Lexical Approaches to Satire Detection

Rachel Fong (rfong@mit.edu)
6.864 Fall 2010

We experiment with lexical approaches to classifying satirical news articles. Primarily, we investigate bag-of-words features, and also propose the use of sentiment analysis to detect sarcasm. We achieve high accuracy and reasonable precision and recall using only lexical features; results are comparable with other attempts.

1 Introduction

Due to its inherent subjectivity, satire recognition is a largely unexplored problem in natural language processing. Even humans do not always recognize satire – clever executions of satire require real-world knowledge and occasionally a particular viewpoint to understand. Subsequently, devising approaches toward the problem of satire recognition requires some insight into the human perspective. As suggested by the Oxford English Dictionary[1]:

"*satire* (n.) – the use of humor, irony, exaggeration, or ridicule to expose and criticize people's stupidity or vices, particularly in the context of contemporary politics and other topical issues."

Related literature on humor recognition (Reyes et al., 2009) suggests that this component of satire is far too complex to tackle within our timeframe. Similarly, any incorporation of real-world knowledge would be time-consuming and require a massive, constantly updating database. Few attempts at satire recognition have been made, but impressive results have been achieved through experimenting with various feature scaling methods using only features based on lexical properties and entity recognition (Burfoot and Baldwin, 2009).

Many cutting-edge problems in natural language processing, particularly those concerning social media, require data about informal speech. We obtain data from online open-content dictionaries in order to supplement our analysis.

2 Corpus

For the purposes of this study, we will examine only news articles, since reliable classifications of news articles are more straightforward to obtain in large quantities than other forms of satire, such as literature or essays. In addition, for the sake of testing simplification, we will limit our corpus to documents written in English.

We use a corpus made publicly available[2] by Burfoot et al. It contains 4000 real news articles and 233 satire news articles. Each of the satire articles is topically related to at least one news article in the corpus, increasing the probability that topically relevant characteristics will be distinguishable from satirical characteristics. All documents are formatted in plaintext and

[1] http://www.oed.com/

[2] http://www.csse.unimelb.edu.au/ research/lt/resources/satire/

edited to remove characteristics particular to their sources.

3 Approach

3.1 Features

Many characteristics particular to satire can be effectively quantitatively evaluated with a simple bag-of-words model. Below, we describe several characteristics and corresponding features. In order to reduce ambiguity, we parsed and grammatically tagged words in the corpus using the Stanford Parser[3].

Colloquialism: Satirical articles are less likely to adhere to formal language than serious articles are. We scrape Wiktionary[4] entries tagged as 'informal', 'colloquial', or 'slang' to obtain an initial slang vocabulary V_S consisting of approximately 6000 words. Parameter training is discussed in Section (3.2). We then define the informality of a document x, comprised of words $x_1, ..., x_n$, as:

$$S(x, V) = \frac{1}{n} \sum_{i=1}^{n} s(x_i, V^t)$$

where s is a function on a word w and vocabulary V^t:

$$s(w, V^t) = \begin{cases} 1 & w \in V^t \\ 0 & \text{otherwise} \end{cases}$$

Exaggeration: Satire articles tend to exaggerate situations in order to emphasize their message. We will define the exaggeration of a document x similarly to colloquialism, using Wiktionary entries tagged as 'degree adverb' to obtain an initial vocabulary V_E consisting of approximately 150 words.

$$E(x, V^t) = \frac{1}{n} \sum_{i=1}^{n} e(x_i, V^t)$$

where e is a function on a word w and vocabulary V^t:

$$e(w, V^t) = \begin{cases} 1 & w \in V^t \\ 0 & \text{otherwise} \end{cases}$$

Topicality: Satire articles ridicule by creating absurdity. A common characteristic of absurdity is irrelevance. For example, one satire article in the corpus is headlined 'Device Flips Pachyderms So They Can Get Pedicures'. The probability of seeing 'pachyderms' and 'pedicures' in a document together is low, increasing the probability that this document is 'absurd'.

We target this characteristic by using a naive Bayes classifier. First, we calculate the probability that a word will appear in a document, given the presence of another word:

$$P_I(w_2|w_1) = \frac{1}{m} \sum_{i=1}^{m} p(w_2, x^i)$$

where p is a feature describing whether a word w is present in a document x:

$$p(w, x) = \begin{cases} 1 & w \in x \\ 0 & \text{otherwise} \end{cases}$$

Then for each test document x, we define the non-topicality, or irrelevance, as:

$$I(x) = \frac{1}{Z} \prod_{i=1}^{n} \prod_{j=1}^{n} [1 - P_t(x_i \,|\, x_j, i \neq j)]$$

where Z is a normalizing constant, and zero values are ignored.

3.2 Training

Our slang and exaggeration features each take a parameter θ: a vocabulary comprised of a set of words $w_1, ..., w_k$. Words tagged as colloquial often have more ubiquitous alternative definitions that are considered formal; for example,

[3] http://nlp.stanford.edu/software/lex-parser.shtml
[4] http://en.wiktionary.org/w/api.php

'kicks' is slang for shoes, but is also simply the present tense of the verb 'to kick'. This suggests that more accurate classifications on the training data could be achieved by pruning the vocabulary.

3.2.1 Scoring

We score a set of classifications using its F-score:

$$\begin{aligned} \text{F-score} &= 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \\ &= 2 \cdot \frac{\#\text{ correct results}}{\#\text{ satire docs} + \#\text{ results}} \end{aligned}$$

Then, given documents $x^1, ..., x^m$ and a set V of possible vocabulary words, we select the vocabulary subset yielding the maximal F-score possible.

3.2.2 Efficient score computation

In order to determine optimal vocabularies, we must compute document scores over each vocabulary. Since our feature score is evaluated via a simple sum, we can minimize computation by only changing our vocabulary incrementally and using dynamic programming. Our table π has $\|V\| \cdot m$ entries such that:

$$\begin{aligned} \pi[i][j] &= \underset{V_i \| \|V_i\|=i}{\operatorname{argmax}} score(x^j, V_i) \\ &= \underset{w \in V}{\operatorname{argmax}}\ \pi[i-1][j] + c(x^j, w)/\|x^j\| \end{aligned}$$

This method of calculating feature score is equivalent to the simple definitions shown in Section (3.1), but enables faster computation by taking advantage of our incremental vocabulary selection.

3.2.3 Algorithm

Our goal is to obtain the subset of the initial vocabulary yielding the highest score over all documents.

We incrementally search for the best vocabulary subset of each possible length, using a linear support vector machine to find the best separator and determine a temporary set of classifications from our score set. When the best score for the current length is maximized, we quit. Our algorithm has time complexity $O(|V|^2 \cdot m \lg(m))$, although in practice it is faster because we quit after selecting only a fraction of the initial vocabulary.

3.3 Feature weighting

Previous literature (Burfoot and Baldwin, 2009) suggests that bi-normal separation (Forman, 2008) is a highly accurate method of feature scaling. We use it to determine the final classification for our documents. The weighting for a particular feature as calculated by BNS is:

$$|F^{-1}[P(\text{feature}|+)] - F^{-1}[P(\text{feature}|\text{-})]|$$

where F^{-1} is the inverse cumulative distribution function.

4 Results & Conclusions

feature	A	P	R	F
exaggeration	0.909	0.336	0.460	0.388
slang	0.932	0.459	0.510	0.483
topicality	0.329	0.078	0.890	0.143
BNS final	0.932	0.459	0.510	0.483

The topicality feature had notably poor performance, likely caused by our neglect to compensate for words appearing in the test set which were not present in the training set. However, we achieved high accuracy and reasonable precision overall using only lexical features over a substantial corpus of real-world documents. These results are promising for future work in satire recognition.

It is also of note that our vocabulary training algorithm stopped searching once the best local score stopped increasing; it is possible that we could have achieved better performance by instead stopping once the best local score began decreasing, as our vocabulary would then be larger and perhaps better equipped to deal with the training set.

4.1 Future work

We had initially considered implementing other promising features, but did not due to time constraints.

Sarcasm detection: Sarcasm is difficult to approach lexically. We planned to detect sarcasm by learning positive and negative connotations for words, and then using simple sentiment analysis to evaluate sentiment consistency. This would be highly effective for certain ubiquitous types of sarcasm such deliberate contradictions, as when a negative entity is described in a positive way or vice versa. For example:

"*I was robbed earlier. What a great day.*"

Real-world entity recognition: This could be used to detect fabricated scenarios. Unfortunately, it might require an unreasonably large vocabulary, since this feature would need to recognize more specific entities than the other lexical features we implemented.

References

[1] Antonio Reyes, Davide Buscaldi, Paolo Rosso. *The Impact of Semantic and Morphosyntactic Ambiguity on Automatic Humour Recognition.* 2009.

[2] Clint Burfoot and Timothy Baldwin. *Automatic Satire Detection: Are You Having A Laugh?* 2009.

[3] George Forman. *BNS scaling: An improved representation over TF-IDF for SVM text classification.* 2008.