

Is it Over Yet?

Learning to Recognize Good News in Financial Media

Anthony Brew, Derek Greene, and Pádraig Cunningham

School of Computer Science and Informatics, University College Dublin
{anthony.brew,derek.greene,padraig.cunningham}@ucd.ie

University College Dublin
Technical Report UCD-CSI-2010-1¹
January 2010

Abstract. Until recently, tracking sentiment in news media required professional annotators to identify the polarity of individual articles so that general trends could be identified. In the work described here we use *crowdsourcing* to gather non-expert annotations, in conjunction with a supervised learning strategy that generalizes from the manual annotations to label a larger body of news articles. Our analysis of this strategy shows that, while it is effective, there are three key issues that have to be addressed: consensus, coverage, and bias. By obtaining multiple annotations for an article we can establish a *consensus* for the article. Alternatively we can seek only a single annotation for each article in order to maximize *coverage*, but without the benefit of a group consensus. With *bias*, we are not so much concerned by bias among the annotators as by the bias in the learning system which can favor the majority class. In this paper we address these three issues in the context of an analysis of media sentiment towards the Irish economic situation.

1 Introduction

We are concerned with the challenge of tracking general sentiment trends on specific topics in online content. In the demonstration system discussed here we focus on sentiment in online news sources concerning the Irish economic situation². The insights offered from such an analysis are best explained with reference to the time-plot shown in Figure 1. This plot shows aggregate sentiment from three news sources, together with a micro-average reflecting overall sentiment. For instance, we can see that RTE, the national broadcaster (indicated by top line) is more optimistic than the other sources – at least in the news feeds analyzed here. This is somewhat surprising as the Irish government frequently refer

¹ This research was supported by Science Foundation Ireland (SFI) Grant Nos. 05/IN.1/I24 and 08/SRC/I1407.

² See: <http://sentiment.ucd.ie>

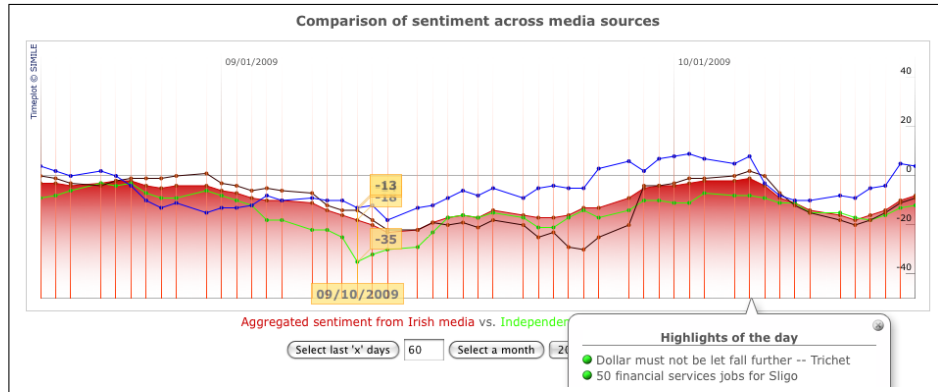


Fig. 1. A screenshot of the time-plot generated by the system, which tracks economic sentiment from the various news sources over time.

to what they claim to be unwarranted criticism from RTE. We can also see a peak towards the right of the graph which corresponds to an improvement in sentiment associated with the referendum on the EU Lisbon Treaty in early October 2009. The system also helps *decompose* sentiment by providing tag clouds of discriminating positive and negative terms, along with lists of highly positive and negative articles (Figure 2).

Rather than relying on polarity judgments from a single expert, such as an individual economist, the strategy adopted in this system is to generate trend statistics by collecting annotations from a number of non-expert users. These annotations are then used to train a classifier to automatically label a much larger set of news articles. It is worth emphasizing that the annotators are volunteers, so we are not dealing with crowdsourcing in the micro-task markets sense (*e.g.* Amazon’s Mechanical Turk [9]), where annotators are paid for their efforts [8, 13]. The main reward for the annotators is the representation used in the annotation process itself – a *Really Simple Syndication* (RSS) feed providing a distillation of topical news stories.

The combination of machine learning and crowdsourcing has a number of advantages in the context of sentiment analysis:

- Once the classifier has been trained, a large number of unlabeled items can be classified to provide more robust statistics regarding sentiment trends.
- Statistics can be generated after the annotation process ends. The extent to which this can be done depends on the amount of *concept drift* that occurs over time in the specific domain of interest.
- Once trained, the classifier has less *variance* than individual annotators alone.

In [4] we introduced the economic sentiment analysis system and described a strategy for managing the interaction with the annotators. In this work we dis-

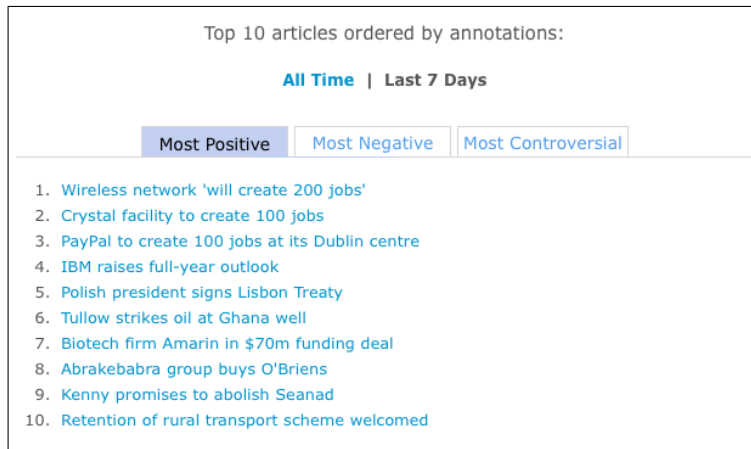


Fig. 2. The system provides lists of the most positive, negative, and controversial articles – both over a seven day window and over the lifetime of the system.

cuss and address three core issues that arise when integrating crowdsourcing and machine learning techniques for sentiment analysis:

- The first is the issue of identifying *consensus* on a label among the non-expert annotators.
- The second issue relates to the *coverage* of the training data. It is important that the training data for the classifier should cover as much of the domain as possible. However, given the effort required to manually annotate items, there is likely to be a trade-off between coverage and consensus.
- Given that the main objective of the proposed system is to generate plots of the type shown in Figure 1, it is important that the classifier should not be *biased*. In Section 6 we show that nearest neighbor, naïve Bayes, and Support Vector Machine (SVM) classifiers are biased toward the majority class in our task.

These issues are examined in the context of a new real-world dataset of news article annotations, described in more detail in Section 3.1.

The remainder of the paper is structured as follows. In the next section we provide an overview of research related to our task. In Section 3 we describe the task in more detail, and outline our strategy for integrating crowdsourcing and supervised learning. Further detail on the approach for selecting articles for annotation is given in Section 4. Then in Section 4 the impact of the annotation selection policy is demonstrated. In Section 5 the trade-off between annotation consensus and coverage is discussed. In Section 6 the problem of classifier bias is discussed, and a novel solution is presented.

2 Related Work

The general problem of detecting the polarity (positive or negative) of opinions in online content has recently become an area of particular interest for researchers in the natural language processing and machine learning communities. Common approaches have included the identification of authors' attitudes based on applying standard text classification techniques to document bag-of-words representations [11], searching for opinion-carrying terms in documents [1], and frequent pattern mining to identify syntactic relations between sequences of terms that may be indicative of sentiment polarity [10]. Most frequently these techniques have been applied to tasks such as classifying movie reviews [11] or product reviews [3] based on the polarity of review text.

Traditionally, datasets for sentiment analysis tasks have been manually constructed by small groups of expert annotators with specific training (*e.g.* the MPQA corpus [17]). While this approach to annotating sentiment in text corpora can provide detailed, high-quality data, it will often be infeasible in real-world tasks due to time constraints or lack of access to domain experts. As an alternative, services such as Mechanical Turk [9] have demonstrated the utility of harnessing crowds of non-expert users to perform time-consuming labeling tasks. There is already a significant research literature on the problem of aggregating a number of medium quality annotations in order to generate a good quality annotation. Two important early contributions in this area are the work of Dawid and Skene [6] and the work of Smyth *et al.* [15]. Recently there has been renewed interest in the area with the advent of crowdsourcing as a fast and effective mechanism of generating medium quality annotations [16, 8, 7, 13]. A key question in this area relates to the importance placed on data quality. Snow *et al.* show that, for text annotation tasks similar to that addressed in our work, crowdsourced annotators are not as effective individually as experts. But when non-expert opinions are aggregated together, it is possible to produce high-quality annotations [16]. So this work establishes the merit of aggregating a number of annotations in order to generate good quality annotations.

The question of the balance between data coverage and annotation quality arises frequently in the literature. Raykar *et al.* [13] proposed a strategy that simultaneously induces "ground truth" (or gold standard) from multiple annotations, while also building a classifier based on this labeling. The authors suggest that having effective annotators is more important than data coverage, and emphasize the use of multiple annotations for each item, in conjunction with weights for annotators based on their agreement with the induced ground truth. Smyth *et al.* [15] also highlighted the difficulty of performance evaluation in tasks where annotations are available from multiple annotators, but no ground truth is available as a reference. In such cases we must rely on annotator consensus as a proxy when measuring annotation quality.

3 Task Description

The primary objective of our task is to produce unbiased assessments of sentiment in a dynamic collection of news articles, so that trends and differences between sources can be identified and visualized as shown in Figure 1. In the system implementation, articles are collected from a pre-defined set of RSS feed URLs published by the news sources of interest. After applying a relevance classifier, most articles not pertaining to economic news are filtered from the candidate set. From the remaining relevant articles, a subset is chosen based on an appropriate article selection mechanism. The resulting subset of articles is then presented via an RSS feed to the annotators, who are encouraged to label the articles as *positive*, *negative*, or *irrelevant*. These annotations are subsequently used to retrain the classification algorithms on a daily basis.

The main components of the system are outlined in Figure 3. The selection of articles for annotation takes place at (A), and the polarity classification and bias correction happens at (B). Given that there is a large collection of articles to be annotated (either manually or by the classifier), the article selection policy for manual annotation has a considerable impact on the overall annotation quality. This issue is discussed in detail in Section 4, while a solution for bias correction is proposed in Section 6.

3.1 Irish Economy Dataset

Using the system outlined in Figure 4, we retrieved articles from three online news sources (RTE, The Irish Times, The Irish Independent) during a three month period (July to October 2009). A subset of these were annotated on a daily basis by a group of 33 volunteer users. The first month constituted a “warm-up” period, which allowed us to train the relevance classifier to a point where it achieves approximately 90% accuracy. This provided an initial dataset containing 3858 articles, with 2693 user annotations covering 354 individual articles. For the latter two months of the experiment, we collected a dataset for evaluating the machine learning questions arising from the sentiment analysis task. This second “main” dataset comprises 12469 documents, with 6910 user annotations resulting in 1306 labeled articles. Both datasets have been made available online³ for further research. Unless otherwise stated, for the remainder of this paper our experiments focus on the positive vs. negative classification problem as applied to the “main” dataset.

3.2 Baseline Classification

For the classification components of the system, we considered three supervised learning techniques that have previously been effective in text classification tasks [5]. These are naïve Bayes, SVMs, and k -nearest neighbor (k -NN). In order to select the classifier that was best suited to our task, we performed a baseline

³ See <http://mlg.ucd.ie/sentiment>

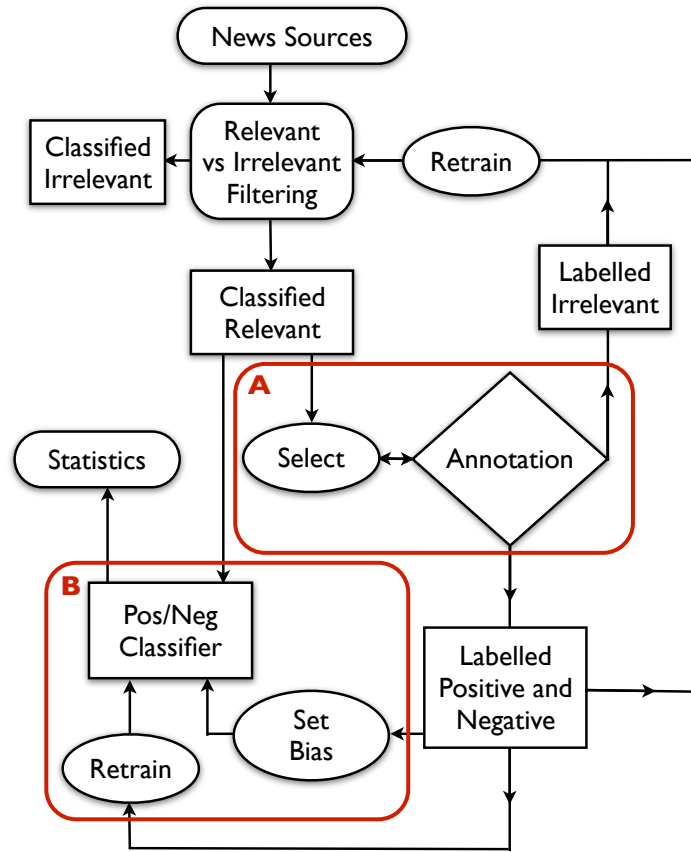


Fig. 3. Overall design of the economic sentiment analysis system. For the purpose of this paper, the important components are (A) the article selection and annotation process, and (B) the training of the classifier where classification bias is controlled.

assessment using cross-validation. In all cases we follow Pang *et al.* [11] who suggested the use of unigram bag-of-words features to represent documents, although we do make use of term frequency information rather than merely looking at the presence or absence of terms.

The results of the evaluation are shown in Table 1. Accuracy figures are reported for each of the three classification techniques on two different tasks (positive vs. negative and relevant vs. irrelevant). We also report AUC (area under the ROC curve) figures [12], as these consider classifier performance across a range of thresholds and are thus independent of bias considerations.

These results corroborate the previous findings in [11], which showed that SVMs tend to only marginally out-perform naïve Bayes in sentiment classification tasks. The k -NN classifier does not do well in the evaluation and we do not consider

Measure	<i>Positive vs. Negative</i>			<i>Relevant vs Irrelevant</i>		
	Bayes SVM	<i>k</i> -NN		Bayes SVM	<i>k</i> -NN	
AUC	0.80	0.82	<i>0.71</i>	0.90	0.88	<i>0.68</i>
Accuracy	75%	77%	72%	85%	81%	76%

Table 1. Baseline accuracies show that Naïve Bayes marginally outperforms the SVM on the task of classifying between relevant and irrelevant articles while SVM marginally outperforms Naïve Bayes in the Positive versus Negative task.

it further in this work. The Bayes classifier performs best on the relevance task and is competitive on the positive vs. negative task. In addition, since many of our experiments here involve active learning-style scenarios, algorithm time complexity is an important consideration. In this respect the linear training time of naïve Bayes is preferable to the cubic training time of SVMs. Another important consideration is the fact that the Bayes classifier is easier to update than the SVM because the SVM is sensitive to parameter selection. For these reasons we employed a naïve Bayes classifier in our sentiment analysis system.

4 Article Selection

As described previously, only a fraction of all articles retrieved from the news sources will be presented to the users for manual annotation. A natural question arises as to how an appropriate subset of articles should be chosen on a given day – this corresponds to component (A) in Figure 4. In some respects this problem resembles the task of query selection in active learning, where the goal is to select the most informative unlabeled items to present to the oracle. However, another goal to consider in the context of crowdsourced annotation is the selection of a diverse set of items that the users will be willing to annotate. In our particular task we wished to incentivize users to annotate articles by providing them with a useful summary of the day’s economic news stories, delivered in the form of an RSS feed. Ideally from a user’s perspective we would like to avoid the publication of a feed containing duplicate or highly-similar articles which fail to summarize the news of the day.

To identify a diverse set of articles that provides a representative summary of the day’s economic news, we apply a clustering-based article selection strategy. Firstly we construct k clusters of articles by merging all pairs of articles with cosine similarity above a threshold $\tau \in [0, 1]$. This is equivalent to applying complete-linkage agglomerative clustering and choosing a merging cut-off threshold τ . From the set of clusters, we then choose a subset $k' < k$ using weighted farthest-first traversal [2]. This leads to the selection of a sufficiently diverse set of clusters, while also ensuring that large clusters (*i.e.* representing dominant news stories for a particular day) are likely to be selected. The most representative article from each of the k' clusters (*i.e.* most similar to the cluster centroid) is then selected for annotation. In practice we found that approximately $k' = 10$

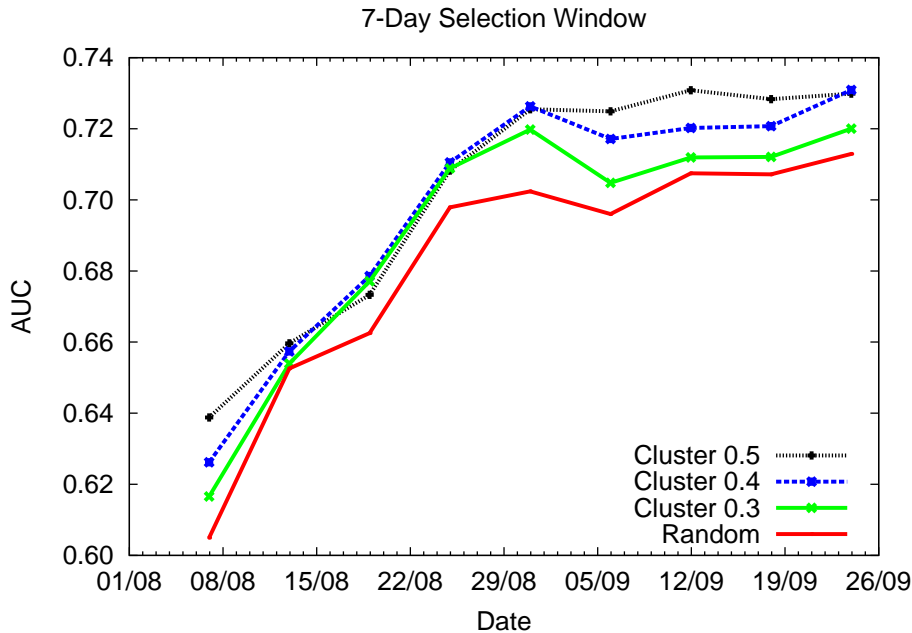


Fig. 4. Comparison of article selection performed with a clustering-based strategy versus random selection (averaged over multiple runs).

articles was a reasonable number of articles to present to users for annotation each day.

To examine the utility of the proposed selection strategy, we compare the strategy for different values of τ with a baseline random selection strategy. While it is not possible to evaluate the strategy on a daily basis since we only have ≈ 10 labeled articles per day, we can approximate the daily selection process by applying article selection to a set of labeled articles from a window of 7 consecutive days. Figure 4 shows the AUC performance of the clustering strategy ($\tau \in [0.3, 0.5]$) as 10 additional articles are selected from each 7 day window and added to the classifier’s training set. The clustering strategy out-performs random selection for all τ parameter values tested, with the best AUC scores achieved by $\tau = 0.5$ which corresponds to a more conservative agglomeration of articles.

We observe that, when values of $\tau \geq 0.6$ are used, many singleton clusters are produced, even in cases where articles cover the same news story. In practice a selection of a conservative threshold $\tau \approx 0.5$ for our task has the effect that articles on distinct stories are not treated as being identical, while highly-similar articles, reporting on the same news story, are grouped together so that only the most representative article is presented to the annotators. This ensures that the article selection strategy is not only beneficial for the subsequent training phase in terms of covering as much of the domain as possible (discussed further in the

next section), but also ensures the selection of a diverse set of reading material for the annotators.

5 Consensus versus Coverage

The development of crowdsourcing has provided a fast and effective mechanism for obtaining annotations [8, 7, 13, 16]. When such non-expert opinions are combined together, it may be possible to produce higher quality aggregated annotations [16]. While this facility is relevant to our task, there is one important difference. In our task good quality annotations are not an end in themselves. Rather we require a body of annotated data that can be used to subsequently train a classifier. Given the role of the classifier in the overall sentiment analysis system, a natural question arises – is it better to use the annotation “budget” to produce consensus judgments, or should annotator effort be spread out across as many items as possible to provide better coverage of the domain? Which is preferable – 300 single annotations on 300 items, or 60 annotations based on 5 annotations per item?

In the remainder of this section we develop a policy for obtaining multiple annotations for articles, which considers the trade-off between these two important considerations:

- **Consensus:** How does the agreement between the annotators, or lack thereof, affect the performance of the classifier?
- **Coverage:** If we have a limited annotation budget, how can we make the most effective use of this budget to adequately cover the domain?

5.1 Consensus

Previously Sheng *et al.* [14] showed that, as annotator quality diminishes, the need to acquire additional annotations increases. The authors manually added different levels of noise to the data to simulate annotators of varying degrees of quality, thereby demonstrating the importance of annotation quality in the training of the learning system. In real-world annotation scenarios, this kind of noise could potentially arise due to lack of annotator expertise, interest or familiarity with the annotation procedure.

The notion of annotation quality needs to be treated with care in the context of our task. In some tasks, aggregated high-quality annotations will always correspond to the “correct answer”. Whereas in the context of the Irish economy dataset, the answer is likely to be far more subjective. Indeed expert economists or political scientists might have strongly divergent opinions regarding the topics discussed in many of the news articles. Similarly, when our non-experts disagree with the majority on the polarity of a particular article, this does not imply that their annotation is incorrect, but rather that they potentially have a different viewpoint on the article. These individual opinions can be naturally aggregated

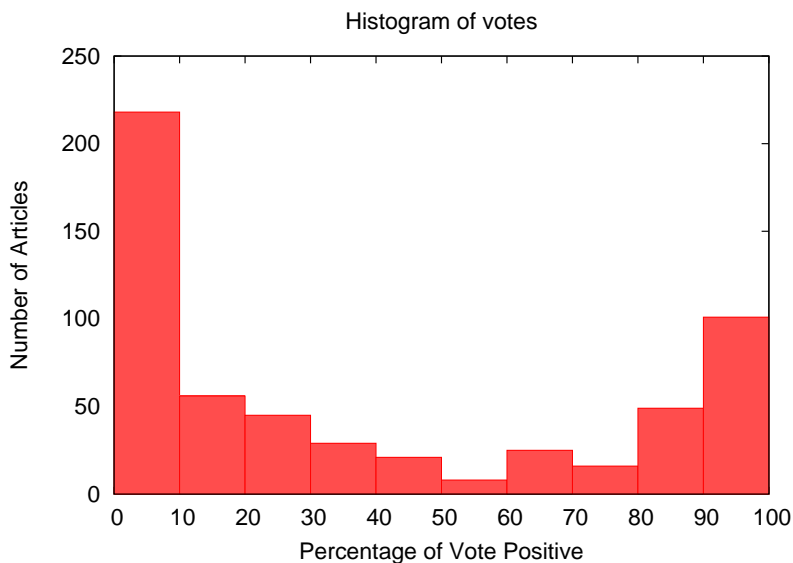


Fig. 5. Histogram showing the range of agreement in the dataset on articles that had 5 or more annotations.

as votes to get a majority consensus opinions on an article. But recall that our ultimate objective is to track aggregate trends across large collections of articles.

At this point it is worth formally defining *consensus* – it represents the margin by which users agree on the polarity of an article. A simple quantitative measure for the degree of consensus on a single article is given by:

$$\text{consensus} = \left| \frac{\text{votes positive} - \text{votes negative}}{\text{votes positive} + \text{votes negative}} \right|$$

An illustration of the variation in consensus scores across articles in the Irish economy data can be seen in Figure 5. While it is clear that there exists strong agreement between annotators on many articles (the leftmost and rightmost bars), a significant proportion of articles (45%) do not achieve 100% consensus.

To measure the impact of consensus on the classifier used in our system, we selected a set of 350 articles that had five or more annotations. Articles were then labeled according to their majority vote, and then separated into positive and negative sets. Note that the 350 articles were selected to ensure these sets were balanced in size. Bayes classifiers were trained by presenting this data using two different ordering policies: presenting the articles from low consensus to high consensus (“Weak to Strong”), and vice versa (“Strong to Weak”). At each step, an article was added from both the positive and negative sets to ensure the classifier remained balanced. This experiment was run in a 10 fold cross validation setup and repeated 100 times with random shuffling to eliminate any effects arising from data ordering.

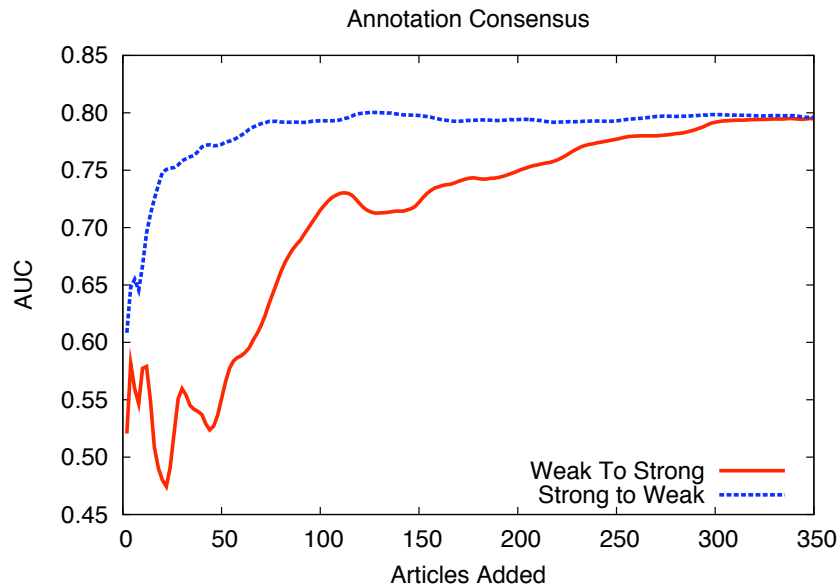


Fig. 6. Learning curves where articles were presented using two different ordering policies. In the “Weak to Strong” policy, articles with lower consensus were added first, while in “Strong to Weak”, articles with higher consensus were added first.

The results from this experiment, in terms of AUC scores, are shown in Figure 6. The difference between the two ordering policies is quite marked. It is clear that articles with a high level of consensus are very beneficial for training the classifier, while articles on which consensus is low tend to be far less useful to the classifier. Indeed there is some evidence that the learning process would be better off without them.

5.2 Coverage

If we consider annotation effort as a limited resource, then the management of the budget for annotation is an important consideration. Specifically we must address the trade-off between coverage and consensus. Ideally we could cover as much of the unlabeled data as possible in the training set. But if we only collect a single non-expert annotation for an article, we have no notion of consensus confidence for that article.

To examine the impact of coverage at the expense of consensus, we performed an evaluation where we use training data with labels chosen using different budgeting strategies: single annotation votes, best of three, and best of five votes. To avoid unnecessary spending of the budget, additional votes are not sought for an article if a clear majority has already been achieved. For this reason the alternative budgeting strategies are referred to as “First to 1”, “First to 2” and “First to 3” in Figure 7. This experiment entailed the same 100 times 10-fold cross

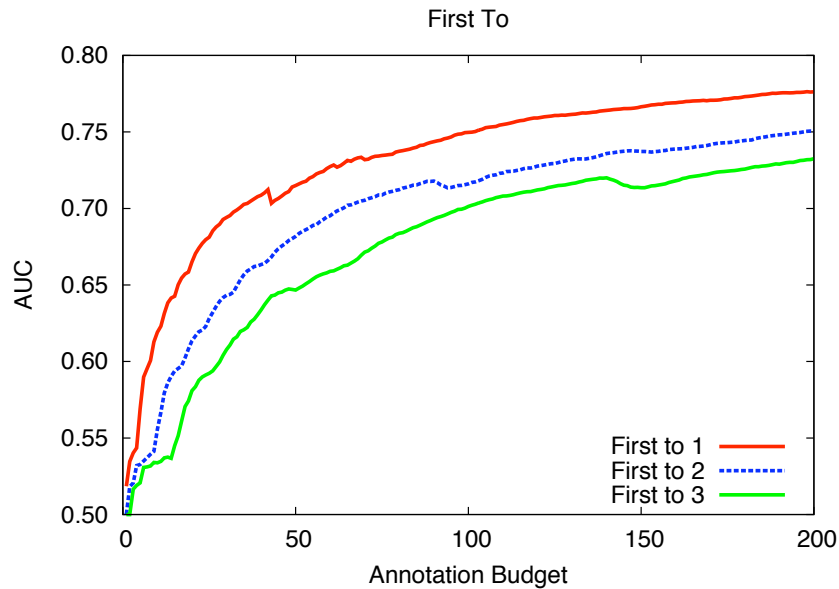


Fig. 7. This graph shows how the learning curve improves based on different strategies for spending the annotation budget. The higher coverage afforded by the “First to 1” strategy proves most effective.

validation setup as before. Note that, for each run, annotations were randomly selected from those available for each article.

The results in Figure 7 suggest that, for this annotation experiment, coverage is more important than consensus opinion on articles. Labeling a large volume of articles here proved more important than obtaining multiple votes on individual articles. It is worth emphasizing that this is the situation when learning performance is graphed against annotation effort. If we plot performance against the number of annotated articles (ignoring annotation cost) the order is reversed and the consensus annotations do best.

In the next section we show that coverage in the training data is not always more important than consensus. The balance will depend crucially on the level of disagreement between the users participating in the annotation system. We also show that often not all annotators will contribute equally to the system – some will be more useful than others.

5.3 Consensus and Coverage

In the evaluation performed in the previous section, we observed that coverage was particularly effective in improving the classifier on the data in question because in 89% of the cases, individual annotators agreed with the majority label. To examine what happens in situations where the consensus is lower, we followed the approach described in [14] by adding noise to the training data. To do

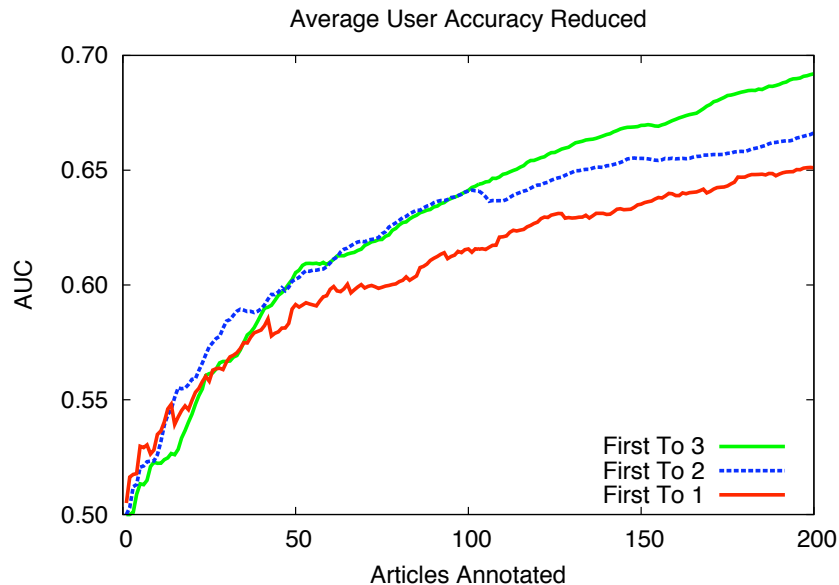


Fig. 8. When reducing average annotator-majority agreement to 67%, coverage is still important in the early stages of the learning curve. By acquiring more annotations per article, consensus can be attained and, if a sufficient budget is available, classification performance can be improved.

this 25% of the annotations were flipped in order to decrease annotator-majority agreement to approximately 67%. When the evaluation shown in Figure 7 is repeated with this data the results are quite different, as shown in Figure 8. Early in the learning process we see that coverage is still important as the classifier benefits from a breadth of examples. However, as learning continues, the availability of more reliable consensus labels assigned to articles by requesting a 2nd and 3rd opinion becomes more important. In contrast, the simple strategy of attaining maximum coverage begins to become less competitive. The evaluations shown in Figure 7 and Figure 8 demonstrate that the management of the annotation budget will often depend on the level of agreement among annotators. In our specific annotation task, the average level of inter-annotator agreement is high, and therefore it is less important to spend annotation effort to obtain consensus opinions.

The final issues we consider in this section concerns the variation between annotators. From the evaluation presented in Section 5.2, one might conclude that having just one user annotate all articles would be sufficient. However, when we look at the extent to which individual users agree with the consensus label, the level of agreement ranges from around 75% to 95%. Donmez *et al.* [7] recognized the importance of using strong annotators, and described a strategy for identifying weaker annotators. These weaker annotators were then removed from the annotation process in order to make best use of the annotation bud-

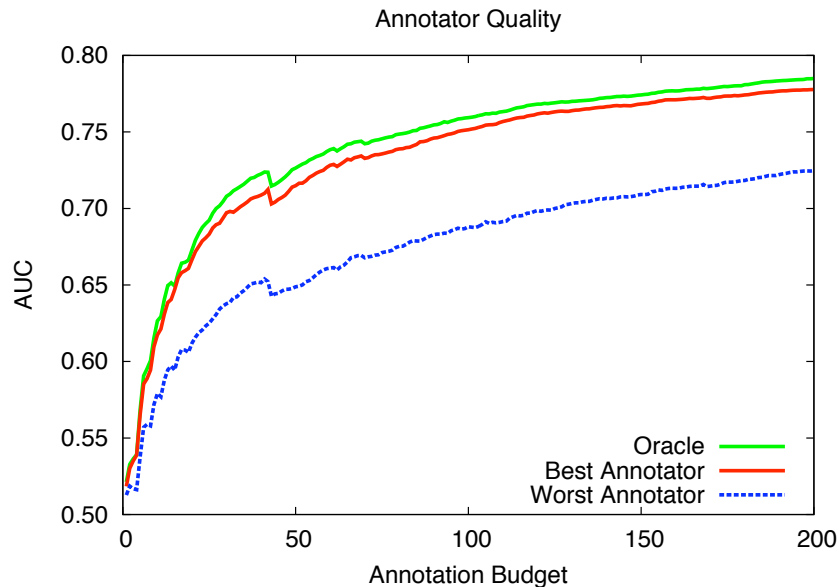


Fig. 9. This figure demonstrates the benefits of strong annotators over weak annotators. Training with annotations that use the strongest annotator only is almost as effective as training on the majority consensus annotations – the “Oracle” in this graph.

get. To demonstrate the need for care when selecting annotators, we repeat the experiment shown in Section 5.2 for the single annotation strategy. However, rather than randomly selecting annotations for each article from those available in the data, we select annotations from the “strongest” annotator (*i.e.* user having the highest agreement with the consensus opinion) and “weakest” annotator (*i.e.* user having the lowest agreement with the consensus opinion).

The difference in AUC performance shown in Figure 9 clearly highlights the benefit of using strong annotators. It is interesting to note that the best single annotator is almost as good as using the consensus judgment (referred to as the “Oracle”) to train the system. These results motivate an effective way of managing the annotation budget. Once annotators that are close to the consensus opinion have been identified, other less informative annotators can be dropped from the process with little or no deterioration in classifier performance.

6 Managing Bias

A common problem with classification algorithms is that they can become biased towards the dominant class. To some extent this is inevitable as a bias towards the majority class may minimize overall prediction error. However, the direction of this error is important when the learning task involves producing trend statistics, such as those shown in Figure 1.

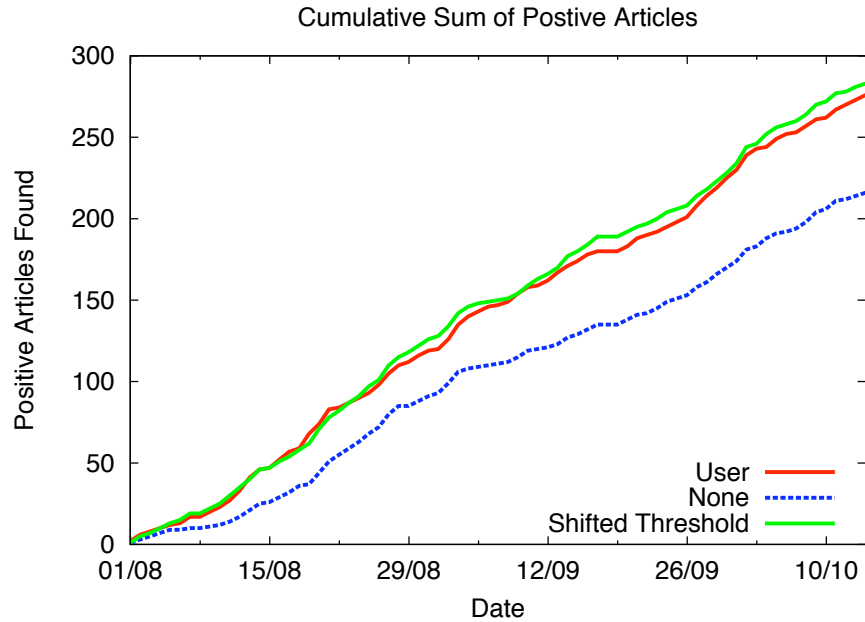


Fig. 10. This cumulative distribution of positive annotations shows how the naïve Bayes classifier is biased against the majority class unless corrected.

In general, the error of a classification scheme can be split into *bias* and *variance* components:

- The *bias* of a learner is the difference between its predicted label and the real label. In our scenario a system that outputs negative labels more often than is justified has a negative bias.
- The *variance* refers to variation in predictions. If a committee of annotators disagree on a prediction then variance is high. If learning systems trained on different subsets of the training data disagree then the learning process has *high variance*.

If we allow our classifiers to be biased then trend statistics will favor one class, causing an overly positive or negative trend to be predicted. On the other hand, by having a classifier where the dominant component of error is variance, we can expect that, while incorrect predictions are made, these errors cancel each other out, and the aggregate statistics will be reliable.

To demonstrate how classifiers can be biased, we trained the three classifiers mentioned in Table 1 on a set of 796 articles from the “main” dataset described in Section 3.1, which have been annotated by users as either positive or negative. The label for each article was determined by majority vote. We trained and tested in a 10 fold cross validation setup allowing k -NN and SVM to optimize parameters for each of the 10 runs (without allowing access to the test set).

Label	Bayes		SVM		KNN		Actual %
	+	-	+	-	+	-	
+	22.2	13.6	22.9	12.9	20.4	15.5	35.8
-	11.1	53.1	9.8	54.4	12.3	51.9	64.2
%	33.3	66.7	32.7	67.3	32.7	67.3	

Table 2. The contingency tables for classifiers tested using cross validation on a set of 796 articles. 64.2% of training data is labeled negative, but 66.7% to 67.3% of the classifier predictions are negative.

From the results of this experiment we have produced contingency tables for the classifiers, shown in Table 2. In this dataset 35.6% of the samples are actually positive. However, the classifiers only predict 33.3%, 32.7% and 32.7% positive for naïve Bayes, SVN and k -NN respectively. So if one of these classifiers was used to produce the timeplot in Figure 1, the predictions would be out 3 to 4 points, which represents a significant amount of bias. The graph in Figure 10 provides another perspective on this problem. It shows the cumulative sums of positive articles as annotated by the users and as predicted by a naïve Bayes classifier without bias correction (labeled None). It is clear that the uncorrected classifier under-predicts the number of positive articles.

6.1 Bias Correction

We now propose a simple method for removing bias from the system. A binary classifier can be viewed as an algorithm that, given an example to classify, produces a *score* that is then compared against a threshold. If the score is greater than the threshold, we assign the item to the positive class otherwise it is assigned to the negative class. For a naïve Bayes classifier the threshold will default to 0.5 – the bias can be adjusted by moving the threshold.

A better threshold can be estimated using cross validation as follows. At each iteration we build a classifier using $N - 1$ folds and find a *score* for each item in the hold-out fold. In each fold we know exactly how many positive and negative items exist, so we shift the threshold to predict the correct proportions for that fold. The classifier incorporating this threshold may still not be very accurate (due to high variance) but at least it has low bias. The unbiased threshold for all the training data is taken to be the average of the unbiased thresholds across the folds.

In the sentiment analysis system as it operates at the moment, the classifier is retrained each day and the threshold is selected using this cross validation methodology. For instance on 2nd August a classifier is trained on all labeled data up to 1st of August and the ‘unbiased’ threshold is selected using all the training data. In the future we will experiment with setting the threshold using a window of the data only.

Figure 10 shows a cumulative sum of predictions from a ‘bias-corrected’ classifier. It is clear that, when compared with the uncorrected classifier, the bias correction procedure has the desired effect. It is important to note to allow bias

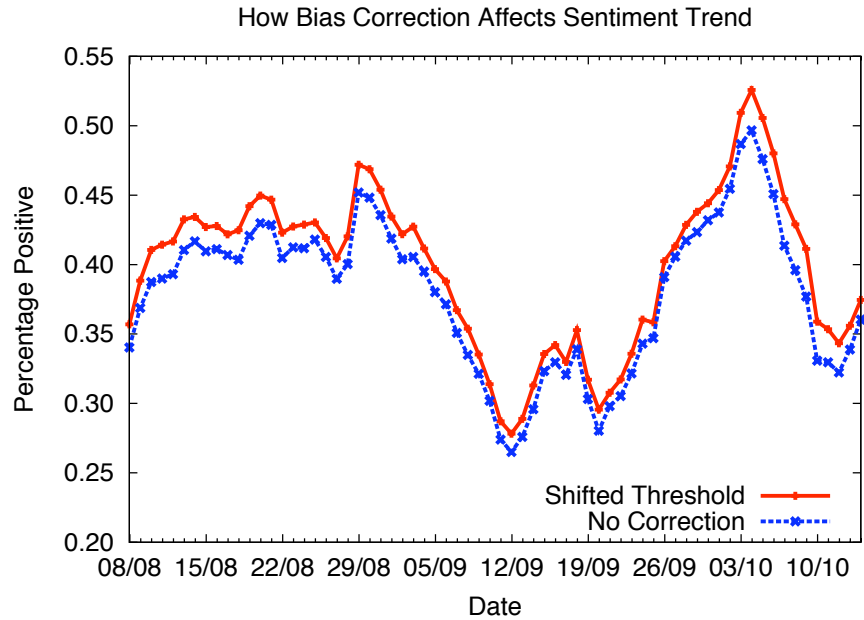


Fig. 11. Time-plots produced with and without bias-correction.

correction to start from the first day the classifiers training was augmented with data from the “warm up” dataset described in Section 3.1 as a minimum number of articles was required to find the threshold crossover point in each fold.

The impact of this in an operational setting is illustrated in Figure 11. This figure shows time-plots similar to those in the system screenshot given previously in Figure 1. It is clear that the time-plot without bias correction is overly pessimistic compared to that produced with bias correction. The “warm-up” data was also used in this case to ensure that the irrelevance filter was properly trained.

6.2 Concept Drift

One potential benefit of a system such as the one we have described is the potential to bootstrap the system by providing training up to a given point in time, and then allowing the classifier to operate without further training. The system would continue to produce trend time-plots such as that given in Figure 1 without any further manual annotation.

However, such a strategy is only valid if concept drift is not an issue, *i.e.* if the positive, negative, and irrelevant news concepts do not change over time. In reality it might occur that the sentiment regarding a particular news topic will change over time, and the key terms associated with that topic will change from having positive connotations to negative ones. This phenomenon occurred with

oil-related stories in the lifetime of the current system. In fact we have found that changes in skew in the data are the most influential form of concept drift. If the proportion of positive stories drops from 40% to 30% then the bias correction mechanism needs to be corrected to address that change.

To examine the effect of concept drift, in Figure 12 we show two aggregate time-plots for the period between 17th September and 14th October. The first plot shows continuous updating as normal, while the second shows the case where training stopped on 16th September. Note that the “warm-up” data was used to train the irrelevance filter for both cases in this experiment. We see that the overall *trend* of the second plot is correct, but the bias has shifted. This would indicate that the classifier is still correctly picking up the sentiment in the unlabeled articles, but its bias is in need of readjustment. This suggests that the ability to predict trends will not be very sensitive to concept drift (at least over short time periods). However, the inability to perform up-to-date bias correction will be a concern if the system is expected to continue to operate without any further training data.

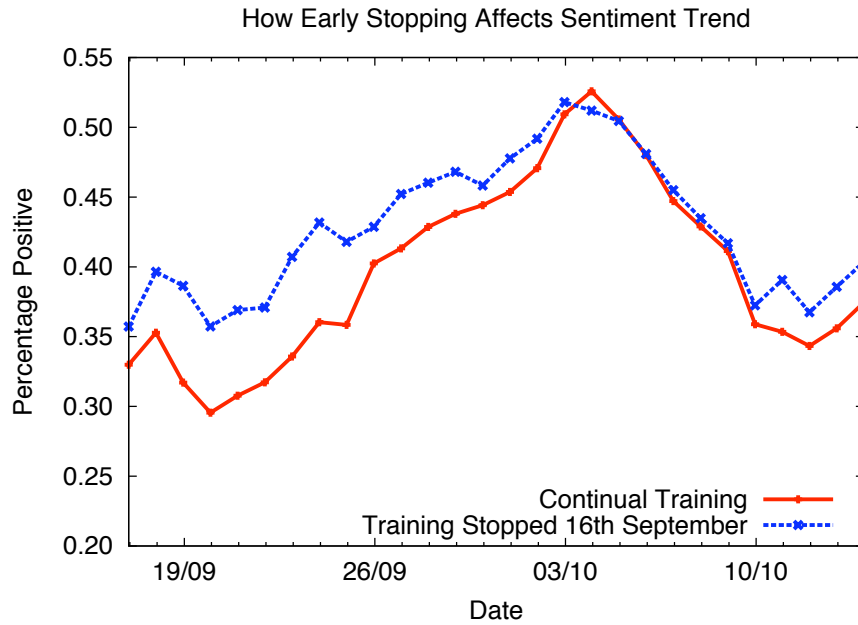


Fig. 12. This plot shows the impact of concept drift when the classifier is no longer updated with additional training data.

7 Conclusions

We have presented an analysis of the challenges in training a sentiment analysis system using data collected from non-expert annotators. Our objective has been to produce unbiased aggregate statistics on sentiment for large collections of news articles. We have focused on three key issues: *coverage* and *consensus* in the annotation process, and *bias* in the training process. We have also considered related issues, such as the selection of diverse items for annotation, and the effect of concept drift on classifier bias.

Our main conclusion is that the commissioning of a system such that described here should be preceded by a data *characterization* phase. This would explore the extent of the agreement between annotators and the amount of skew in the data. Our first important finding is that, if there is good agreement between annotators, then annotation effort should be expended on maximizing coverage rather than on identifying consensus. Our second finding is that, even when the skew in the data is modest, there is a clear need to correct for bias in the training of the classifier.

References

1. G. Attardi and M. Simi. Blog mining through opinionated words. In *Proc. 15th Text REtrieval Conference (TREC 2006)*, 2006.
2. S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 59–68, 2004.
3. J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, 2007.
4. A. Brew, D. Greene, and P. Cunningham. Using crowdsourcing and active learning to track sentiment in online media. Under review for IUI 2010.
5. P. Cunningham, M. Cord, and S. Delany. Supervised Learning. In M. Cord and P. Cunningham, editors, *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, pages 21–49. Springer, 2008.
6. A. Dawid and A. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.
7. P. Donmez, J. G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pages 259–268, 2009.
8. P. Hsueh, P. Melville, and V. Sindhvani. Data Quality from Crowdsourcing: A Study of Annotation Selection Criteria. In *Proc. NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 27–35, 2009.
9. A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proc. 26th Annual ACM Conference on Human Factors in Computing Systems (CHI'08)*, pages 453–456, 2008.

10. S. Matsumoto, H. Takamura, and M. Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. In *Proc. 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, pages 301–310. Springer, 2005.
11. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, volume 10, pages 79–86, 2002.
12. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proc. 15th International Conference on Machine Learning (ICML'98)*, pages 445–453, 1998.
13. V. Raykar, S. Yu, L. Zhao, A. Jerebko, C. Florin, G. Valadez, L. Bogoni, and L. Moy. Supervised Learning from Multiple Experts: Whom to trust when everyone lies a bit. In *Proc. 26th Annual International Conference on Machine Learning (ICML'09)*, pages 889–896, 2009.
14. V. Sheng, F. Provost, and P. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008.
15. P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of Venus images. *Advances in Neural Information Processing Systems (NIPS'95)*, 1995.
16. R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, pages 254–263. Association for Computational Linguistics, 2008.
17. T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354. Association for Computational Linguistics, 2005.