**Matt Barnard**

**Yanan Li**

**Abby Mann**

**Just Kidding!: Detecting Irony to Increase Accuracy of Sentiment Analysis**

**Introduction**

Almost simultaneously with its appearance, Twitter became a frequently used tool by academics as well as businesses. Tweets are not just easy to access and process; they come from a wide variety of users and provide a manageable amount of textual data that is public information unless the user opts out (and only about 10% of users do). These factors make twitter “a valuable resource for tapping into the zeitgeist of the internet, its users, and often beyond,” argue Michael Zimmer and Nicholas Proferes (2014). As early as 2010 (O’Conner et al), researchers were attempting to correlate twitter content with political outcomes, but, as Bovet et al report, early results were inconclusive, in part because studies recorded political attention as opposed to political support (2018).

Thus sentiment analysis, the use of machine learning to predict the overall attitude of a speaker or document, offers a valuable tool in forecasting political outcomes through analysis of tweets, a need that has become the more pressing because of the dramatic failure of polls in predicting events such as Brexit, the 2016 US Presidential election, and 2017 elections in France, Britain and Chile (Bowman 2018). This decline in accuracy in polls, as well as the wealth of political material on twitter, makes it an obvious fit for new forms of prediction, and various methodologies such as tracking sentiment analysis over time (Porcaro and Muller 2016), doing content analysis to rack emotional intensity and behavioral patterns (Kondik 2017) and tracking commentators opinions of a candidate (Bovet et al. 2018) have all been shown to be validated in correlation with external polls and election results.

However, as Bovet et al point out, a “lexicon-based approach perform[s] poorly on the informal, unstructured, sometimes ironic, language of Twitter.” This was marked as one of the major problems with the data-set we are examining, a corpus of tweets about 2016’s first GOP debate. As one user on Kagle noted, ntlk and Naive Bayes “works rather well for negative comments. The problems arise when the tweets are ironic, sarcastic[,] has reference or [a] difficult context” (Nagy 2017).

Thus a fundamental issue arises in using twitter for political forecasting: the use of irony and sarcasm makes the very act of sentiment classification difficult. While there are numerous other issues to tackle beyond sentiment analysis in using twitter as a predictor of political outcomes, without accurate sentiment analysis, many paths are closed off.

**Research Question**

Thus the fundamental issue we address in this project is whether applying what is known about how sarcasm and irony appear in tweets will lead to increased accuracy in classifying the sentiment of such tweets.

Take, for instance, one of the tweets from our data set: "Muhaha, how sad that the Liberals couldn't destroy Trump. Marching forward." This is actually a positive tweet--it expresses approbation-- but the appearance of “sad” and “destroy” caused it to be read as negative. Our goal, then, is to reduce the number of false negatives by using markers of sarcasm as features increase the accuracy of sentiment readings.

This is a slightly different take than most of the current work on irony in social media, which focuses on accurately detecting the presence of irony: or hypothesis is that adding the recognized elements of ironic expression in tweets to the feature set for sentiment analysis will result in more accurate sentiment analysis.

**Related Work Section**

There are several distinct approaches and goals to studying political twitter content. In addition to the studies cited above, which offer us a sense of the stakes of accurately classifying sentiment and how it can be used for forecasting, Christopher Mascaro and Sean Goggins’ “Technologically mediated political discourse during a nationally televised GOP primary debate” (2016) addressed what they saw as a gap in previous analysis of sentiment in political tweets: the “effects of specific statements or content during the debate” on sentiment. Their particular focus was on whether Twitter can serve as a “technological public discourse sphere,” and they concluded, by tracking interactions before, during, and after the debate that it was not, because users tended to reamplify previous information or tweet directly at candidates, as opposed to engaging with each other. This offers a useful baseline establishment of what research people are doing about political tweets and why it matters. The question of whether Twitter can serve as a true public sphere is an interesting one and the ability to accurately code irony or sarcasm may well help in analyzing how this debate takes place. Additionally, the emphasis on the temporal elements of political twitter are important because irony is often marked through reference to world knowledge (Reyes et. al 2008, Maynard and Greenwood 2014).

While the majority of the literature focuses on classifying ironic or satirical tweets, in the 2014 conference presentation “Who Cares about Sarcastic Tweets?,” Diana Maynard and Mark Greenwood investigate the effect of identifying sarcasm in tweets on sentiment analysis, finding doing so could improve sentiment detection by nearly 50 percentage points, although accuracy was still fairly low overall. They offer two particularly useful approaches: firstly, they

developed a hashtag tokenizer for GATE (Cunningham et al., 2002), which means multi-word hashtags can be checked for content and sarcasm markers. Additionally, they point out that a sarcastic marker does not mean that one can automatically flip the polarity because there are often multiple sarcastic or ironic markers that interact with each other, so they developed a set of rules for using multiple hashtags with sarcastic intent.

Much of the work on sarcasm and irony has been focused on being able to identify its presence. In “A Multidimensional Approach for Detecting Irony in Twitter,” Antonio Reyes, Paolo Rosso and Tony Veale tackle the fact that there can not be a “computational silver bullet” to identifying irony because it “cuts through every aspect of language, from pronunciation to lexical choice, syntactic structure, semantics and conceptualization,” but they argue that finding the aspects amenable to computational analysis will allow a way to address the issue (240). Florian Kunneman et al’s seminal “Signaling Sarcasm: From Hyperbole to Hashtag” used the marker #sarcasm, and several other related markers, to train an algorithm to detect sarcasm, irony or cynicism, detecting 87% of the tweets when the hashtag was removed, but noting that the presence of the #sarcasm marker seemed to lessen “the further use of linguistic markers for signaling sarcasm, such as exclamations and intensifiers” (although this study was conducted with Dutch users, they found similar patterns in a cross-lingual study). Kunneman et al detail four types of markers for sarcasm that commonly occurred ( “intensified as well as unintensified evaluative words, exclamations, and non-sarcastic hashtags,”) and noted that unintensifed language was the strongest indicator, by a narrow margin.They also note other markers that indicate sarcasm: “other linguistic markers, such as rhetorical questions, repetition, echo, change of register, interjections, or diminutives, many of which cannot simply be inferred from the presence of words or n-grams of words. Since gender, education and profession can be predictors of the use of irony, just like previous use of irony, we also need to further explore the possibilities of including speaker and context characteristics in the model” (508).

Reyes et al. look at a similar set of features (textual elements, typographical and parts of speech, that throw focus onto certain parts of the text; unexpectedness, within a situation, between contexts, or temporal; style, which is the distinctive patterns of language across a text; emotional scenarios, which go beyond emoticons and single terms to vectors of activation, imagery, and pleasantness) and discovered almost all of them were statistically significant and led to increased accuracy of irony detection, although accuracy was lower when dealing with a test set of about 30% ironic tweets (a normal distribution) as opposed to an artificially balanced one. The authors argue that the test results so far suggest the value of continuing to seek out discriminating features to increase accuracy and “that a system of textual features can capture the linguistic patterns used by people when communicating what they believe to be ironic state” (255). One particular focus they offered was the idea of polarity s-grams, measuring the ratio of positive and negative sequences, as opposed to words themselves, since irony often uses positive words negatively and vice versa. Maynard and Greenwood (2014) also noted that word combinations with opposite polarity, especially ones using a strong sentiment word or a swear word, marked sarcastic intent.

Reyes et al also noted that position could be useful in identifying sarcasm, which was often in the first 4-7 words of a tweet, making that portion have the strongest predictive value.

**Approach:**

Our approach was to work from documented data and models used in previous studies in order to establish the baseline classifier, feature space and performance metrics used in hopes of replicating a comparable model of our own. The data we worked with includes 13871 tweets and labels sourced from CrowdFlower regarding the sentiment (positive, negative, or neutral) of these tweets as they pertain to the 2015 GOP debate.The underlying distribution of the sentiment gathered from these tweets was quite skewed towards negative: 2236 (~16%) were positive, 3142 (~23%) neutral, and 8943 (~61%) negative. The original model only trained the classifier on negative and positive sentiment, thus reducing the applicable data to a size of 11179 instances and used SKlearn’s train\_test\_split in order to partition and train the data on 90% of the total instances while saving 10% for testing purposes. For continuity sake, we also followed this approach when separating our training and testing data.

The original model cleaned these tweets by removing whitespace and carriage returns, http formatted url links, retweets symbols (RT), abbreviations (ie. don’t -> do not), non-ASCII characters, and downcasing all the text. The original model also decided that removing certain features would help improve the performance, and opted to remove stopwords from the english corpus in the NLTK library, standard forms of punctuation, and noisy/uninformative words like ‘gopdebate’ or ‘foxnews’ that were initially useful in gathering tweets. Additional curation methods used by the original model include tokenizing each word, and stemming these words using a Snowball algorithm included in NLTK’s stem library in order to create a “bag-of-words” feature space to be used in a Naive Bayes learning algorithm for label prediction.

In regards to our approach to creating and combining features with this pre-existing space, some approaches require different text processing: characters and words left-out in the original analysis were vital to the creation of features we planned to test. For instance, the original model removed punctuation like exclamation points or question marks; we thought these were important considerations when trying to establish irony in a tweet and should be added back into a separate feature list. After creating features enumerated in the following section, we combined these features individually with the original feature space for each individual instance with a similar bag-of-words representation in order to determine which combination added performance gains.

**Features**

We developed 5 feature subsets based on the existing literature about markers of irony in social media, particularly tweets.

Hashtag Content:

Kunneman et al. found 4 persistent markers of sarcasm through hashtags: #sarcasm; #irony, #cynicism, #not. Our data set contained none of those markers, but the study did suggest that non sarcastic hashtags were also one of the 4 most common markers of irony : the same held true of Maynard and Greenwood;s work, which focused on tokenizing hashtags and comparing based on content. We therefore decided to tokenize the hashtags, which was not done in the original data set, and see if adding in that information (both the existence of hashtags and their content) helped increase the accuracy of sentiment prediction.

Non-linguistic Markers:

Reyes et al. and Kunneman et al. both noted that non-linguistic markers such as exclamation points, question marks and other typographical markers in marking irony, so we added these back into the feature set.

Position:

Following Reyes et al.’s observation that sarcasm was most often marked in the first 4-7 words of a tweet, we cut out the first half of all tweets greater than 10 words to see if the second half of the tweet resulted in greater accuracy.

Content & Relationships between Words:

Multiple authors suggested that content knowledge is closely linked to ironic statements, so we sought to create a larger feature set by not removing the content specific language the original annotators removed.

We changed the tokenization and cleaning of the tweets in a separate feature space firstly by considering terms like ‘gopdebate’, ‘fox news’, and others originally taken out in the first model. From here, we also included url links, indication of retweets, and chose not to abbreviate words like “don’t” to “do not”.

Once we had a more robust set, we were able to examine the relationship between the features more clearly. Multiple studies suggested that irony resides in the relationship between words rather than the words themselves, so we wanted to investigate the value of adding bigrams to our feature set to better capture linguistic relationships. While Reyes et al. specifically measured the polarity of bigrams and n-grams, we wanted to see if simply adding bigams as features helped more accurately classify tweets by adding weight to such relations.

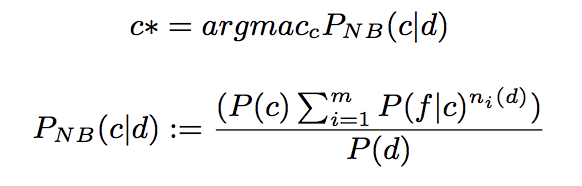
Kunneman et. al specifically noted the use of intensified and unintensified evaluative language in marking irony, so we manually identified all intensified and unintensified bigrams in our data set (see appendix A) and used the most commonly occurring ones as feature sets. While intensified language was a slightly stronger marker or irony, unintensified language also was a strong indicator, so we created a set with both.

Finally, we wanted to look for any relations marked by intensified or unintensified language as opposed to the specific bigrams that appeared in our training set, so we weighted the mere appearance of intensified and unintensified evaluations (since they are adjectives they would necessarily be part of a bigram) by adding them a second time to the unigram feature set.

Please note: we had a communication breakdown and did not create a trial that used Nagy’s original data preparation with just the addition of the bigrams in various permutations. Though we do think that the expanded set was needed to actually track relations since the relation was often linked to content, it would obviously have been best practice to see if adding bigrams to the initial set resulted in increased precision.

**Models**

Naive Bayes is a simple model which works well on text categorization. We use a multinomial Naive Bayes model. Class c∗ is assigned to tweet d, where



In this formula, f represents a feature and ni(d) represents the count of feature fi found in tweet d. There are a total of m features. Parameters P(c) and P(f|c) are obtained through maximum likelihood estimates, and add-1 smoothing is utilized for unseen features.

**Evaluation**

Confusion Matrix:

We used a confusion matrix to assess the prediction results of our classifier, as is common practice in measuring the accuracy of Naive Bayes. In our binary classification problem, the table had 2 rows and 2 columns. Across the top were the predicted class labels and down the side were the observed class labels. Each cell contained the number of predictions made by the classifier that fall into that cell. This table displayed both the class distribution in the data and the classifiers predicted class distribution with a breakdown of error types.

Precision:

The class of our dataset was not balanced as negative labels accounted for 79% among all negative and positive labels. In this case, accuracy could be misleading because of accuracy paradox, which means that a model can predict the value of the majority class for all predictions and achieve a high classification accuracy, the problem is that this model is not useful in the problem domain. Therefore, we calculated precision for both negative and positive labels.

Precision for negative labels is the number of true positives divided by the number of true positives and false positives, while precision for positive labels is the number of true negatives divided by the number of true negatives and false negatives.

**Results**

**Original Classifier (Nagy):**

POS Precision: 0.548

NEG Precision: 0.867

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 768 | 85 |
| POS | 117 | 103 |

**All Hashtags:**

POS Precision: 0.533

NEG Precision: 0.862

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 768 | 85 |
| POS | 123 | 97 |

**Presence of Hashtag:**

POS Precision: 0.504

NEG Precision: 0.833

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 793 | 60 |
| POS | 159 | 61 |

**Punctuation( ‘!’, ‘?’, ‘;’):**

POS Precision: 0.513

NEG Precision: 0.862

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 760 | 93 |
| POS | 122 | 98 |

**Second Half of Tweet:**

POS Precision: 0.482

NEG Precision: 0.856

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 752 | 101 |
| POS | 126 | 94 |

**BAG OF WORDS(UNIGRAM all characters)**

POS Precision: 0.702

NEG Precision: 0.868

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 812 | 41 |
| POS | 123 | 97 |

**BAG OF WORDS(UNIGRAM + BIGRAM)**

POS Precision: 0.762

NEG Precision: 0.866

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 824 | 29 |
| POS | 127 | 93 |

**BAG OF WORDS(UNIGRAM + Intensified Values Bigram)**

POS Precision: 0.704

NEG Precision: 0.871

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 811 | 42 |
| POS | 120 | 100 |

**BAG OF WORDS(UNIGRAM + Intensified Unigram)**

POS Precision: 0.676

NEG Precision: 0.87

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 805 | 48 |
| POS | 120 | 100 |

**BAG OF WORDS(UNIGRAM + Unintensified Value Bigram)**

POS Precision: 0.64

NEG Precision: 0.86

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 803 | 50 |
| POS | 131 | 89 |

**BAG OF WORDS(UNIGRAM + Unintensified Value Unigram)**

POS Precision: 0.627

NEG Precision: 0.859

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 800 | 53 |
| POS | 131 | 89 |

**BAG OF WORDS(UNIGRAM + BIGRAM +**

**Intensified Value Bigram + Intensified Value Unigram**

**+ Unintensified Value Bigram + Unintensified Value Unigram)**

POS Precision: 0.715

NEG Precision: 0.861

Confusion Matrix:

|  |  |  |
| --- | --- | --- |
| Actual/Predicted | NEG | POS |
| NEG | 818 | 35 |
| POS | 132 | 88 |

**Discussion**

Overall, we had the most success by creating a feature set of all content unigrams (as opposed to the set prepared by Nagy), and then adding in either all bigrams (a positive precision of .762 and negative precision of .866) or just the intensified value bigrams (a lower positive precision, at .706, but slightly higher negative precision of .871).

This runs counter to best practices in terms of limiting the number of features and avoiding having more features than examples because of the gaps caused. However, this may be a function of the complicated nature of irony. As Reyes et al. point out, there can be no “’computational silver bullet’ to identifying irony because it “cuts through every aspect of language, from pronunciation to lexical choice, syntactic structure, semantics and conceptualization” Reyes et al. and others include content in this list and much of the literature on sarcasm and irony agrees that one difficulty of identifying irony is that “[o]ne needs to have a good understanding of the context of the situation, the culture in question, and perhaps the very specific topic or people involved in the sarcastic statement (Maynor and Greenwood 2014). Multiple studies indicated that combining elements was the best way to identify irony. Since a significant number of the misidentified tweets involved ironic language adding in the content knowledge indicated by urls and hashtags, as well as the relation between words captured by bigrams offered a collection of features that taken together offered a way to capture a body of features, one or two of which occurred in each ironic tweet.

Using just the intensified value adjectives as a form of instance weighting resulted in slightly higher accuracy of negative sentiment, but decreased the positive sentiment accuracy. However, our data set had a significantly lower number of positive tweets, so having more positive tweets overall might have allowed us to create a more effective algorithm that could take advantage of the markers of irony.

One particularly striking pattern was that a number of the new false negatives or positives in our more accurate trials were actually ironic tweets that had been misidentified by the human annotators. This suggests our features were strongly indicated with irony and that the algorithm, given a baseline of some properly labeled sentiments for tweets with irony could exceed human annotators in accuracy.

Two inter-related assumptions shaped our hypothesis. The first was that just weighting the appearance of a/multiple sarcasm marker could correct for the surface meaning of a tweet. This itself laid on an assumption that we did not articulate: that ironic statements did not appear in equal ways and equal proportion between positive and negative tweets. That is, if the presence of a “...” and a bigram starting with “very” appeared equally in ironically positive tweets and ironically negative tweets and those tweets were equally distributed, adding them to the feature set would not result in a useful increase in mutual information. The fact that we were able to get an increase in our accuracy by adding in a combination of features suggests that these assumptions held true. It does seem to be feature specific and not directly correlated to the utility of a feature in identifying sarcasm, however: Reyes et. al found unitensified value markers to be stronger indicator of the presence of sarcasm, but we found intensified bigrams to be more useful

**Conclusion**

We focused on features identified to be linked to irony in earlier studies: the existence and content of hashtags, special characters, the first half of tweets, and content and relations between words. On their own,adding features and hashtags to the feature set resulted in very slight decreases in precision, but with shifts in what was and wasn’t accurately classified that suggested these elements each had some correlation with sentiment.

By increasing the feature set to include all content and adding in bigrams, we were able to increase accuracy of both positive and negative tweets. Adding just the intensified value bigrams (strongly associated with irony) made us significantly more precise in identifying positive sentiment tweets. This also had the effect of actually “catching” some mis-annotated ironic tweets, suggesting both that ironic language was indeed linked to these features and that the machine algorithm surpassed crowd-sourced human annotators in identifying them.

These results suggest both that, as per Reyes et al. and Kunneman et al. irony appears in multiple features rather than in any consistent manner. In their studies, it was best identified by combining features: in our study, adding content and relations resulted in greater accuracy of sentiment prediction. Further studies could examine adding in punctuation, character markers, and hashtags to that set: our results of those alone decreased accuracy, but the shifting classifications suggests that each may be adding some information about ironic tweets and their sentiment.

We would also, in the future, look to evaluate using multifold variable testing as opposed to a 90/10 split. Overfit is an issue to always be aware of, both with the temporal nature of a twitter set and with the possibility of human annotators incorrectly identifying irony in the first place.

Any increase in accuracy of political tweets is key in the applications for which such tweets are used, such as forecasting results and microtargeting populations of areas for get out the vote activities.

While our focus is on sentiment analysis of political tweets, better analysis of the sentiment of tweets with sarcastic or ironic language will have larger applications: as Reyes et. al point out “[l]arge companies have the most to gain from the appreciation of irony in social media, since these media are increasingly being used to comment on products and services and thereby encourage or discourage new customers. If a company can look beyond the distortional effect of irony, it can more accurately gather valuable marketing knowledge from the opinions of its users” (257).

It is also possible that further we do will help further advance the work of accurately classifying sarcastic language, particularly in political contexts. Classifying such language is always a challenge--- but as noted above, essential to properly understand the sentiments and attitudes connected to content that appears in social media outlets such as twitter.

**Works Consulted**

Bovet, A., Morone, F., & Makse, H. A. (2018). Validation of twitter opinion trends with national

polling aggregates: Hillary Clinton vs Donald Trump. *Scientific Reports*, 8(1), 8673. doi:10.1038/s41598-018-26951-y

Bowman, Karlyn. (Spring 2019). The trouble with polling. *National Affairs*. Vol 39. Retrieved

through <https://www.nationalaffairs.com/publications/detail/the-trouble-with-polling>

Kondik, Kyle. (December 2017). Can twitter predict elections? *Sabato’s Crystal Ball* (The

University of Virginia Center for Politics). Retrieved from <http://www.centerforpolitics.org/crystalball/articles/can-twitter-predict-elections/>

Kunneman, F., Liebrecht, C., van Mulken, M., & van den Bosch, A. (2015). Signaling sarcasm:

From hyperbole to hashtag. *Information Processing & Management*, 51(4), 500–509. https://doi-org.libproxy.lib.unc.edu/10.1016/j.ipm.2014.07.006

Mascaro, C. M. & Goggins, S. P. (2015). Technologically mediated political discourse during a

nationally televised GOP primary debate.” *Journal of Information Technology & Politics*, 12(3), 252–269. https://doi-org.libproxy.lib.unc.edu/10.1080/19331681.2015.1071687

Maynard, D.G. and Greenwood, M.A. (Accepted: 2014) Who cares about sarcastic tweets?

Investigating the impact of sarcasm on sentiment analysis.” In: *LREC 2014 Proceedings. Language Resources and Evaluation Conference* (LREC), 26-31 May 2014, Reykjavik, Iceland. ELRA . Retrieved from: http://eprints.whiterose.ac.uk/130763/1/sarcasm.pdf

Nagy, Peter. NTLK sentiment analysis. *Kaggle.com.* Retrieved from <https://www.kaggle.com/anebzt/nltk-sentiment-analysis>

O’Connor, B., Balasubramanyan, R., Routledge, B. R. & Smith, N. (2010). From tweets to

polls: Linking text sentiment to public opinion time series.” *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media.* 122–129. DOI:citeulike-article-id:7044833

Pocaro, Guiseppe and Muller, Henrik. (November 16 2016). Tweeting brexit: narrative

building and sentiment analysis. *Bruegel.org*. Retrieved from <http://bruegel.org/2016/11/tweeting-brexit-narrative-building-and-sentiment-analysis/?utm_content=buffer9c7e8&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer+(bruegel)>

Poe, Nathan (August 11, 2005). "Big contradictions in the evolution theory, page 3".

christianforums.com. Archived from the original on January 14, 2017. Retrieved January 14, 2017.

Reyes, A., Rosso, P., & Veale, T. (2013). A multidimensional approach for detecting irony in

twitter.” *Language Resources and Evaluation,* 47(1), 239-268. Retrieved from <http://www.jstor.org/stable/42637262>

Tenuto, J. (2015, December 17). “What the GOP debate taught us about machine learning.”

[Blog post]. Retrieved from <https://www.figure-eight.com/what-we-learned-about-our-machine-learning-classifier-during-gop-debate/>

Zimmer, Michael and Nicholas John Proferes. (2014). A topology of twitter research:

disciplines, methods, and ethics." *Aslib Journal of Information Management*, Vol. 66 Issue: 3, pp.250-261, https://doi.org/10.1108/AJIM-09-2013-0083

Appendix A: Bigrams and Manual Classification of Intensified and Unintensified Value Bigrams

Please See <https://docs.google.com/spreadsheets/d/1Wq8gx9IWAlHLVXGJe16g0UKR84Y_BsPt-lVVqVdEeYw/edit?usp=sharing>

Appendix B: Results of Each Trial

All Hashtags:

Overall, this slightly decreased our accuracy. The decrease was entirely in positive sentiment tweets, with six more false negatives.

One interesting pattern we noted in adding hashtags was the way they “flipped” sentiment, even without other features

|  |  |
| --- | --- |
| Shift | Tweet |
| False Positive changes to True Negative: (Total: 2) | Original Tweet: Unknown, but must have been been tokenized and cleaned to the point where nothing is included in the bag of words representation.  BOW:  [] |
| True Negative changes to False Positive: (Total: 12) | Original Tweet:  RT @libertyjibbet: Hey .@realDonaldTrump You know who didn't whine about "unfair" questions? @CarlyFiorina Grow a set why don't ya'. #GOPDebate  BOW:  ['hey', 'realdonaldtrump', 'know', 'whine', 'unfair', 'question', 'carlyfiorina', 'grow', 'set', 'ya'] |
| True Positive changes to False Negative: (Total: 8) | Original Tweet:  Unknown, but must have been been tokenized and cleaned to the point where nothing is included in the bag of words representation. |

Two things to note here: the hashtag #GOPDebate, which was the shared factor among all tweets and thus what would “reappear” in the tweets which had appeared as unknown, seems to have a stronger correlation with negative, resulting in some true negatives and more false negatives. However, oddly, a clearly negative tweet (“Hey .@realDonaldTrump You know who didn't whine about "unfair" questions? @CarlyFiorina Grow a set why don't ya'. #GOPDebate”) was falsely reclassified as positive by adding the same hashtag.

This does suggest that hashtags have a fairly strong effect on the predicted polarity of tweets. Though not directly related to irony, since Kunneman et. al note that the presence of any hashtag is connected to irony, it does seem possible that adding hashtags into the feature set does reflect the polarity shift created by irony and thus it might be possible, with further adjustments, to tweak the algorithm in such a way that this could be a useful marker on its own. Note that when combined with the other features we added in, it did result in greater accuracy, suggesting that some in conjunction with other features, using hashtags can increase accuracy, perhaps because of their strong relation to polarity that “flips” the effect of other features.

Punctuation:

Adding in the punctuation resulted in a slight decrease in accuracy for both positive and negative accuracy.

|  |  |
| --- | --- |
| Shift | Tweet |
| Example where False Negative changes to True Positive: (Total: 10) | Original Tweet: RT @JessicaValenti: as an italian-american i appreciate walker's ability to talk with his hands. that's the nicest thing i have to say #GOP  BOW:  ['jessicavalenti', 'italian-american', 'appreciate', 'walker', "'s", 'ability', 'talk', 'hands', "'s", 'nicest', 'thing', 'say', 'jessicavalenti', 'italian-american', 'appreci', 'walker', "'s", 'abil', 'talk', 'hand', "'s", 'nicest', 'thing', 'say'] |
| True Negative changes to False Positive: (Total: 18) | Original Tweet: RT @TheJimHughes: There will be a lot of jobs available if Bush serves 2 terms, because most Conservatives will have committed suicide. #GO  BOW:  ['thejimhughes', 'lot', 'jobs', 'available', 'bush', 'serves', '2', 'terms', 'conservatives', 'committed', 'suicide', 'go', 'thejimhugh', 'lot', 'job', 'avail', 'bush', 'serv', '2', 'term', 'conserv', 'commit', 'suicid', 'go'] |

In both cases, the addition of punctuation shifted the sentiment to positive. The first tweet suggests a likely reason: the existence of a second sentence suggests a shift in sentiment, which seems to happen more frequently in positive sentiments (thus the comma in the second representative tweet carries a heavier positive correlation than the negative correlation of “suicide.”)

However, the first tweet seems to have been misclassified: it should be negative and was classified as positive. This points out the problems inherent in training an algorithm to detect sentiment in ironic statements such as that tweet: humans are not particularly good at detecting irony (Poe’s Law is a famous statement of this problem, arguing that “without a winking smiley or other blatant display of humor, it is utterly impossible to parody a Creationist in such a way that someone won't mistake for the genuine article” (2005)). This adds another layer to the difficulties of detecting the sentiment of tweets with irony: if the tweets were originally misclassified, the effect of the polarity is not being accurately correlated.

The very minimal loss of accuracy along with the fact that accuracy was gained in some placed while lost in others suggests that punctuation does shift what gets read as positive and negative and thus could be useful in conjunction with other features.

Second Half of Tweet:

Decreased on both positive and negatives. We did run into a question of which tweets to treat this way: some tweets were so short that cutting out the first half would leave one word, or just a hashtag. We decided to simply remove the first half of all tweets, so it’s possible that this could be useful if we developed a sliding scale between length and words removed, but given the fairly large drop in accuracy in this crude truncation, it does not seem like we would get much, or any, value for the work involved to develop such a scale. Because of this, we did not closely examine the different classes.

Bag of Words (Unigram):

Unlike the original classifier developed by Nagy, we created a bag of words without removing ‘gop','debate','gopdeb','gopdebate','gopdebates','fox','news','foxnew','foxnews', 'amp’ as stop words. We did not convert abbreviations(ie. don’t -> do not), or remove http formatted url links and retweets symbols (RT). This significantly increased our precision in classifying both positive and negative tweets.

|  |  |
| --- | --- |
| Shift | Tweet |
| True Positive changes to False Negative: | Original Tweet: My response when @megynkelly tried to call @realDonaldTrump a sexist: "CAN'T STUMP THE TRUMP!" #GOPDebate |
| False Negative changes to True Positive: | Original Tweet:RT @tedcruz: If elected, on my first day as President, I'll rescind every illegal and unconstitutional executive order enacted by Pres. Oba |
| False Positive changes to True Negative: | Original Tweet: RT @Jayedilla: Surprise surprise, @GovMikeHuckabee thinks #trans life is a "social experiment" #GOPDebate |

The new true negative is an instance of an ironic statement (marked by the surprise surprise) being correctly classified when content was added back in to the feature set, suggesting that content does have a correlation with ironic statements (the new true positive suggests that this increase was also true in non ironic statements, suggesting that content may have higher mutual information in political data than in other sets).

The new false negative is an intriguing case: is the tweet actually positive? This is one of the cases in which irony is difficult to detect even for a human annotator: the fact that such tweets are changed in sentiment suggests that this feature set certainly captures some elements linked to irony.

Unigrams+ Bigrams

Adding in bigrams increased the true negatives slightly more, though it decreased the accuracy of positive sentiment analysis.

|  |  |
| --- | --- |
| Shift | Tweet |
| True Positive changes to False Negative: | Original Tweet: RT @LadySandersfarm: We're sick of political correctness, it's killing the heart and soul of this country. #GOPDebate #Trump2016 |
| False Positive changes to True Negative: | Original Tweet: RT @RWSurferGirl: Jeb Bush reminds me of elevator music. You hear it but you don't listen. #GOPDebate #GOPDebates |

The new true negative is ironic and now accurately classed. Adding in bigrams allowed for better weighting of relationships between words, one of the prime markers of irony. We can see that this can lead to misclassifications in the case of the false negatives, in which the strong connected polarities lead this to be read as negative. Again, though, the size of the positive corpus makes discussions of accuracy difficult.

Unigrams and Intensified Value Bigrams

This slightly increased the accuracy of true positives, while very slightly decreasing the true negatives.

|  |  |
| --- | --- |
| Shift | Tweet |
| Example where True Negative changes to False Positive: | Original Tweet: RT @AmyMek: We all owe @realDonaldTrump a huge thank you 4 exposing to the world@megynkelly, Chris Wallace, &amp; @FrankLuntz are true Rhinos! |
| True Positive changes to False Negative: | Original Tweet: RT @tedcruz: Over the past six years we've seen the results of the Obama-Clinton foreign policy. Leading from behind is a disaster. #GOPDeb |
| False Negative changes to True Positive: | Original Tweet: RT @RWSurferGirl: Trump has got it right, nobody would talk about immigration, not untill be brought it up. #GOPDebate #GOPDebates |
| False Positive changes to True Negative: | Original Tweet: RT @monaeltahawy: Has any candidate received a word from God?! Presidential hopefuls?! This is America?! Christian Brotherhood of America |

Interestingly, almost every misclassified tweet had some element of irony in it, again suggesting that we are creating a feature set that addresses the problem we wish it to. The one exception is the new false negative “Original Tweet: RT @tedcruz: Over the past six years we've seen the results of the Obama-Clinton foreign policy. Leading from behind is a disaster. #GOPDeb” in which the lack if intensified values bigrams seems to have led to classifying as negative--perhaps intensifed value bigrams are more strongly correlated with positive sentiments?

Unigrams and Intensified Value Unigrams

This slightly decreased the true negatives and increased the false positives from the unigrams on their own. Thus it appears that the existence of the intensifier is not a useful marker of sentiment without being attached to specific content.

While the false positive here was led by the “better ratings,” this was an ironic statement, suggesting that looking for bigrams did lead to changes in the sentiment classification of ironic tweets, although incorrectly in this case: however, overall, the accuracy increased suggesting that the correlations were meaningful.

The true positive was another misidentified tweet: it certainly reads as negative. Thus we can see another case in which the lack of intensifying bigrams, linked to irony, actually led to better classification of a very straightforward tweet.

UNIGRAM + Unintensified Value Bigram

This also showed a decrease in true negatives and of true positives, suggesting that in this data set, unintensified bigrams are not a useful predictor of sentiment: this aligns with the observation that intensified value bigrams offered a slight overall gain in accuracy, through increasing the amount of true negatives.

UNIGRAM + Unintensified Value Unigram

Unsurprisingly, based on the above, this resulted in lower values

UNIGRAM + BIGRAM + Intensified Value Bigram + Intensified Value Unigram + Unintensified Value Bigram + Unintensified Value Unigram

Somewhat surprisingly, instance weighting both intensified and unintensified value unigram words and the specific bigrams they created offered a relatively high accuracy; though slightly lower than the unigrams plus bigrams or unigrams plus intensified bigrams, it was higher than other instances, suggesting that the presence of the intensified bigrams offers a significant value in predicting sentiment.

|  |  |
| --- | --- |
| Shift | Tweet |
| Example where True Negative changes to False Positive: | Original Tweet: RT @BraunKen: Oh my. Reasonable answere by #Kasich2016 on gay question, and applause by crowd. Maybe a new GOP? #GOPDebate |
| True Positive changes to False Negative: | Original Tweet: Hey .@realDonaldTrump You know who didn't whine about "unfair" questions? @CarlyFiorina Grow a set why don't ya'. #GOPDebate |
| False Positive changes to True Negative: | Original Tweet: The reality TV show is over &amp; I'm trying to get back to the day but Im so obsessed with the hilarious memes that were generated #GOPDebate |
| False Negative changes to True Positive: | Original Tweet: @PamelaGeller @AnnCoulter Kim/Kanye are great examples of Hillary's "useful idiots" that @RealBenCarson was referring to in #GOPDebate . |

Another interesting case in which the new false positives and negatives actually were incorrectly annotated by the human annotators, suggesting that further weighting may actually catch the wide variety of approaches to irony.