

Negative News No More: Classifying News Article Headlines

Karianne Bergen and Leilani Gilpin
kbergen@stanford.edu lgilpin@stanford.edu

December 14, 2012

1 Introduction

The goal of this project is to develop an algorithm that will classify a news article headline according to how positive or uplifting that news story is. The algorithm should be able to distinguish between positive story “*Baby elephant rescued in Kenya with rope and a land rover*” and negative headline “*Firms on alert for letter bombs*”. The algorithm will also identify as neutral those stories that are neither strongly positive or negative (e.g. “*iPad changing how college textbooks are used*”). The focus is on classifying the content/topics of the articles as positive/negative rather than the attitude of the author toward the subject, since news articles are typically written in an objective style.

1.1 Data Collection and Data Set

For this task we collected two data sets. The first is a set of 4294 news article headlines. The second is a set of 2504 news article headlines with short text excerpts from the article (typically the first 2-4 sentences of article text). Most of the data samples were extracted from RSS feeds over several weeks during Fall quarter 2012. We did not include stories collected on election day, as these stories were repetitive and overwhelmingly neutral (strongly polarizing headlines are classified as neutral). The sources of the news feeds include Google News, CNN, BCC, Fox News, and The New York Times. We included headlines from a pre-existing data set consisting of headlines from news websites in 2007 [2]. In order to avoid a skewed data set, we obtained many “positive” headlines from [3] and [4]. Our final data sets are roughly evenly divided between the three classes, with each class representing 30-40% of the samples in each data set. The positive/neutral/negative split is 38.0/29.4/32.6% for headline-plus-text data, and 31.4/34.2/34.3% for headline-only data.

Each data sample, corresponding to a single news article, was assigned to one of three classes, “positive,” “negative,” or “neutral.” The data samples were classified by the two project team members. Articles were classified as “positive” if they featured a happy, inspiring, funny, or uplifting topic. Articles classified as “negative” typically include themes of violence, crime, natural disasters, and loss of life or property. Articles that did not strongly fall into either category, including polarizing articles (e.g. on controversial or political subjects), were classified as “neutral” (see Table 4 for examples).

2 Method

2.1 Feature Extraction

We use a bag-of-words model for headline classification. One set of features was generated for each of the two data sets. Each feature set was based on a dictionary of words extracted from the headline (and text excerpt) data. Features represent individual word fragments (tokens), with an additional feature indicating whether a numeric value appears in the headlines. The feature set excludes stop-words (e.g. “about,” “over,” “with”). Suffixes were removed, both

automatically and manually, from dictionary words to create the list of tokens appearing in each data set. The feature set for the headline-only data includes 5781 features (5780 tokens, 1 numeric) and the feature set for the headline-plus-text data includes 10010 features (10009 tokens, 1 numeric).

2.2 News Article Classification

For news article classification, we used both naive Bayes and support vector machine (SVM) classifiers. 70% of the data was used for training and cross-validation and 30% for testing. The training/test sets included 2994/1300 samples for headline-only data and 1754/750 samples for headline-with-text data. 30% of the training data (21% of the total data set) was used for cross-validation.

2.3 Naive Bayes

We used two different naive Bayes models; one models the three classes, “positive,” “negative,” and “neutral,” while the other models two classes, “positive” and “negative,” and uses a threshold parameter to define the “neutral” class from this model. Both the two-class and three-class naive Bayes classifiers use a multinomial event model with Laplace smoothing.

Our initial attempts using a two-class naive Bayes classifier involved mapping multiple results to a single class prediction. We trained separate classifiers for positive vs. non-positive, negative vs. non-negative and neutral vs. non-neutral. However, this method had limited success, as discrepancies among the three predictions for a single sample degraded performance of the method.

Our successful use of the two-class naive Bayes classifier attempts to exploit the fact that the neutral class represents an intermediate class between positive and negative. In our two-class model, the data set was trained on “positive” and “negative” samples only, disregarding the “neutral” examples. The prediction for the “neutral” class is based on a thresholding scheme; if the difference in posterior probabilities for the positive or negative classes is below a specific threshold for a given test sample, it will be classified as “neutral.” The best thresholding method (absolute difference of log-probabilities) and the threshold value (3.1 and 1.5 for headline-and-text and headline-only, respectively) were selected using hold-out cross-validation.

The three-class naive Bayes classifier was the most successful classifier for our classification problem. One of the model parameters we experimented with was a weighting scheme for words that appeared in the headline as opposed to those in the text-excerpt. Since the headline generally contains the most pertinent keywords relating to the article’s content, we wanted to take advantage of the distinction between the headline and text-excerpt data. In this model, the frequencies of words appearing in the headline are multiplied by a weight $\alpha \geq 1$ and frequencies of words in the text-excerpt are weighted $\beta = 1$. We used hold-out cross-validation to select the optimal value of α , however we found that varying the weighting factor α did not have a statistically significant impact on the accuracy of the classification. Therefore, the unweighed frequencies ($\alpha = 1$) were used in our model.

2.4 Support Vector Machine

We also used a support vector machine classifier with a linear kernel, implemented using the Spider for Matlab library [5]. Initially, the SVM did not include regularization. As a result the classifier tended to over-fit the training set and have an accuracy roughly 10% lower than

that of the three-class naive Bayes. Therefore, we introduced regularization via soft-margin parameter, C . We used hold-out cross-validation to select the parameter value, $C = 0.2$ and $C = 0.02$ for headline-only and headline-with-text data respectively.

2.5 Feature Selection

We also tried using feature selection to improve algorithm performance. We used a filter feature selection method, using the mutual information measure to score the features. We then applied hold-out cross-validation to select the optimal number of features. The result was that for both the naive Bayes and SVM classifiers for each data set, the best performance was obtained if the full features set was used. We did find however, that the marginal improvement of features was relatively small after roughly 60% of the features with largest mutual information scores were included in the features set. Thus we found that the number of features can be reduced by 40% with a 5% loss in accuracy and 2-3% increase in positive/negative errors.

2.6 Performance Metric

One of the difficulties of categorizing news stories as positive, negative or neutral, is that these class distinctions are relatively subjective, especially for the “neutral” label. For this application, we are most concerned with our algorithm correctly differentiating between positive and negative stories.

In assessing the performance of the algorithm, we divide the results into three cases: “Exact Match,” “Neutral Error,” and “Positive/Negative Error.” “Exact Matches” represent test samples for which the predicted class is identical to the ground-truth class. “Neutral Errors” represent test examples that are either incorrectly predicted to be in the neutral class, or are neutral samples that have been incorrectly predicted as positive or negative. “Positive/Negative Errors” represents test samples for which the ground-truth class is either positive or negative and the algorithm predicts the sample belongs to the opposite class. To measure algorithm performance, we use the percentage of “Exact Matches” and “Positive/Negative Errors” in the test set.

3 Results

In our milestone, diagnostics showed that our algorithm performance may be improved with more data (either more headline samples or more text per article). Figures 1 and 2 show that the accuracy of the three-class naive Bayes levels out and no longer improves significantly for larger training sets. Therefore, while we observe that the results still show moderate bias, we do not believe that the current performance is limited by the size of our data set.

A comparison of the performance for different classifiers is shown in Tables 1-3 and Figures 3 and 4. Figures 3 and 4 indicate that three-class naive Bayes has the best performance on both data sets. Three-class naive Bayes has 70.4/65.5% accuracy, 4.9/5.7% positive-negative error on headline-and-text/headline-only data. The second-best performing method for headline-only data is two-class naive Bayes with 62.8% accuracy and 3.5% positive-negative error. The regularized SVM was the second-best performing method on headline-and-text data with 68.8% accuracy and 6.1% positive-negative error. Generally, the classification of headline-and-text news data has greater accuracy than the classification of headline-only text. When text data is included, accuracy improves by 2-9% for a given classification algorithm.

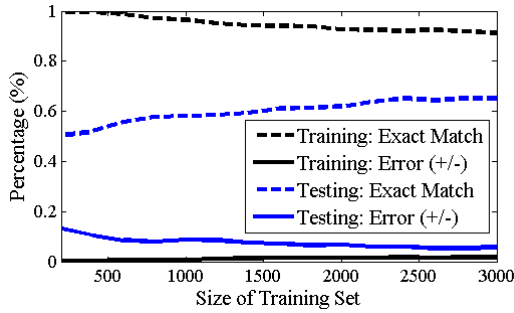


Figure 1: Accuracy vs training set size for headline-only news articles, 1300 test samples

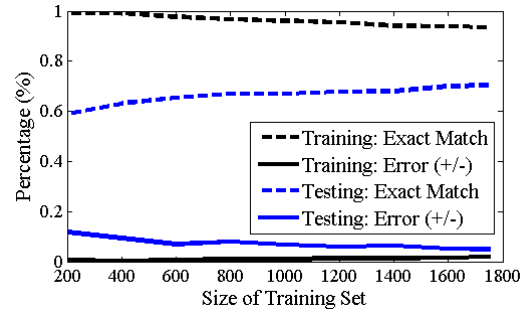


Figure 2: Accuracy vs training set size for headline-and-text news articles, 750 test samples

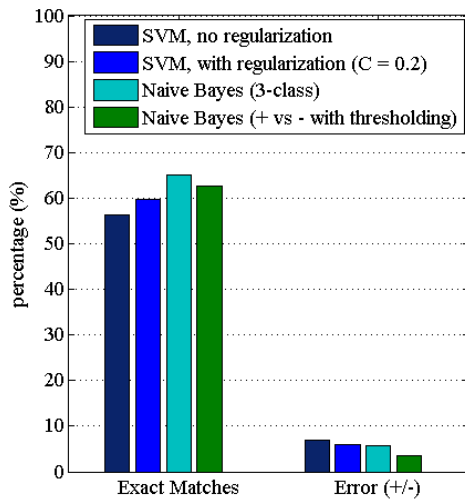


Figure 3: Classification of headline-only news articles, 1300 test samples

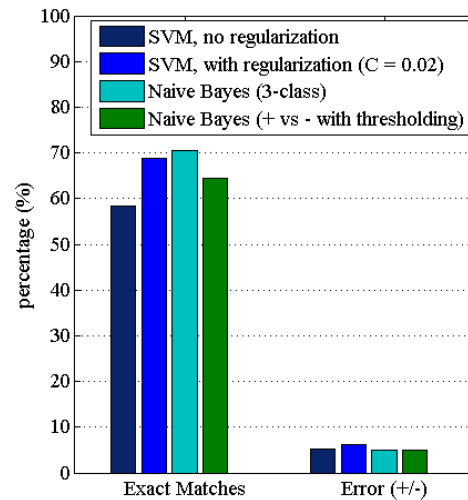


Figure 4: Classification of headline-and-text news articles, 750 test samples

Tables 1-3 show the results of three different classification schemes on the headline-with-text data. Each entry in the table corresponds to the number of test samples, out of a total of 750 (279 positive, 225 neutral, 246 negative), that fell into each category. Green denotes an accurate prediction, yellow denotes errors in the classification of neutral articles, and orange denotes the more significant positive-negative errors. Tables 1 and 2 indicate similar performance trends for three-class naive Bayes and regularized SVM classifiers.

The best accuracy achieved by our classifiers is around 70% for headline-and-text data, and 65% for headline-only data. While this still leaves a substantial number of misclassifications, the positive-error rate is roughly 5% and thus the majority of the incorrect predictions involve the neutral class. However, as discussed above, such errors are likely related to the subjective class definitions. As a baseline, an experiment in which 315 articles (headlines-with-text) were independently classified by two different individuals indicated that human classifications match for roughly 70% of news articles (including a 1-2% disagreement on positive and negative classes). Therefore, the algorithm performance is roughly on par with

	Output		
	(+)	(O)	(-)
True (+)	220	36	23
True (O)	59	113	53
True (-)	23	40	183

Table 1: SVM with regularization

	Output		
	(+)	(O)	(-)
True (+)	226	33	20
True (O)	55	117	53
True (-)	17	44	185

Table 2: 3-class naive Bayes

	Output		
	(+)	(O)	(-)
True (+)	222	34	23
True (O)	65	70	90
True (-)	15	39	192

Table 3: 2-class naive Bayes with thresholding

human performance in this task.

Table 4 includes examples of classification results for the three-class naive Bayes on headline-with-text data. This gives a sense of the types of headlines for which the algorithm performs well and the types of class ambiguities that exist due to the subjective nature of the classification.

	output (+)	output (O)	output (-)
True (+)	“Aviators give puppies a second chance”	“Breeze through TSA security during the holidays”	“Doc helps others after losing son”
True (O)	“Afghan opium harvest down sharply”	“California’s housing market sees mixed recovery”	“Congress looks at ways to raise taxes”
True (-)	“With a friendly face, China tightens security”	“Zombies attack VIP in California”	“French citizen kidnapped in Mali”

Table 4: Headline classification output

4 Conclusion and Future Work

Given the subjective nature of our classes, future work may involve creating more personalized recommendations of positive and negative stories based on the user’s preferences. We also hope to use our algorithm to create a web application that filters article feeds to bring the users only happy and uplifting new stories for when they’re having a bad day. We plan to make our data set publicly available for the machine learning community.

References

- [1] Sriram, Bharath, Fuhry, David, et al. “Short Text Classification in Twitter to Improve Information Filtering” *SIGIR ’10: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp.841-842, 2010.
- [2] Strapparava, Carlo and Mihalcea, Rada. “Dataset for Emotions and/or Polarity Orientation.” *SemEval-2007: 4th International Workshop on Semantic Evaluations*. 2007.
- [3] “Great News”. <http://www.greatnewsnetwork.org/>
- [4] “HuffPost Good News.” <http://www.huffingtonpost.com/good-news/>
- [5] Spider for Matlab (library). <http://people.kyb.tuebingen.mpg.de/spider/>