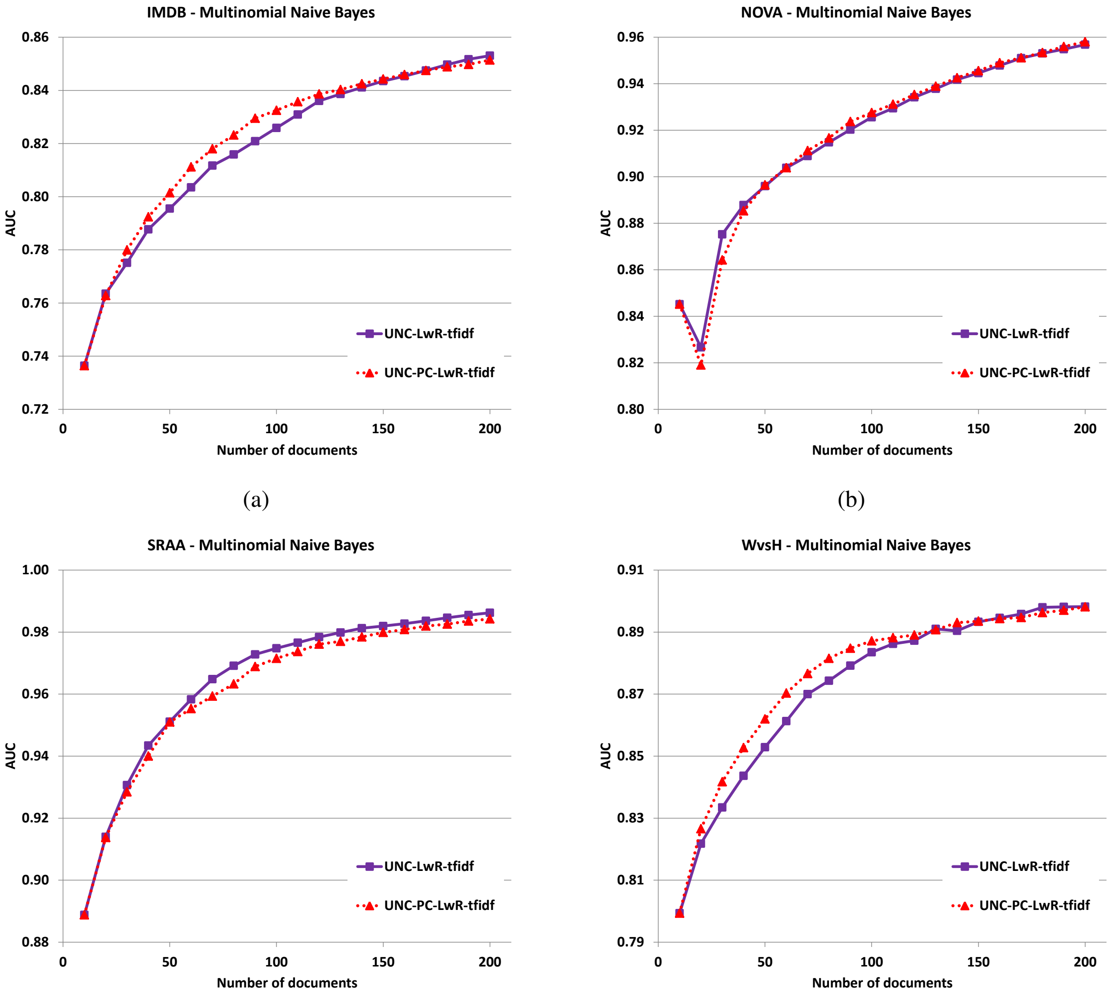
Another line of future work is to allow the labeler to provide richer feedback. This is especially useful for resolving conflicts that stem from seemingly conflicting words and phrases. For example, for the movie review “The plot was great, but the performance of the actors was terrible. Avoid it.” the word “great” is at odds with the words “terrible” and “avoid”. If the labeler is allowed to provide richer feedback, saying that the word “great” refers to the plot, “terrible” refers to the performance, and “avoid” refers to the movie, then the learner might be able to learn to resolve similar conflicts in other documents. However, this requires a conflict resolution mechanism in which the labeler can provide rich feedback, and a learner that can utilize such rich feedback. This is an exciting future research direction that we would like to pursue.

We showed that our strategy to incorporate rationales works well for text classification. The proposed framework can potentially be used for non-text domains where the domain experts can provide rationales for their decisions, such as medical domain where the doctor can provide a rationale for his/her diagnosis and treatment decisions. Each domain is expected to have its own unique research challenges and working with other domains is another interesting future research direction.

6 Related Work

The closest related work deals with eliciting rationales from users and incorporating them into the learning (e.g., (Zaidan et al., 2007), (Donahue and Grauman, 2011), (Zaidan et al., 2008), and (Parkash and Parikh, 2012)). However, much of this work is specific to a particular classifier, such as support vector machines. The framework we present is classifier-agnostic and we have shown that it works across classifiers and feature representations. Additionally, we provide a novel active learning approach tailored for the learning with rationales framework.

Another line of related work is the recent work on active learning with instance and feature annotations (e.g., (Melville and Sindhwani, 2009), (Druck et al., 2009), (Small et al., 2011), (Stumpf et al., 2008), (Raghavan and Allan, 2007), and (Attenberg et al., 2010)). The main difference between the feature annotation work and the learning with rationales framework is that the feature annotations are not tied to particular instances, whereas in the learning with rationales framework, the documents and their rationales are coupled together. Even though



(a)

(b)

(c)

(d)

Figure 2: Comparison of LwR using UNC and UNC-PC for all datasets with tf-idf representation and using multinomial naïve Bayes classifier.

feature annotation work can be utilized for the learning with rationales framework by decoupling rationales from their documents, this is expected to result in information loss (such as weighting features globally rather than locally). The precise effect of decoupling rationales and documents on the classifier performance still needs to be tested empirically.

7 Conclusion

We introduced a novel framework to incorporate rationales into active learning for text classification. Our simple strategy to incorporate rationales can utilize any off-the-shelf classifier. The empirical evaluations on four text datasets with binary and tf-idf representations and three classifiers showed that our proposed method utilizes rationales effectively. Additionally, we presented an active learning strategy that is tailored specifically for the learning with rationales framework and empirically showed that it improved active learning.

Acknowledgment

This material is based upon work supported by the National Science Foundation CAREER award no. 1350337.

References

Josh Attenberg, Prem Melville, and Foster Provost. 2010. A unified approach to active dual supervision for labeling features and examples. In *European conference on Machine learning and knowledge discovery in databases*, pages 40–55.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

Jeff Donahue and Kristen Grauman. 2011. Annotator rationales for visual recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1395–1402.

G. Druck, B. Settles, and A. McCallum. 2009. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 81–90.

Glenn M Fung, Olvi L Mangasarian, and Jude W Shavlik. 2002. Knowledge-based support vector machine classifiers. In *Advances in neural information processing systems*, pages 521–528.

Federico Girosi and Nicholas Tung Chan. 1995. Prior knowledge and the creation of virtual examples for rbf networks. In *Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop*, pages 201–210.

Isabell Guyon. 2011. Results of active learning challenge.

D.D. Lewis and J. Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150.

Prem Melville and Vikas Sindhwani. 2009. Active dual supervision: Reducing the cost of annotating examples and features. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 49–57.

Amar Parkash and Devi Parikh. 2012. Attributes for classifier feedback. In *Computer Vision–ECCV 2012*, pages 354–368. Springer.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Hema Raghavan and James Allan. 2007. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 79–86.

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

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448.

Richard Segal, Ted Markowitz, and William Arnold. 2006. Fast uncertainty sampling for labeling large e-mail corpora. In *Conference on Email and Anti-Spam*.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *ACM Annual Workshop on Computational Learning Theory*, pages 287–294.

Manali Sharma and Mustafa Bilgic. 2013. Most-surely vs. least-surely uncertain. In *IEEE 13th International Conference on Data Mining*, pages 667–676.

Vikas Sindhwani, Prem Melville, and Richard D Lawrence. 2009. Uncertainty sampling and transductive experimental design for active dual supervision. In *Proceedings of the International Conference on Machine Learning*, pages 953–960.

Kevin Small, Byron Wallace, Thomas Trikalinos, and Carla E Brodley. 2011. The constrained weight space svm: learning with ranked features. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 865–872.

Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett, Thomas Dietterich, Erin Sullivan, Russell Drummond, and Jonathan Herlocker. 2007. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91.

S. Stumpf, E. Sullivan, E. Fitzhenry, I. Oberst, W.K. Wong, and M. Burnett. 2008. Integrating rich user feedback into intelligent user interfaces. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 50–59.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66.

Geoffrey G Towell and Jude W Shavlik. 1994. Knowledge-based artificial neural networks. *Artificial intelligence*, 70(1):119–165.

Geofrey G Towell, Jude W Shavlik, and Michiel Noordewier. 1990. Refinement of approximate domain theories by knowledge-based neural networks. In *Proceedings of the eighth National conference on Artificial intelligence*, pages 861–866.

Omar Zaidan, Jason Eisner, and Christine D Piatko. 2007. Using" annotator rationales" to improve machine learning for text categorization. In *HLT-NAACL*, pages 260–267.

Omar F Zaidan, Jason Eisner, and Christine Piatko. 2008. Machine learning with annotator rationales to reduce annotation cost. In *Proceedings of the NIPS* 2008 Workshop on Cost Sensitive Learning*.

J. Zhu and E. Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 783–790.