



A Fact/Opinion Classifier for News Articles

Adam Stepinski
Rice University
6100 Main St.
Houston, TX 77005
adamstep@rice.edu

Vibhu Mittal
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
vibhu@google.com

ABSTRACT

Many online news/blog aggregators like Google, Yahoo and MSN allow users to browse/search many hundreds of news sources. This results in dozens, often hundreds, of stories about the same event. While the news aggregators cluster these stories, allowing the user to efficiently scan the major news items at any given time, they do not currently allow alternative browsing mechanisms *within* the clusters. Furthermore, their intra-cluster ranking mechanisms are often based on a notion of authority/popularity of the source. In many cases, this leads to the classic power law phenomenon – the popular stories/sources are the ones that are already popular/authoritative, thus reinforcing one dominant viewpoint. Ideally, these aggregators would exploit the availability of the tremendous number of sources to identify the various dominant threads or viewpoints about a story and highlight these threads for the users. This paper presents an initial limited approach to such an interface: it classifies articles into *two* categories: fact and opinion. We show that the combination of (i) a classifier trained on a small (140K) training set of editorials/reports and (ii) an interactive user interface that ameliorates classification errors by re-ordering the presentation can be effective in highlighting different underlying viewpoints in a story-cluster. We briefly discuss the classifier used here, the training set and the UI and report on some initial anecdotal user feedback and evaluation.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

algorithms, experimentation, human factors

Keywords

text classification, news aggregation, interfaces, applications

1. INTRODUCTION

Several news aggregators like Yahoo, Google, Digg, Findory, etc. allow users to search and browse news articles from a very large number of online news sources. Since the number of news sources is much larger than the number of newsworthy events occurring at any given point in time, these

portals allow users – in theory – to browse dozens/hundreds of different reporting perspectives on a given story. In practice, however, it is extremely difficult to efficiently find stories that differ in interesting ways. Currently, the articles are presented to the user in a sorted list, much like results from a search engine. Since a user is unlikely to read through dozens of search results to find interesting articles on a topic, the current interface does not capitalize on this wealth of information, and thus the user does not benefit from the news aggregation. Furthermore, the articles related to a given story are sorted by “relevance”. Relevance, however, is not defined as the “most informative” article in a news cluster, but can vary based on the user’s interests, biases and previous browsing history. For example, political affairs are often reported, and seen through, very different eyes.

Ideally, a view-point browser would cluster stories along the major viewpoints and allow users to read the centroid stories in these story-viewpoint-clusters. This paper explores a much simpler version of this browser: by pre-defining two viewpoints, *opinions* and *facts*, it allows users to quickly scan at least two – rather than many – quite different versions. This is accomplished by training a text classifier on opinion/fact stories. (In this way, our work has similarities to both sentiment detection [3] and genre classification [2].) However, a strict classification can still be problematic because of inherent errors in the classification process (and stories are seldom pure fact or pure opinion). To prevent user frustration with obvious errors in classification, as well as to enable quick overviews of both classes, the system uses an interactive slider to select a proportion of opinions vs facts used to order the articles. The rest of this paper discusses the classifier, the preparation of an appropriate training set, the user interface and some preliminary, anecdotal evaluation.

2. ARTICLE CLASSIFICATION

Since articles can have both factual and opinionated parts, the system labels every *sentence* of an article as factual (1) or opinion (−1) using a binary classifier. The overall score for an article is the average of these labels. A slight modification to the algorithm weighs the labels based on the confidence of the classification. Thus, sentences for which the classifier is uncertain are weighted less in the average. By capping the contribution of any single sentence in an article towards the final article score, the resulting classification profiles are smoothly varying and reflect a possible mix of fact and opinion in the article. Sentence classification is done using a Passive-Aggressive (PA) algorithm [1]

trained on unigram, bigram, and trigram features. The PA algorithm is an online classifier that maximizes the margin of classification on poorly classified instances. As a discriminative classifier, it is better suited for our limited problem than a generative classifier such as Naive Bayes.

2.1 Training Set Generation

The sentence classifier is trained on a set of “fact” sentences and “opinion” sentences taken from a set of fact-based articles and opinion-based articles. The training set was generated by crawling online news sources and looking for a small set of specific keywords in the URL: articles with URLs that contained the tokens “opinion”, “editorial”, and “oped” were designated as opinion-based articles, and articles with URLs that contained the tokens “science”, “business”, “world” were designated as fact-based articles. Clearly, it is an oversimplification to assume that a sentence reflects the same level of objectivity as the entire article; for example, an editorial piece often states the facts of the issue being discussed. Therefore, we perform iterative training to reduce the impact of this noise.

2.2 Iterative Training

The iterative training process works by training a classifier using a subset of the desired features. Based on the results of classifying the training set with the classifier, we modify the training set and re-train with a larger set of features. This is similar to boosting, in which misclassified instances from the training set are given more weight in subsequent iterations. In our case, misclassified sentences are likely to be noise in the training set. By removing these problematic instances from the training set, we reduce noise in the features and lower variance in the model. We perform this iterative training three times. The specific process is described below.

1. Using the initial (noisy) training set, we train a classifier on unigram features.
2. We then run the training sentences through the resulting classifier. Misclassified sentences are removed from the training set.
3. We now use the new training set to train a classifier on unigram and bigram features. Again, we remove any sentences misclassified by the new classifier.
4. The final training set is used to train a classifier on uni-, bi-, and trigrams.

3. EVALUATION

For training, we use a set of sentences from 70,000 fact-based articles and 70,000 opinion-based articles collected online using the criteria described earlier. We performed the iterative training process described above to create a classifier trained on unigram, bigram, and trigram features. Training and evaluation were conducted using 5-fold cross validation of the classifier on the iterated training set. The average F1-score of the cross-validation was 85%.

Using additional features such as POS tags and article length lowered the F1-score to 80%. This was somewhat surprising, since (i) one of our assumptions was that “fact” classifications were being triggered by stories having a higher than normal density of numbers and names versus “opinions” that might have higher than normal densities of adjectives and common nouns; (ii) at first glance, fact based

sentences seem shorter compared to opinion sentences, but this does not make a difference in the classifier accuracy, and does not carry over to article length either. One explanation for these features not helping in our experiments may have been due to over-fitting the model on the relatively small data set. Due to time limitations, we could not come up with a conclusive explanation of this degradation in classifier accuracy. Nor were we able to explore more complex features such as relative position of the n-grams in the sentence/article, formatting cues, etc. These are all possible avenues for future work.

4. APPLICATIONS

It is not always clear, even to us, when a story should be classified as “fact” or “opinion”. Clearly, a model based on n-grams seen previously in a training set is going to be occasionally confused as well. In order to make these classifications useful to the user without emphasizing the errors, we had to come up with an interface that was both intuitive and takes advantage of the strengths of the classifier: the fact that sentence classification mistakes can still be overlooked as long as a significant part of the article is as labeled.

The main feature of the interface we eventually used was a dynamic slider. By moving the slider, the user controls the desired proportion of fact and opinion content. The articles on the page dynamically change to display those with the closest matching classification score. In testing, the slider interface provides several advantages over the current interface: (i) The user is in control: users can select the kinds of articles they want to read. A person just learning about yesterday’s top news story may wish to sort the results to get fact-based articles, while a well-informed person may be more interested in the opinions. In either case, the user simply moves the slider until the top results align with their preferences. (ii) Quick and effective exposure to information: with the slider interface, results bubble up to the user, instead of the user drilling down through the results. The sorting criteria provides a very natural ordering for the articles, allowing the user to quickly get an overview of both the facts and opinions on a given topic. (iii) Amelioration of errors: The fact/opinion classifier isn’t perfect. Indeed, if the interface provided a strict “fact or opinion” judgement for all articles, users would not tolerate errors in the classification. By giving the user control over the slider and not offering strict judgments, the interface eliminates sources of user frustration.

5. REFERENCES

- [1] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms, 2006.
- [2] B. Kessler, G. Nunberg, and H. Schütze. Automatic detection of text genre. In P. R. Cohen and W. Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics*, pages 32–38, Somerset, New Jersey, 1997. Association for Computational Linguistics.
- [3] J. Wiebe, T. Wilson, and C. Cardie. Exploiting subjectivity classification to improve information extraction. In *Proc. 20th National Conference on Artificial Intelligence, AAAI*, 200.