Taking the Pulse of the Web: Assessing Sentiment on Topics in Online Media

Anthony Brew
School of Computer Science & Informatics
University College Dublin
anthony.brew@ucd.ie

Derek Greene
School of Computer Science & Informatics
University College Dublin
derek.greene@ucd.ie

Pádraig Cunningham
School of Computer Science & Informatics
University College Dublin
padraig.cunningham@ucd.ie

Abstract

The task of identifying sentiment trends in the popular media has long been of interest to analysts and pundits. Until recently, this task has required professional annotators to manually inspect individual articles in order to identify their polarity. With the increased availability of large volumes of online news content via syndicated feeds, researchers have begun to examine ways to automate aspects of this process. In this work, we describe a sentiment analysis system that uses *crowdsourcing* to gather non-expert annotations for economic news articles. By using these annotations in conjunction with a supervised machine learning strategy, we can generalize to label a much larger set of articles, allowing us to effectively track sentiment in different news sources over time.

1 Introduction

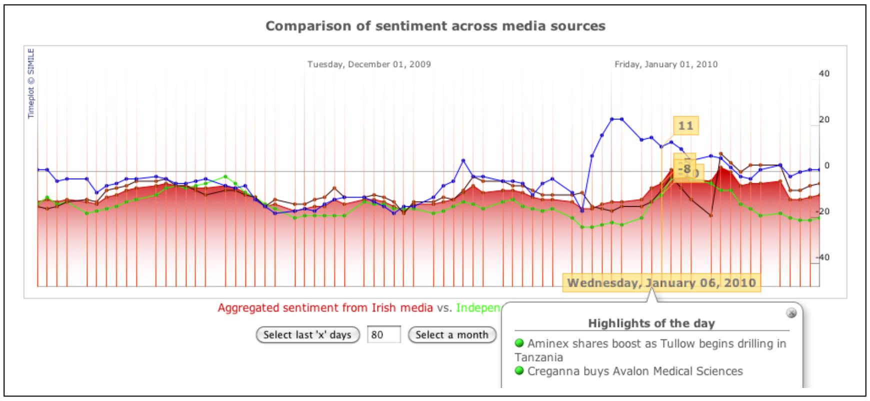
Here we are concerned with the challenge of tracking general sentiment trends on specific topics in online content. Until recently, tracking sentiment in the media required professional annotators with expert training to identify the polarity of individual articles, so that more general trends could be identified [5]. However, with the explosion in volume of online news content – both in the form of traditional news resources, and "new media" sources such as blogs and micro-blogs – the feasibility of relying on a purely manual approach to sentiment analysis is questionable.

Recently, *crowdsourcing* has become a popular way of automating labeling tasks without requiring the employment of expert annotators [2], by using systems such as Amazon's Mechanical Turk[1]. As an alternative to traditional sentiment analysis strategies, we present a system that uses crowdsourcing to gather non-expert polarity annotations, in conjunction with a supervised learning approach that generalizes from the user annotations to label a much larger body of news articles.

In the demonstration system[2] discussed here, we focus on tracking sentiment in the context of established online news sources and their coverage of the Irish economic situation. 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 over time, together with a micro-average reflecting overall sentiment. For instance, we can see that RTE, the national broadcaster (indicated

[1] See https://www.mturk.com

[2] See http://sentiment.ucd.ie


Comparison of sentiment across media sources

Highlights of the day

● Aminex shares boost as Tullow begins drilling in Tanzania

● Creganna buys Avalon Medical Sciences

Timeplot © SIMILE
Tuesday, December 01, 2009
Friday, January 01, 2010
40
20
0
-20
-40
11
-8
Wednesday, January 06, 2010
Aggregated sentiment from Irish media vs. Indepen
Select last 'x' days
80
Select a month

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

by the top line) appears to be more optimistic than the two other sources. By contrast the Irish Independent news feeds indicated by the green (bottom) line are much more negative. 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 (see Figure 3 later).

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 sentiment classifier to automatically label the complete set of news articles. A detailed discussion of the annotation and classification process is provided in Section 2.

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

- Once the sentiment classifier has been trained, a large number of unlabeled items can be classified to provide more robust statistics regarding sentiment trends.
- Once trained, the classifier has less *variance* than individual annotators alone.
- 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.

Our analysis of this strategy of combining machine learning and crowdsourcing shows that, while it is effective, there are a number of key issues that remain to be addressed. In particular, 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 3 we will show that popular nearest neighbor, naïve Bayes, and Support Vector Machine (SVM) classifiers are biased toward the majority class in our task. To address these shortcomings, in Section 4 we present a strategy for managing classifier bias. We also examine the degree to which concept drift impacts on our ability to eliminate bias.

2 System Description

2.1 System Overview

The main components of the sentiment analysis system are shown in Figure 2. The process begins with the retrieval of articles from the relevant online news sources. These articles are marked as *relevant* or *irrelevant* to the topic of interest – in this case economic sentiment – by a classifier that is kept up-to-date with labeled articles coming from the manual annotation process. This classifier achieves approximately 90% accuracy, indicating that separating economic news from other news is not a difficult text classification task. Articles that are considered *relevant* are passed to the annotation process, where a subset of articles is submitted for manual annotation by system users.

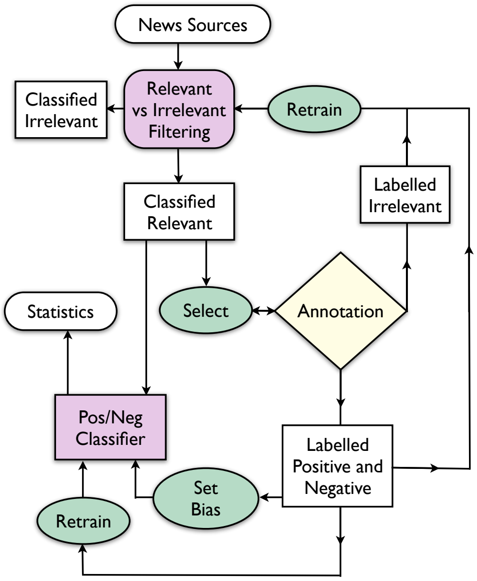


Figure 2: This flow chart shows the main components of the annotation, training and classification parts of the sentiment analysis system.

These manual annotations are used to retrain and improve the *positive/negative* classifier. Details on the management of the annotation process are provided in the next section.

In a system such as this, the value for users is based on a variety of channels providing relevant "interesting" content, many of which are enabled by the classification components. For example, the statistical visualizations of Figure 1 reward users with a sense of how their efforts are contributing to the system as a whole, as well as providing direct access to trending sentiment with current news. Users can review lists of the most positive, negative, and controversial articles for instance. Yet another example is presented in Figure 3, where users can benefit from tag-cloud summaries which highlight the most representative terms that are appearing in positive or negative articles at a given point in time. Moving the date slider provides a dynamic view of the most positive and negative topics over time.

2.2 The Annotation Process

Articles are collected from a pre-defined set of RSS feed URLs at the beginning of each day. In cases where only short descriptions are provided for RSS items, the original article body text is retrieved from the associated item URL. Those articles coming from the same domain (*i.e.* from the same

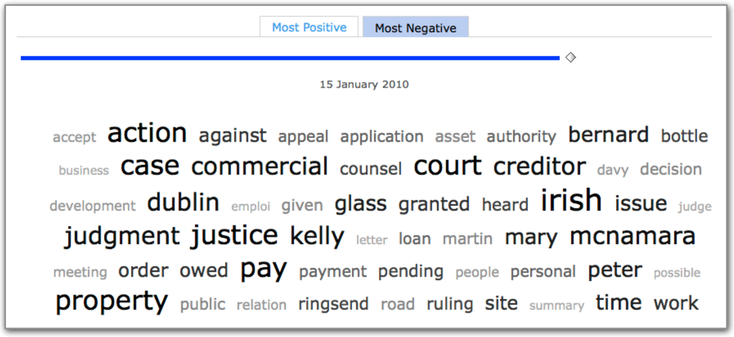

Most Positive
Most Negative
15 January 2010
accept action against appeal application asset authority bernard bottle
business case commercial counsel court creditor davy decision
development dublin emploi given glass granted heard irish issue judge
judgment justice kelly letter loan martin mary mcnamara
meeting order owed pay payment pending people personal peter possible
property public relation ringsend road ruling site summary time work

Figure 3: A screenshot of a tag cloud generated by the system, highlighting the most representative words occurring in articles annotated by users as being *negative* in sentiment at a given point in time.

news source) are grouped together. After applying the relevance classifier as described previously, articles not pertaining to economic news are filtered from the candidate set.

From the remaining relevant articles, a subset of approximately ten articles is chosen based on an appropriate query selection mechanism. To identify a diverse range of articles that provides a representative summary of the day's economic news, we apply a clustering-based article selection strategy, which applies complete-linkage agglomerative clustering to the data and takes representative articles chosen from the resulting clusters. As well as selecting articles that will be beneficial for the subsequent training phase in terms of covering as much of the domain as possible, this strategy also ensures the selection of a diverse subset of reading material for the annotators. This subset is then published as a custom individual RSS feed for each of the system's users.

To support the annotation process, a footer is appended to each RSS item in the custom feed containing links corresponding to the three annotation choices: Positive, Negative, and Irrelevant. Selecting a link submits a user's vote to the system on the article in question. The use of an RSS feed as a means of both delivering articles to be annotated and receiving annotation votes is designed to minimize the work-load of the annotation procedure in the context of a user's existing routine. We found that many users integrated the process as part of their existing news-reading habits – either via an online RSS reader (*e.g.* Google Reader) or a desktop news aggregator (*e.g.* Apple Mail). For those users who do not currently make use of an online or desktop RSS reader, many modern web browsers include the facility to render and display RSS feeds as web pages.

Annotations received from users are subsequently used to retrain the classification algorithms on a daily basis. The effectiveness of the next day's relevance filtering process is improved based on newly-collected *relevant* (*i.e. positive* or *negative*) or *irrelevant* votes. Similarly, articles that have been annotated as either *positive* or *negative* are included when re-training the second classifier. This is used to improve the quality of the summary statistics and visualizations on the web interface described previously.

3 Sentiment Classification

3.1 Experimental Setup

Using the system outlined in Section 2, 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 3,858 articles, with 2,693 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 12,469 documents, with 6,910 user annotations resulting in 1,306 labeled articles. Both datasets have been made available online[3] for further research.

3.2 Baseline Classification

For the classification components of the system, we examined three alternative supervised learning techniques that have previously been effective in text classification tasks [1]: 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 assessment using cross-validation. In these experiments we followed the methodology of Pang *et al.* [3], 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.

A summary of the results of the baseline evaluation are provided in Table 1. Accuracy figures are reported for each of the three classification techniques on two different sentiment analysis tasks (*i.e.* positive vs. negative, and relevant vs. irrelevant). We also report AUC (area under the ROC

[3] See `http://mlg.ucd.ie/sentiment`

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

Table 1: Comparison of baseline classifier accuracies for Positive vs. Negative and Relevant vs Irrelevant sentiment classification tasks.

curve) figures [4], as these consider classifier performance across a range of thresholds, making them independent of bias considerations.

These results corroborate the previous findings of Pang *et al.* [3], 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 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 deployed a naïve Bayes classifier in our sentiment analysis system.

4 Managing Bias

4.1 Classifier 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.

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).

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 4 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.

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.

4.2 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 2^{nd} August a classifier is trained on all labeled data up to 1^{st} 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 4 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. Note that a minimum number of articles was required to find the threshold crossover point in each fold. Therefore, to allow bias correction to start from the first day of the dataset time window, the training set was augmented with articles from the "warm up" dataset.

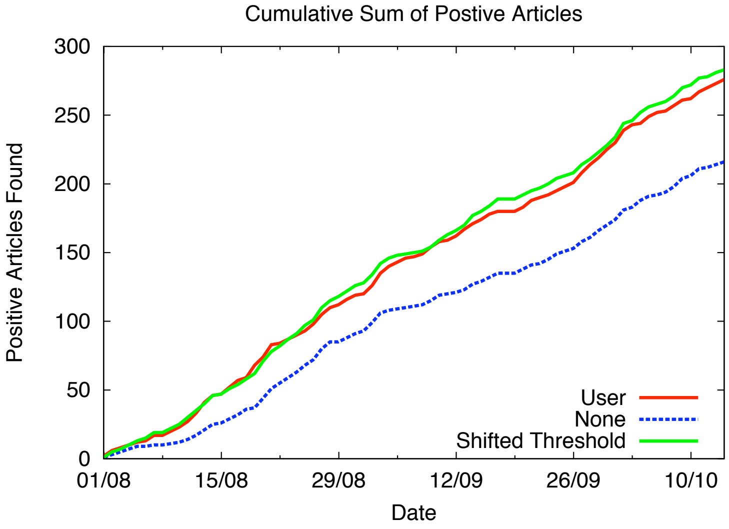


Figure 4: The cumulative distribution of positive annotations shows how the naïve Bayes classifier is biased against the majority class unless corrected.

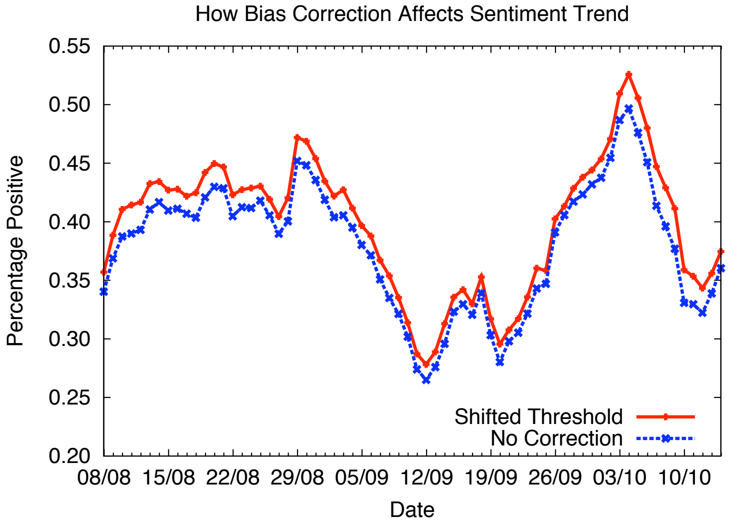


Figure 5: Time-plots produced with and without bias-correction.

The impact of this in an operational setting is illustrated in Figure 5. 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.

4.3 Concept Drift

One potential benefit of a system such as the one have we 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 6 we show two aggregate time-plots for the period between 17^{th} September and 14^{th} October. The first plot shows continuous updating as normal, while the second shows the case where training stopped on 16^{th} 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.

5 Conclusions and Future Work

We have presented a system for tracking sentiment trends in online content, using data collected from non-expert annotators. Our objective has been to produce useful aggregate statistics regarding sentiment for large collections of economic news articles. We have examined two key issues that arose when trying to meet this objective: dealing with classifier bias, and quantifying the effect of concept drift on bias.

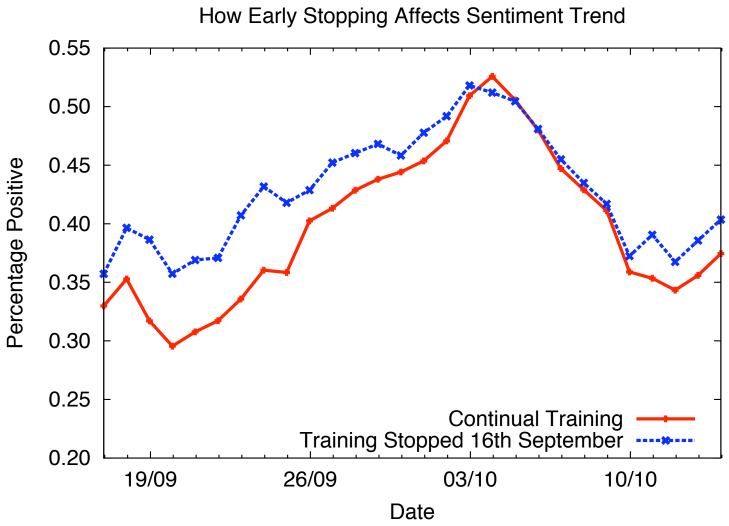


Figure 6: The impact of concept drift when the sentiment classifier is no longer updated with additional training data.

Currently the web interface to the system does not provide users with detailed information on how they compare to one another in terms of polarity. A potential risk here is that this information could influence annotator behavior, encouraging users to either conform to the average, or deliberately skew their annotations to be more positive or negative than other annotators. However, it may be the case that incorporating an additional social dimension to the system (*e.g.* visual comparisons of relative user polarity, or forums discussing controversial annotation decisions) could lead to improved annotation quality – through the self-correction of potentially erroneous annotations. Our next step will be to incorporate facilities for communication between the community of users to investigate this social dimensions in a sentiment annotation task.

Acknowledgments

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

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.