Towards Learning with Feature-Based Explanations for Document Classification

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

Abstract

Traditional supervised learning approaches for document classification ask human labelers to provide labels for documents. Humans know more than just the labels and can provide extra information such as domain knowledge, feature annotations, rules, and rationales for classification. Researchers indeed tried to utilize this extra information for labeling of features and documents in tandem, and more recently for incorporating rationales for the provided labels. We extend this approach further by allowing the human to provide explanations, in the form of domain-specific features that support and oppose the classification of documents, and present an approach to incorporate explanations into the training of any off-the-shelf classifier to speed-up the learning process.

1 Introduction

Supervised learning approaches learn the class concepts using instances that are annotated with labels. When the labels for instances are not available, traditional active learning approaches [Ramirez-Loaiza *et al.*, 2016; Settles, 2012] ask humans to curate datasets by providing labels for selected instances to learn an effective classifier. While examining instances, human labelers can provide information beyond just label annotations. Humans can provide domain knowledge, point out important features, provide feature annotations, rationales, and rules for classification. Many studies have shown that, unsurprisingly, supervised learning can benefit if the domain knowledge or reasonings for classification are imparted to the models. However, the main challenge has been to effectively incorporate this domain knowledge, which is often noisy and uncertain, into the training of the machine learning system.

Transmitting domain knowledge to learning systems has been studied for many years. For example, expert systems relied heavily on eliciting domain knowledge from the experts (e.g., Mycin system [Buchanan *et al.*, 1984] was built through eliciting rules from the experts). Several explanation-based learning approaches (e.g.,[Mitchell *et al.*, 1986] and [DeJong and Mooney, 1986]) were developed to utilize domain knowledge to generalize target concepts using a single training example, and relied on domain experts to provide explanations for generalization. Examples of explanation-based learning systems include GENESIS [Mooney, 1986] and SOAR [Laird *et al.*, 1986]. Ellman [1989] provides a survey on explanation-based learning. However, incorporating domain knowledge into the learning process and teaching the classification reasonings to supervised models is not trivial. Many supervised learning systems operate on feature-based representations of instances. For example, in document classification, instances are typically represented as feature vectors in a bag-of-words model. The domain knowledge elicited from the experts, however, often cannot be readily parsed into the representation that the underlying model can understand or operate on. The domain knowledge often refers to features rather than specific instances. Moreover, the domain knowledge is often at a higher level than instances, and sometimes, the domain knowledge is provided as unstructured information, such as free-form text entries.

Several approaches have been developed for knowledge-based classifiers such as knowledge-based neural networks [Towell and Shavlik, 1994; Girosi and Chan, 1995; Towell *et al.*, 1990] and knowledge-based support vector machines [Fung *et al.*, 2002]. Recent approaches for document classification have explored incorporating feature annotations [Melville and Sindhwani, 2009; Druck *et al.*, 2009; Small *et al.*, 2011; Stumpf *et al.*, 2008; Raghavan and Allan, 2007; Attenberg *et al.*, 2010] and eliciting rationales for text classification [Zaidan *et al.*, 2007; Donahue and Grauman, 2011; Parkash and Parikh, 2012]. These approaches were specific to classifiers, and hence, in a more recent work [Sharma *et al.*, 2015], we proposed an approach to incorporate rationales for classification into the training of any off-the-shelf classifier, and used evidence-based framework [Sharma and Bilgic, 2013; 2016] for actively choosing documents for annotation.

In this paper, we ask the labeler to provide explanations for their classification. Specifically, we ask the labeler to highlight the phrases in a document that *support* its label (i.e., the phrases whose presence reinforces the belief in the provided label) and phrases that *oppose* its label (i.e., the phrases whose presence weakens the belief in the provided label). For example, in a movie review “The actors were great but the plot was terrible. Avoid it”, that is labeled as a ‘negative’ review, the phrases ‘terrible’ and ‘avoid’ support the ‘negative’ classification, whereas the word ‘great’ opposes the 'negative' classification. We present a simple and effective approach to incorporate these two types of explanations along with the labeled documents into the training of any off-the-shelf classifier. We evaluate our approach on three document classification datasets using multinomial naïve Bayes and support vector machines.

The rest of the paper is organized as follows. In Section 2, we provide a brief background on eliciting labels and explanations during the curation of datasets. In Section 3, we describe our approach for incorporating explanations into the training of classifiers. In Section 4, we discuss experimental methodology and results. Finally, we discuss limitations and future work in Section 5, and then conclude in Section 6.

2 Background

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: $x^i \triangleq \{f_1^i, f_2^i, \cdots, f_n^i\}$. Each feature f_j^i represents the binary presence (or absence), frequency, or tf-idf representation of the 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 supervised learning approaches for data curation 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 the training set $\mathcal{L}$.

Several approaches looked at eliciting more than just labels from annotators. For example, feature annotation work looked at annotating features in tandem with labeling of documents (e.g., [Melville and Sindhwani, 2009; Druck *et al.*, 2009; Small *et al.*, 2011; Stumpf *et al.*, 2008; Raghavan and Allan, 2007; Attenberg *et al.*, 2010]). More recently, [Zaidan *et al.*, 2007] and [Sharma *et al.*, 2015] looked at eliciting rationales for the chosen label. In this paper, we go one step further, and instead of asking simply the rationales, we ask for an explanation for the chosen label.

Explanations can be pretty broad, such as free-form text entries, rules, feature annotations, and rationales for classification. In this paper, we focus on explanations for document classification where the human annotator highlights phrases in the document as explanations. Specifically, we ask the labeler to provide two kinds of highlighting. In the first kind, the human highlights the phrases that *support* the underlying label. For example, in sentiment classification, the human would highlight the positive sentiments in a positive review. In the second kind of highlighting, the human highlights the kind of phrases which, if were not present, would make the provided label even more correct. For example, these would be the negative sentiments in a generally-positive review.

Formally, in the learning with explanations framework, when a document is chosen for annotation, the labeler provides label y^i for a document x^i, and explanations, which correspond to supporting features $SF(x^i)$ and opposing features $OF(x^i)$ for the label of x^i: $SF(x^i) = \{f_k^i : k \in x^i\}$ and $OF(x^i) = \{f_j^i : j \in x^i\}$. It is possible that the labeler cannot pinpoint any supporting or opposing phrases, in which case $SF(x^i)$ and $OF(x^i)$ are allowed to be empty sets. Next, we describe our approach for incorporating explanations into the learning process.

3 Learning with Explanations

In this section, we describe our approach to incorporate feature-based explanations into the training of any off-the-shelf feature-based classifier. We assume that the explanations, i.e. the supporting and opposing features, returned by the labeler already exist within the dictionary of the underlying model.[1] For each labeled document, $\langle x^i, y^i, SF(x^i), OF(x^i) \rangle$, we create four types of pseudo-documents as follows:

- For each supporting feature in $SF(x^i)$, we create one pseudo-document containing only one phrase corresponding to a supporting feature, weight the supporting feature by w_s, and assign this pseudo-document the label y^i.
- For each opposing feature in $OF(x^i)$, we create one pseudo-document containing only one phrase corresponding to an opposing feature, weight the opposing feature by w_o and assign this pseudo-document the label $\neg y^i$, where $\neg y^i$ is the opposite class label.
- We create one pseudo-document, d' which is same as the original document, except the supporting and opposing features are removed, the remaining features are weighted by $w_{d'}^{y}$, and label y^i is assigned to this pseudo-document.
- We create another pseudo-document, d' which is same as the original document, except the supporting and opposing features are removed, the remaining features are weighted by $w_{d'}^{\neg y}$, and label $\neg y^i$ is assigned to this pseudo-document.

We incorporate these pseudo-documents into $\mathcal{L}$, on which the classifier is trained. We call this approach to incorporate explanations as *learning with explanations* (LwE).

We present a sample dataset with two documents, a positive movie review and a negative movie review, below. In these documents, the words that are returned as supporting features are underlined and the words that are provided as opposing features are in strikethrough.

Document 1: This is a ~~***weird***~~ *low-budget movie. It is* ~~***awful***~~ *but it pulls off somehow, that is why I* ***love*** *it.*

Document 2: This movie had ~~***great***~~ *acting,* ~~***good***~~ *photography, but the plot was* ***terrible****. Ultimately it was a* ***failure****.*

As this example illustrates, there are supporting positive (negative) words and opposing negative (positive) words in a positive (negative) document. Table 1 shows the traditional binary representation and LwE representation for *Document 2*.

One would expect that the weights for documents that contain only the explanations (w_o and w_s) would be higher than the ones that exclude explanations ($w_{d'}^{y}$ and $w_{d'}^{\neg y}$), to emphasize the supporting features for the chosen label and opposing

[1] If the features corresponding to the explanations do not exist in the dictionary, the dictionary can be expanded to include the new phrases, e.g. by creating and adding the corresponding n-grams to the dictionary.

Table 1: The binary representation (top) and its LwE transformation (bottom) for Document 2 (D2). Stop words are removed. LwE creates multiple pseudo-documents with various feature weights and class labels.

	movie	great	acting	good	photography	terrible	plot	ultimately	failure	**label**
				Binary representation						
D2	1	1	1	1	1	1	1	1	1	−
				LwE transformation of the binary representation						
$D2_1$						w_s				−
$D2_2$									w_s	−
$D2_3$		w_o								+
$D2_4$				w_o						+
$D2_5$	$w^y_{d'}$		$w^y_{d'}$		$w^y_{d'}$		$w^y_{d'}$	$w^y_{d'}$		−
$D2_6$	$w^{\neg y}_{d'}$		$w^{\neg y}_{d'}$		$w^{\neg y}_{d'}$		$w^{\neg y}_{d'}$	$w^{\neg y}_{d'}$		+

Table 2: Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary.

Dataset	Task	Train	Test	Vocab.
IMDB	Sentiment analysis of movie reviews [Maas *et al.*, 2011]	25,000	25,000	27,272
NOVA	Email classification (politics vs. religion) [Guyon, 2011]	12,977	6,498	16,969
WvsH	20Newsgroups (Windows vs. Hardware)	1,176	783	4,026

features for the opposite label, and de-emphasize the remaining phrases in both classes. Moreover, the documents that exclude explanations would be weighted higher for the chosen label, $w^y_{d'}$, than for the opposite label, $w^{\neg y}_{d'}$, since even without the supporting features, the document would more likely belong to class y than to class $\neg y$. This is because the document is overall labeled as y and the labeler is not necessarily asked to provide all the explanations for classification.

This approach to incorporate explanations is not tied to any classifier. Any off-the-shelf classifier that can work with numerical features, such as multinomial naïve Bayes (for which all the feature weights, w_s, w_o, $w^y_{d'}$, and $w^{\neg y}_{d'}$, must be non-negative), logistic regression, and support vector machines, can be used as the underlying model.

4 Experimental Methodology and Results

In this section we first describe the settings, datasets, and classifiers used for our experiments and how we simulated a human labeler to provide explanations for document classification. Then, we present the results comparing *traditional learning* (TL), *learning with rationales* (LwR), and *learning with explanations* (LwE). We use LwR strategy presented in [Sharma *et al.*, 2015] as a baseline for our LwE approach.

4.1 Methodology

We experimented with three document classification datasets, which are described in Table 2. We evaluated our strategy using multinomial naïve Bayes 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.

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. Iteratively, 10 documents were chosen at random and were annotated by TL, LwR, and LwE approaches. This process was repeated 10 times, and average learning curves over 10 different runs are presented. We evaluated all the strategies using AUC (Area Under an ROC Curve) measure. For this study we selected documents randomly, as opposed to using active learning approaches such as uncertainty sampling [Lewis and Gale, 1994], to run a controlled experiment where TL, LwR, and LwE, all operated on the same set of documents.

The LwR approach [Sharma *et al.*, 2015] elicits rationales for classification, and modifies the document to weight rationale features higher than other features within that document. On the other hand, LwE approach elicits explanations, where supporting features are rationales for classification, but it goes one step further than LwR, and elicits opposing features. For each document, the rationales provided for LwR by the simulated labeler are the same as supporting features provided for LwE, so compared to LwR, LwE has the additional advantage of receiving opposing features from the labeler. However, we cannot argue that the difference between LwR and LwE strategies is only due to eliciting opposing features, since LwE creates several pseudo-documents for explanations, whereas LwR re-weights features within a document.

For LwR baseline, we used the same weights that were used in [Sharma *et al.*, 2015]. That is, we set the weights for rationales and the remaining features of a document to 1 and 0.01 respectively (i.e. $r = 1$ and $o = 0.01$). For LwE using multinomial naïve Bayes, we set the weights $w_s = 100$, $w_o = 100$, $w^y_{d'} = 1$, and $w^{\neg y}_{d'} = 0.01$. For LwE using support vector machines, we set the weights $w_s = 1$, $w_o = 1$, $w^y_{d'} = 0.1$, and $w^{\neg y}_{d'} = 0.001$. These weights worked reasonably well for all three datasets. We experimented with fixed weight settings in this section to show that LwE can do well across datasets even without parameter tuning. In Section 4.3, we also present results using the best possible parameter settings for all approaches.

We simulated the human labeler in the same way as [Sharma *et al.*, 2015]. The simulated labeler recognized phrases as positive (negative) features that had the highest χ^2 (chi-squared) statistic in at least 5% of the positive (negative) documents. To make the labeler's effort as small as possible, we ask the labeler to highlight any one feature as supporting feature and any one feature as opposing feature, as opposed to asking the labeler to highlight all supporting and opposing features. We also allowed the labeler to skip highlighting any phrase as supporting or opposing, if the answer is not obvious, i.e. if the labeler cannot pinpoint any phrase as a supporting/opposing feature.

4.2 Results

Figure 1 presents the learning curves on three document classification datasets using multinomial naïve Bayes and support vector machines. The results show that LwE provides huge improvements over TL for all datasets and classifiers. We performed pairwise one-tailed t-tests under significance level of 0.05, where pairs are area under the learning curves for 10 runs of each method. If a method has higher average performance than a baseline with a significance level of 0.05 or better, it is a win, if it has significantly lower performance, it is a loss, and if the difference is not statistically significant, the result is a tie. For all three datasets and two classifiers, LwE statistically significantly outperforms TL. For NOVA dataset, LwE outperforms TL on the first half of the learning curve, but later loses to TL under fixed-parameter settings. As we show later in Section 4.3, under best parameter settings, LwE outperforms TL for NOVA at all budget levels. LwR also performs much better than TL, and is therefore a strong baseline for LwE, however, LwE provides further improvements over LwR. The t-test results show that for IMDB and NOVA datasets, LwE statistically significantly wins over LwR using both classifiers. For WvsH dataset, LwE wins over LwR using multinomial naïve Bayes and LwE ties with LwR using support vector machines.

It is not a surprise that LwE is able to outperform TL and LwR. LwE is asking the human to provide further information than just the labels. What we are arguing, however, is that LwE is able to integrate this extra information into the learning process effectively. In Table 3, we compare the number of annotated documents required by TL, LwR, and LwE to achieve a target AUC performance using multinomial naïve Bayes. The results using support vector machines are similar and are omitted to avoid redundancy. The ratios of number of documents required by TL and LwE (**TL/LwE**) in this table show that LwE drastically accelerates learning. For example, for IMDB dataset, in order to achieve AUC of 0.85 using multinomial naïve Bayes, TL requires labeling 233 documents, whereas LwE achieves the same AUC with just 51 labeled documents. We note that providing explanations might take more time than providing just the labels, however, for this case, if the labeler does not take more than 4.5 times the amount time in providing explanations, it is better to ask labeler to provide explanations along with labels for documents. Moreover, LwE often requires fewer labeled documents compared to LwR to achieve the same target AUC. As Table 3 shows, the ratio of number of documents required by the LwR and LwE (**LwR/LwE**) is often greater than 1 and sometimes as large as 3.2. That is, if the labeler is already providing a rationale, then if the labeler does not spend more than 3 times the amount of time in providing an opposing feature, labeling documents with explanations is worth the expert's time.

Table 3: Comparison of number of documents required to achieve a target AUC by TL, LwE, and LwR using multinomial naïve Bayes. 'n/a' represents that a target AUC cannot be achieved by a method.

	Target AUC	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
IMDB	TL	37	65	106	233	841	n/a
	LwR	10	16	37	164	n/a	n/a
	LwE	5	9	18	51	379	n/a
	TL/LwE	**7.4**	**7.22**	**5.89**	**4.57**	**2.22**	**n/a**
	LwR/LwE	**2**	**1.78**	**2.06**	**3.22**	**n/a**	**n/a**
NOVA	TL	3	3	5	12	28	126
	LwR	2	3	4	11	31	110
	LwE	2	3	4	9	16	51
	TL/LwE	**1.5**	**1**	**1.25**	**1.33**	**1.75**	**2.47**
	LwR/LwE	**1**	**1**	**1**	**1.22**	**1.94**	**2.16**
WvsH	TL	17	33	57	127	380	n/a
	LwR	4	6	12	33	188	n/a
	LwE	4	6	12	30	146	n/a
	TL/LwE	**4.25**	**5.5**	**4.75**	**4.23**	**2.6**	**n/a**
	LwR/LwE	**1**	**1**	**1**	**1.1**	**1.29**	**n/a**

4.3 LwE vs. TL and LwR under Best Parameter Settings

So far we have seen that LwE provides improvements over TL and LwR. All three strategies used a fixed weight setting for hyper-parameters. In this section, we examine how TL, LwE, and LwR methods would behave when they are tuned using the best parameter settings. To find out, we searched over several parameters, optimizing on the test data. Note that, normally, one would never optimize over the test data in practical settings. This is a hypothetical setting, and the purpose is to conduct a controlled experiment to tease out whether the LwE framework is different, better, or worse than the LwR framework, when both are tuned using best possible parameter settings.

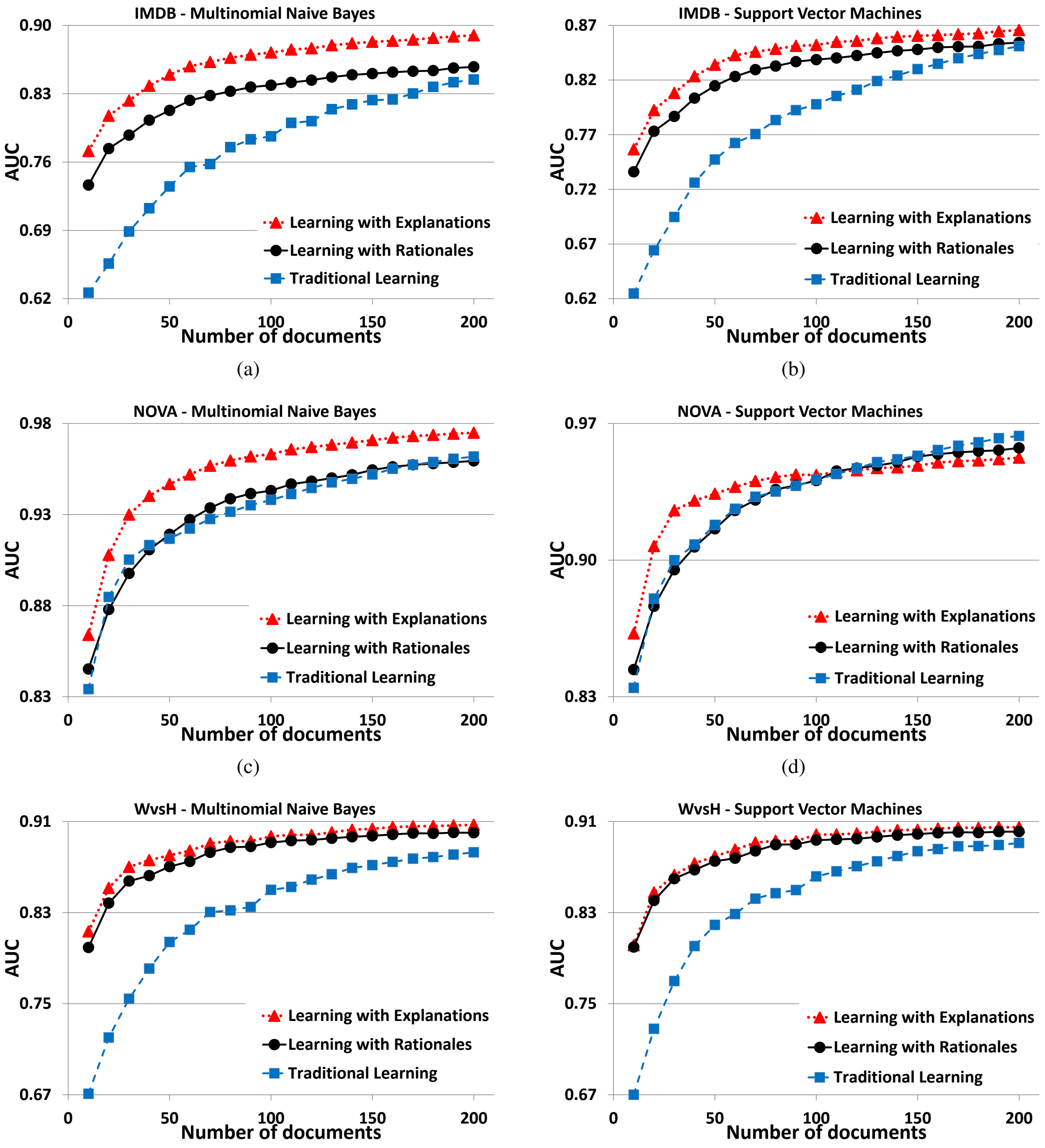
For LwR, we searched for weights r and o and for LwE, we searched for weights w_s, w_o, $w_{d'}^{y}$, and $w_{d'}^{\neg y}$. In addition to these parameters, for multinomial naïve Bayes, we searched for the smoothing parameter, α, and for support vector machines, we searched for the regularization parameter, C. For all hyper-parameters, we performed grid search for values between 10^{-3} and 10^{3}. TL searches for only α and C.

Figure 2 presents learning curves comparing LwE to TL and LwR under best parameter settings. The t-tests results show that LwE statistically significantly wins over TL for all three datasets. For IMDB and NOVA datasets, LwE wins over LwR using both classifiers. For WvsH dataset, LwE ties with LwR using both classifiers.

5 Limitations and Future Work

A limitation of our work is that we simulated the labeler in our experiments. We asked the labeler to provide any one supporting and any one opposing feature to explain the classification. However, a user study is needed to experiment with potentially noisy labelers, and measure how much performance and efficiency improvements learning with explanations provides over traditional learning and over learning with rationales.

We used fixed weight settings and best weight settings for the hyper-parameters. Ideally, one should search for optimal hyper-parameters using cross validation on training set. A



(a) (b)

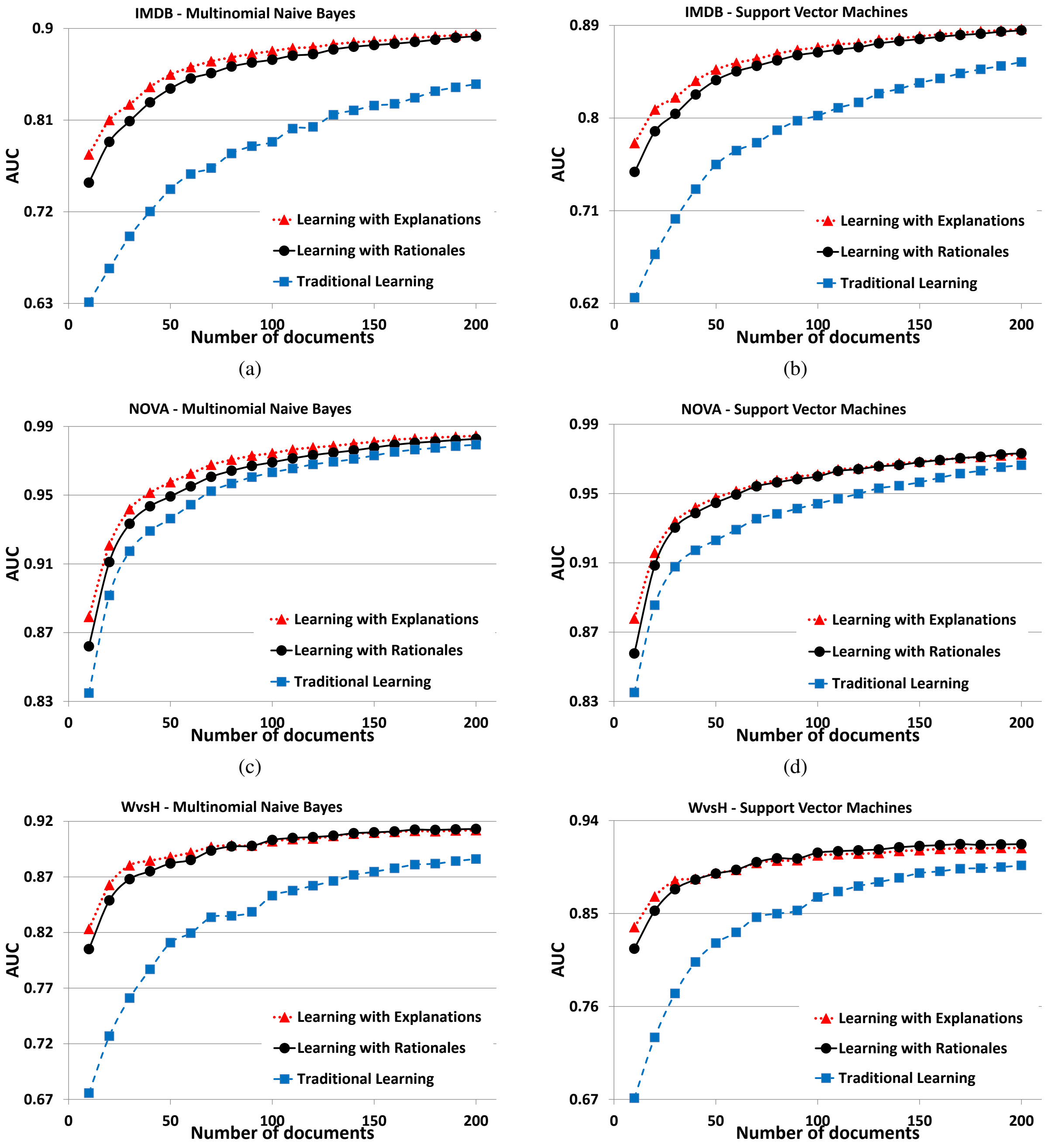
(c) (d)

(e) (f)

Figure 1: Comparison of LwE to TL and LwR. LwE provides significant improvements over TL. LwE statistically significantly wins over LwR for (a), (b), (c), (d), and (e). LwE ties with LwR on WvsH dataset using support vector machines (f).

challenge is that both LwE and LwR provide further benefits over TL when the labeled data is small (which makes sense because domain knowledge is invaluable especially when labeled data is small), but hyper-parameter tuning using cross-validation on the training data itself, when the training data is small, is not trivial. On the bright side, however, both LwE and LwR perform better than TL even under fixed parameter settings.

We provided a framework to incorporate explanations in the form of supporting and opposing features for classification. Another line of future work is to allow labelers to provide other types of explanations, where explanations can be complex conjunction or disjunction of domain-specific features, or free-form text entries.



(a) (b) (c) (d) (e) (f)

Figure 2: Comparison of LwE with TL and LwR under best parameter settings.

6 Conclusion

We introduced a framework to incorporate explanations into learning for text classification. Our simple strategy to incorporate feature-based explanations can utilize any off-the-shelf classifier. The empirical evaluations on three text datasets and two classifiers showed that our proposed method can incorporate simple explanations, in the form of supporting and opposing features, effectively for document classification.

Acknowledgments

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

References

[Attenberg *et al.*, 2010] Josh Attenberg, Prem Melville, and Foster Provost. 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, 2010.

[Buchanan *et al.*, 1984] Bruce G Buchanan, Edward Hance Shortliffe, et al. *Rule-based expert systems*, volume 3. Addison-Wesley Reading, MA, 1984.

[DeJong and Mooney, 1986] Gerald DeJong and Raymond Mooney. Explanation-based learning: An alternative view. *Machine learning*, 1(2):145–176, 1986.

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

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

[Ellman, 1989] Thomas Ellman. Explanation-based learning: A survey of programs and perspectives. *ACM Computing Surveys (CSUR)*, 21(2):163–221, 1989.

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

[Girosi and Chan, 1995] Federico Girosi and Nicholas Tung Chan. 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. IEEE, 1995.

[Guyon, 2011] Isabell Guyon. Results of active learning challenge, 2011.

[Laird *et al.*, 1986] John E Laird, Paul S Rosenbloom, and Allen Newell. Chunking in soar: The anatomy of a general learning mechanism. *Machine learning*, 1(1):11–46, 1986.

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

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

[Melville and Sindhwani, 2009] Prem Melville and Vikas Sindhwani. 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, 2009.

[Mitchell *et al.*, 1986] Tom M Mitchell, Richard M Keller, and Smadar T Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine learning*, 1(1):47–80, 1986.

[Mooney, 1986] Raymond J Mooney. Generalizing explanations of narratives into schemata. In *Machine Learning*, pages 207–212. Springer, 1986.

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

[Pedregosa *et al.*, 2011] 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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Raghavan and Allan, 2007] Hema Raghavan and James Allan. 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. ACM, 2007.

[Ramirez-Loaiza *et al.*, 2016] Maria E. Ramirez-Loaiza, Manali Sharma, Geet Kumar, and Mustafa Bilgic. Active learning: an empirical study of common baselines. *Data Mining and Knowledge Discovery*, pages 1–27, 2016.

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

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

[Sharma and Bilgic, 2016] Manali Sharma and Mustafa Bilgic. Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, pages 1–39, 2016.

[Sharma *et al.*, 2015] Manali Sharma, Di Zhuang, and Mustafa Bilgic. Active learning with rationales for text classification. In *NAACL HLT*, pages 441–451, 2015.

[Small *et al.*, 2011] Kevin Small, Byron Wallace, Thomas Trikalinos, and Carla E Brodley. The constrained weight space svm: learning with ranked features. In *ICML*, pages 865–872, 2011.

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

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

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

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