Identifying Fake News through Stance Detection

Elise Georis and Isabelle Ingato

May 15, 2017

Abstract

Fake news has become increasingly prevalent and problematic. As a first step towards automatically detecting it, this paper addresses the sub-problem of stance detection. More specifically, we present a variety of approaches for determining the relationship between headline and body texts before quantitatively evaluating their relative merits. Our results have largely exceeded our initial expectations and point to promising areas for future exploration.

1 Introduction

In an era dominated by social media, widespread Internet access, and a tense political climate, fake news has erupted. The *New York Times* defines "fake news" as a false story created to attract and mislead readers [25]. One can also think of it as a story that contradicts more honest news sources.

As such false stories continue to spread, they become increasingly difficult to detect. Moreover, the use of deceptively high-quality writing or of convincing (but faulty) data can lend a sense of legitimacy to any source. We thus turn to natural language processing in the hopes that computers can identify falsities that humans overlook.

This hope is bolstered by research suggesting that machine learning algorithms trained to identify and demote stories with "clickbait titles" are more effective than humans asked to manually flag suspicious articles [3]. Confidence in this technology is consequently growing, particularly as companies like Facebook rely more heavily on machine learning techniques than on user reporting in the fight against fake news.

2 Motivation

Fake news has tangible consequences. These include a misinformed and divided citizenry, a widespread mistrust of the mainstream news media, and even a spike in violence as people fail to distinguish between factual and falsified stories.

For example, in December 2016, a man attacked a pizzeria with a firearm after reading a piece of fake news suggesting that the restaurant was involved in crime [25]. His actions emphasized the potential seriousness of fake news, as well as the need to halt its spread. As Tavernise (2016) aptly summarizes, "Fake news, and the proliferation of raw opinion that passes for news, is creating confusion, punching holes in what is true, causing a kind of fun-house effect that leaves the reader doubting everything. . . " [25, n.p.].

Motivated by these and other problems, we aim to improve upon existing technologies for fake news detection. More specifically, we focus on stance detection as a first step towards the ultimate goal of classifying content as valid or invalid.

3 Problem Definition

3.1 Stance Detection

For the purposes of this project, we define "stance detection" as the classification of the relationship between a headline and a body text in one of four categories [4]:

- **Agree**: The body text agrees with the claim made in the headline.
- **Disagree**: The body text disagrees with the claim made in the headline.
- **Discuss**: The body text discusses the content in the headline but neither agrees nor disagrees with the claim that it makes.
- **Unrelated**: The body text is unrelated to the content in the headline.

Software that can determine the perspectives of different texts and the relationships between them will be invaluable when addressing the larger challenge of automatic fake news detection. For example, when determining whether a given text is misleading or false, it could be helpful to know whether it bolsters or disputes the claims made by more trusted news sources.

3.2 The Fake News Challenge

We tackled stance detection in the context of the Fake News Challenge (FNC), whose professed goal "... is to address the problem of fake news by organizing a competition to foster development of tools to help human fact checkers identify hoaxes and deliberate misinformation in news stories using machine learning, natural language processing and artificial intelligence" [4, n.p.]. We chose to do so in part because the FNC supplied a labeled corpus of headline-body pairs. It also provided a barebones sample implementation of a stance detection program, thereby giving us a starting point, as well as a baseline accuracy score to beat.

4 Literature Review

We studied several previous attempts at stance detection to inform the design of our own program. We present the most interesting findings below.

For example, Wojatzi and Zesch (2016) employed stacked classifiers to detect stance in social media postings [27]. Interestingly, in their preprocessing stage, the authors lemmatized plurals so that spelling errors stood out. (We eventually chose to do the same, reasoning that errors in a headline may be reflected in a related body.) Next, for each body text, the researchers counted the number of modals (e.g., "can" or "shall") and the number of words signifying negation (e.g., "none" or "never"). They also counted the number of exclamation marks, question marks, instances of punctuation overuse (e.g., "?!"), and instances of the word "if" in each headline and body text. Wojatzi and Zesch made these decisions based on linguistic research suggesting that modals, conditionals, and certain patterns of punctuation may point to stance-taking behavior. Finally, the pair also considered the presence or absence of common bigrams and trigrams in each body text, as well as stance lexicons for agreement and disagreement. After performing an ablation test, the authors found that these stance lexicons had the greatest impact on all types of targets, which in turn encouraged us to try the same form of feature extraction in our own program.

Next, Krejzl, Hourova, and Steinberger (2017) used MaxEnt to detect stance in English tweets [7]. While Twitter represented a slight departure from our focus on the news, the domains overlapped enough to make several of the researchers' techniques helpful in our own project. More specifically, although we initially doubted that Twitter symbols (e.g., "@" and "#") are relevant to news, a closer examination of our data set revealed that journalists actually *do* use these signifiers—as well as hyperlinks—in their headlines and body texts. Therefore, we ultimately adopted the tactic employed by [7] and counted the number of "#" symbols, "@" symbols, and hyperlinks in each headline and body text. In addition to these more specific features, the researchers evoked the aforementioned stance lexicons used by Wojatzi and Zesch when they built stance-predictive keyword lists containing all of the unigrams that appear at least four times more often in bodies

belonging to one stance than the other. They consequently achieve an F1 score of .63 for known targets and .42 for unknown targets, suggesting that syntactic analysis may help to boost performance—a lesson we retained for our own approach.

We also consulted the work of Augenstein *et al.* (2016), who performed semi-supervised training on bag-of-words autoencoders using unlabelled tweets containing all possible targets such that they could generate feature representations of those tweets [1]. After the feature representations were generated, they then trained a logistic regression classifier with L2 regularization on labelled autoencodings of tweets, combined with several other features, including whether or not the exact target is contained in the tweet—a binary feature that they called `targetContainedInTweet`. This target is simply the focus of the stance. For our own purposes, however, that focus may not always be easy to identify, since the target may not be in the training data, or even explicitly mentioned in the text. We thus morphed `targetContainedInTweet` into `headlineContainedInBody`, which we hoped would have more success in our wordier domain.

In yet another past experiment, Sobhani, Mohammad, and Kiritchenko (2016) predicted tweet stances using a linear kernel Support Vector Machine trained on individual targets, alongside word and character n-grams, word2vec embeddings, sentiment analyses, and a count feature for the number of words with a character repeated more than two times (e.g., "yesss") [23]. They also tried to use an oracle model that mapped sentiment labels of tweets directly to stance labels. They found that the most successful classifier combined n-gram features and the sentiment-based oracle, and they achieved an impressive F1-score of 0.78.

Similarly, Patra, Das, and Bandyopadhyay (2016) achieved an F1-score of 0.6 by using Support Vector Machines to detect stance in tweets [10]. Perhaps more interestingly, though, they also used the Stanford Parser to determine word dependencies in their texts. For example, given the line "I support campaign," they would identify the word "campaign" as the direct object of the word "support." If (1) the direct object appeared in a favor bag related to the target (i.e., a pre-identified set of favorable or positive-associated target words created with RitaWordNet), and if (2) the verb of that direct object were more positive than negative according to SentiWordNet, then their algorithm would add 1 to a count feature called "favor_positive" [10, p. 442] To combat any noise when doing so, they preemptively implemented basic negation detection using keywords like "not."

Finally, we studied the baseline implementation created by the team behind the Fake News Challenge itself [5]. The authors used the following features: (1) character-grams (2, 4, 8, 16) and n-grams (2-6), (2) binary token co-occurrences between tokens in the headline and body, (3) the ratio of word overlap (intersection) between types of words in the headline and body to all types (union) in both headline and body, and (4) the presence or absence of words that signify refutation (e.g., "fake" or "bogus"). They evaluated their baseline on a corpus of nearly 50,000 articles using a `GradientBoostingClassifier`, and they rated their accuracy such that 25 percent of the score comes from binary (i.e., "related" v. "unrelated") classification accuracy, while 75 percent comes from the multi-class case (i.e., "agree" v. "disagree" v. "discuss" v. "unrelated"). Ultimately, they achieved an accuracy score of about 0.79.

This was the number that we hoped to beat in our own implementation.

5 Methods

5.1 Preprocessing

We performed the same preprocessing steps on both the headline and the body after performing standard tokenization and part-of-speech tagging on the texts using the NLTK tools [2]:

1. **Lowercase all tokens**
2. **Remove stopwords**

   It is standard in the literature to remove non-content-words, such as "the," because they

are minimally informative when comparing two pieces of text. We used NLTK's English stopwords corpus [2] to do so.

3. **Lemmatize all tokens**

   In [7], the authors suggest that it is important to lemmatize tokens not only to improve recall in the sense of comparing documents but also to highlight spelling mistakes and to determine whether spelling is consistent in the headline and body. (We noticed that the FNC baseline implementation employs a similar step in its own preprocessing [4].) First, we experimented with using the `RegexpStemmer` class from [2] to perform simple regular expression-based stemming (e.g., removing "-ing" suffixes) on tokens with lengths of at least four. However, we achieved stronger results when we employed the WordNet `Lemmatizer` [11], since this tool goes so far as to morphologically analyze the tokens.

4. **Replace keywords belonging to specific categories with special tokens**

   In [27], the authors note that conditionals, negation words, and modals often indicate stance-taking behavior. In response, they replace those words with special tokens that group such words by type for later analysis. Based on their success, we used a similar approach: we replaced "not," "no," "none," "nor," "never," "nobody," "neither," "nowhere," and words ending in "n't" with the special "NEGATION" token, and we replaced "can," "should," "may," "might," "must," "shall," "should," "will," and "would" with the special "MODAL" token.

   We also made the novel decision to replace all words that have the part-of-speech tag "CD"—indicating a cardinal, or number—with the special "CARDINAL" token. We reasoned that texts containing numbers may be less likely to be opinion-based (i.e., "agree" or "disagree") and also less likely to be nonsensical (a category under which many of the "unrelated" articles fall).

5.2 Feature Extraction

Next, we combined linguistic knowledge with other heuristics to consider both published and novel features for stance detection. In most of the following cases, a "feature" is a pair consisting of the feature values for the (1) headline and (2) body individually.

1. **Number of question and exclamation marks**

   This set of features, which we obtained by counting the appearance of question marks in the (1) headline and (2) body, as well as exclamation marks in the (3) headline and (4) body, was published in [27]. It has been reasoned that these kinds of punctuation are most often used in opinion-based text. This feature set was particularly useful in conjunction with the next feature.

2. **Frequency of overuse of punctuation**

   By counting the appearance of non-overlapping, back-to-back punctuation symbols (e.g., "?!" or "!!!!"), we determined the frequency of overuse of punctuation in the (1) headline and (2) body. This feature set was also employed in [27]. Although such overuse of punctuation is more frequent in social media than in our own domain of online news, we witnessed an interesting discrepancy in their use in the headline as opposed to the body of an article in instances labeled as "unrelated."

3. **Frequency of NEGATION words**

   As mentioned, negations, modals, and conditionals often point to stance-taking behavior [27]. Having replaced negation words with the NEGATION token in the preprocessing stage, we simply counted the appearances of this token in the (1) headline and (2) body.

4. **Frequency of MODAL words**

Similarly, based on the same research from [27], we counted the numbers of MODAL tokens in the (1) headline and (2) body.

5. **Frequency of conditional statements**

   Once again again based on the work in [27], we approximated the numbers of conditional statements in the (1) headline and (2) body by counting statements beginning with the word "if."

6. **Frequency of CARDINAL words**

   We had a novel idea to count the numbers of CARDINAL tokens in the (1) headline and (2) body. After all, we reasoned, if a headline mentions a number, then a number should appear in the body, so long as the headline and body are related.

7. **Initial trigrams**

   As demonstrated in [7], we used the first three words of the (1) headline and (2) body as a feature pair.

8. **Final trigrams**

   Along the same vein, we recorded the last three words of the (1) headline and (2) body as another feature pair [7].

9. **Length after preprocessing**

   [7] also showed that the text length after preprocessing the (1) headline and (2) body is a useful feature when comparing the relative complexity of the claims made in the headline and body.

10. **`headlineContainedInBody`**

    In [1], the authors found that `targetContainedInBody`, a simple binary feature, outperformed an advanced autoencoder at the stance detection task. We thus adapted this feature for our purposes to determine whether the exact preprocessed headline appears, word-for-word, in the preprocessed body.

11. **Percentage of word overlap**

    To measure the similarity between the headline and body, we calculated the percentage of word types that are shared between the headline and body, out of all possible word types in either one or both of the texts. This can be expressed as follows:

$$\frac{\text{Headline Tokens} \cap \text{Body Tokens}}{\text{Headline Tokens} \cup \text{Body Tokens}}$$

12. **Paivio Meaningfulness**

    Paivio Meaningfulness refers to a rating between 100 and 700 that is assigned to each word in the MRC Psycholinguistic Database [8]. (The original ratings were based on the number of related words that human subjects identified for a given term.) To take advantage of these measurements, we normalized all ratings in [8] before computing the average meaningfulness of the words in the (1) headline and (2) body, separately. We reasoned that similar average meaningfulness ratings for the headline and body may signal relatedness not only in content but also in style.

13. **Familiarity**

    The MRC database [8] also includes ratings for the familiarity (i.e., simplicity) of words. After normalizing and averaging across ratings for all words in the (1) headline and (2) body, we included this feature pair for each instance. We hypothesized that more familiar words will appear more frequently in opinion pieces (and, ultimately, in fake news).

14. **Sentiment**

    Many papers, including [23], have used the sentiment scores of the headline and body to predict stance. The idea is that headlines and bodies that are either both positive or both negative are more likely to express similar opinions than texts that exhibit two different sentiments. We thus used SentiWordNet [22] to obtain the positivity, negativity, and objectivity rating of a word and its part-of-speech, which were combined in a sentiment synset. We then averaged over all words in the (1) headline and (2) body to obtain the normalized positivity, negativity, and objectivity feature scores.

15. **Presence of words from a Unigram Stance Lexicon**

    In [27], the researchers created unigram stance lexicons, which contained only those words that appeared at least four times more frequently in articles belonging to one particular stance than in articles belonging to any of the other stances. We pre-computed these stance lexicons for our "agree" and "disagree" labels only, and we generated a binary feature vector for each (1) headline and (2) body that records the presence or absence of words from those lexicons.

16. **Most common bigrams and trigrams**

    The researchers from [27] also determined the most common bigrams and trigrams in their corpus, and, for each headline and body, generated a binary feature vector recording the presence or absence of those key bigram and trigram phrases in the texts. We adapted this to our own implementation by looking at the 100 most common bigram and trigram phrases in our corpus.

17. **Semantic frame intersection**

    An original contribution of our work involved the use of dependency parsing information obtained through SpaCy [24], as well as the interface to the FrameNet dictionary [13], to evaluate the semantic frame intersection between a headline and body. A semantic frame is "a description of a type of event, relation, or entity and the participants in it" [26, n.p.]. After performing dependency parsing on the headlines and bodies, we generated a list for each that contained pairs of each noun chunk and its related semantic frame. We did so under the hypothesis that noun chunks would generally appear with the same semantic frames in the headline and body if the texts were related. To that end, we counted the number of pairs in the headline that were missing from the body, and vice versa.

    For example, consider the phrase "Hillary Clinton loses the election." "Loses" is the frame-evoking verb, while "Hillary Clinton" and "the election" are the frame elements. We would thus identify two pairs: (Hillary Clinton, FINISH_COMPETITION) and (the election, FINISH_COMPETITION). We would then expect for at least some of these pairs (i.e., their semantic references) to occur again in the body. However, we would not assume that the exact same *lexical choices* were made. Rather, we might expect the FINISH_COMPETITION semantic frame to also be evoked by words that are related to "loses," such as "victory," "win," or "draw."

18. **Cosine similarity between headline and body keywords**

    We reasoned that the summarized version of the body should be similar to the headline if the pair are related. Therefore, we used the Gensim implementation of the TextRank automatic summarization technique to extract keywords from the body [12]. The TextRank method is unsupervised and organizes a text into a graph, in which vertices represent textual units and edges represent the similarities between such units. Its goal is to find important phrases that are "central"—that is, related to many other words. We thus set the length parameter of the keyword extractor [12] equal to the number of tokens in the headline. After extracting the body's keywords, we then calculated the cosine similarity between the weighted bag-of-words token vectors for the headline and the keywords of the body.

19. **Mismatch between Named Entities**

    We further hypothesized that related headlines and bodies contain many of the same named entities, including people, organizations, and locations. Thus, we used the Stanford Named Entity Tagger [6] to determine how many named entities in the headline were missing from the body, and vice versa.

20. **Frequency of quotation marks**

    Another novel feature extraction method that we designed involved calculating the frequency of quotation marks in the (1) headline and (2) body. Our belief was that, if quoted text appears in a headline, then it must appear in the body, so long as the two pieces are related. Also, bodies that frequently use quotes may be more discussion- and fact-based rather than opinionated.

21. **Frequency of adjectives**

    We also reasoned that adjectives can signify opinions in texts, so we separately counted the number of regular adjectives (identified by the Penn Treebank "JJ" part-of-speech tag), comparative adjectives ("JJR" part-of-speech tag), and superlative adjectives ("JJS" part-of-speech tag) in the headline and body.

22. **Frequency of social media tokens and links**

    Although we were looking at articles and not social media posts, we learned that many of our given news sources embed social media tokens within more formal articles. We thus decided that including social media tokens may be a stance-taking indicator, while links and citations may be predictive of more factual body texts.

23. **Frequency of personal pronoun usage**

    We also believe that personal pronoun usage is an indicator of stance-taking behavior. Thus, we counted the usage of the words "I," "me," "we," "us," "our," and "my" in the (1) headline and (2) body.

5.3 Feature Selection

We used the ANOVA feature selection method to select the features which were responsible for the highest proportion of variance. We treated the number of features to be selected as a hyperparameter and allowed [21] to automatically tune it.

5.4 Classification

We experimented with five distinct classification methods:

1. Support Vector Machines (SVM) with RBF Kernel [19]
2. Logistic Regression with L2 Penalty [16]
3. Random Forest [20]
4. KNeighbors [18]
5. Gradient Boosting [15]

We used the Sklearn implementations of all five methods [15] [16] [18] [19] [20]. Altogether, we found that the two ensemble methods, Random Forest and Gradient Boosting, outperformed the others.

6 Data and Other Tools

We also used the following resources.

1. Fake News Challenge (FNC-1) Corpus [4]
2. FNC-1 Baseline Implementation [**?**]
3. Natural Language Toolkit (NLTK) [2]
4. StanfordNERTagger [6]
5. SpaCy [24]
6. SentiWordNet [22]
7. WordNet [11]
8. FrameNet [13]
9. NumPy [9]
10. Sklearn [14]
11. Gensim [12]

7 Experiments and Results

7.1 Methods of Evaluation

We trained and tested our algorithm on the first 1000 articles of the corpus. (The time constraints of our feature extraction prohibited us from using a larger number of articles.) We then performed 10-fold cross validation using Sklearn's `StratifiedKFold` class [17] to ensure the most reliable results. We finally evaluated the results based on the following metrics.

7.1.1 Average Precision

Precision, or Positive Predictive Value, is defined as

$$\frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Positives}|}$$

and is equal to 1, minus the False Discovery Rate. In the binary case, we treat "unrelated" as the positive class, while, in the multi-class case, we average over all the classes being considered as the positive class. Precision is useful for interpreting how careful our classifier is when labeling articles as belonging to a certain positive class. In the case of stance detection (and even more in fake news detection as a whole), precision is extremely important because labeling an article as, for instance, the positive class "fake" has unfortunate consequences if the label turns out to be false.

7.1.2 Average Recall

Recall, or Sensitivity, is defined as

$$\frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Negatives}|}$$

As before, we treat "unrelated" as the positive class in the binary case. Recall helps us interpret how responsive our classifier is when identifying and labeling the members of the positive class.

7.1.3 Average F1 score

F1 is the harmonic mean of Precision and Recall and so is defined as

$$\frac{2 * |\text{True Positives}|}{2 * |\text{True Positives}| + |\text{False Positives}| + |\text{False Negatives}|}$$

7.1.4 Average Accuracy

Accuracy is defined as

$$\frac{|\text{True Positives}| + |\text{True Negatives}|}{|\text{Positives}| + |\text{Negatives}|}$$

Thus, accuracy is the fraction of all inputs that are labelled correctly by the classifier. Given that the multi-class case is both harder and more interesting than the binary case, FNC-1 defines total average accuracy as

$$\frac{|\text{Binary Accuracy}| + 3 * |\text{Multi-Class Accuracy}|}{4}$$

7.1.5 Average AUC

The average AUC refers to the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive rate against the False Positive rate at different thresholds. In our case, "unrelated" is again the positive class. AUC thus represents how likely the classifier is to correctly rank a positive class instance higher than a negative class instance.

7.1.6 Confusion Matrices

We further interpret our results using confusion matrices, which represent the counts of items in the classes versus how those items were actually labelled into those classes by our classifier. Along a matrix's top-left to bottom-right diagonal, we can see the counts of correctly labelled items. More interestingly, we can look up indices in the matrix to find whether, for instance, one class was often mistaken for another class. For a sample confusion matrix, refer to Section 7.2 (Results).

7.2 Results

We now provide the results of our binary classifier with just our unique features. We treat "unrelated" as the positive class.

Classifier	Accuracy	Precision	Recall	F1	AUC
SVM	0.475	0.226	0.475	0.306	0.5
Logistic Regression	0.468	0.328	0.468	0.355	0.568
Random Forest	0.635	0.630	0.653	0.631	0.870
KNeighbors	0.382	0.378	0.382	0.376	0.525
Gradient Boosting	0.721	0.7	0.72	0.7	0.94

We also provide the results of our binary classifier with our unique features, combined with the FNC-1 Baseline features.

Classifier	Accuracy	Precision	Recall	F1	AUC
SVM	0.512	0	0	0	0.488
Logistic Regression	0.96	0.952	0.97	0.96	0.99
Random Forest	0.933	0.928	0.937	0.932	0.978
KNeighbors	0.52	0.507	0.514	0.509	0
Gradient Boosting	0.966	0.969	0.961	0.965	0.992

Next, we give the results of our multi-class classifier with our unique features alone.

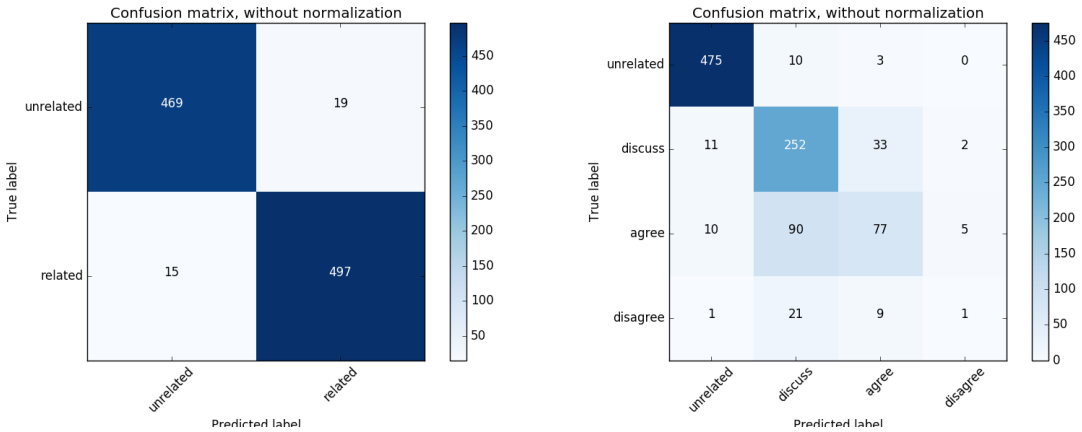
Classifier	Accuracy	Precision	Recall	F1
SVM	0.475	0.226	0.475	0.306
Logistic Regression	0.468	0.328	0.468	0.355
Random Forest	0.635	0.63	0.653	0.631
KNeighbors	0.382	0.378	0.382	0.376
Gradient Boosting	0.721	0.7	0.721	0.695

Finally, we offer the results of our multi-class classifier with our unique features and the FNC-1 Baseline features.

Classifier	Accuracy	Precision	Recall	F1
SVM	0.488	0.238	0.488	0.32
Logistic Regression	0.731	0.612	0.731	0.67
Random Forest	0.755	0.728	0.755	0.739
KNeighbors	0.38	0.363	0.38	0.37
Gradient Boosting	0.805	0.783	0.805	0.787

In general, our unique features performed poorly without the baseline features in the binary case. However, we saw significant improvements over the baseline features alone for the multi-class case, in which we boosted the accuracy score from 0.755 to 0.805. Moreover, when we combined our binary and multi-class results, we achieved an even higher score of 0.845, thus decidedly defeating the established baseline.

Among our unique features, we identified a few as the most important, based on their helpfulness to the Random Forest Classifier's branching procedure. Unsurprisingly, the most valuable features in the binary case were among those that contributed the most to variance: (1) `headlineInBody` (binary feature), (2) cosine similarity between keywords in the body and words in the headline, and (3) number of named entities appearing in the headline that are missing in the body. It makes sense why these these particular features, which all focus on shared content words between the headline and body, would be most important when classifying a pair as "related" or "unrelated." Other significant features to binary classification included: (4) normalized meaningfulness of the body, (5) normalized positivity of the body, (6) normalized negativity of the body, (7) normalized familiarity of the headline, (8) normalized meaningfulness of the headline, (9) normalized objectivity of the headline, and (10) the number of non-comparative/non-superlative adjectives in the body. Some of the most important features for multi-class labelling include items 1 through 9 above, as well as: (1) length of the preprocessed headline, (2) number of numbers in the body, (3) initial trigram of the headline, (4) initial trigram of the body, and (5) final trigram of the body.



In the confusion matrix for the multi-class case, we see that our classifier mislabelled headline-body pairs of type "agree" and "disagree" as "discuss" with relative frequency. This may be due to the fact that our data set (or the part of the data set that we used) includes fewer agree/disagree true labels than discuss examples. However, it may also indicate a high level of noise in our approximation of the positivity and negativity expressed by headlines and bodies. If that is the case, we would want to look at improving our sentiment analysis techniques in the future.

We now analyze some mislabeled examples from our experiments:

1. **Discuss mislabeled as Agree**

   Headline: ISIS militants appear to behead abducted American journalist James Wright Foley in graphic video

   Body: Absolutely awful news. Media are reporting that journalist James Foley, captured in 2012, has been beheaded by ISIS.

   Analysis: The clause that seems to have fooled our classifier is "media are reporting that."

The classifier fails to attribute this statement to the media, instead labeling it as the author's opinion. In reality, the author does not explicitly state her opinion, and this body text simply discusses the headline.

2. **Agree mislabeled as Discuss**

   Headline: Rat problem worsens at One World Trade Center offices of Conde Nast

   Body: They must be the nattiest rats in all of New York City. Journalists at the fashion bible Vogue have been forced to delay their move into swanky new offices on the site of the Twin Towers because of an infestation of rodents. The American magazine has already moved its sales and marketing departments into offices on the 25th and 26th floors of 1 World Trade Center and its writers and editors including Anna Wintour

   Analysis: The true agreement between headline and body occurs in the first sentence of the body. However, the misspelling of "nastiest" as "nattiest" probably fooled our analyzer into missing the expressed negativity in the body, as well as its link to the negative sentiment associated with the term "worsens" in the headline.

3. **Disagree mislabeled as Discuss**

   Headline: KANYE WEST BARRED FROM ALL FUTURE AWARDS SHOWS

   Body: Claim: Kanye West has been banned from all future award shows. FALSE Example: Collected via email, February 2015 Kanye West banned from attending future grammy awards. Is this real? Origins: On 9 February 2015, the entertainment website Adobo Chronicles published an article claiming that rapper Kanye West had been barred from all future award shows: He did it again. Rapper Kanye West reprised his most infamous stunt at Sunday nightś Grammy awards, taking the stage as Beck was accepting the album of the year award. After this latest incident, organizers of the major awards shows like the Grammys, MTV Video Music Awards, Peopleś Choice Awards and the Oscars have unanimously agreed to disinvite and bar West from their respective ceremonies. The television networks that air these awards programs also joined in the Ẃest boycott.Ẃhile it is true that West created a stir at the 2015 Grammy Awards when he briefly interrupted Beck during the musicianś acceptance speech (just as he had done to Taylor Swift in 2009), Kanye has not been banned from all future award shows. The Adobo Chronicles is another "satire" publication that publishes fake news. A disclaimer on the site states that all articles published by them are a mix of facts and fiction: The Adobo Chronicles is your source of up-to-date, unbelievable news. Everything you read on this site is based on fact, except for the lies. Why the title, The Adobo Chronicles, you might ask? Well, adobo is the national dish of our home country. You see, adobo is usually made with pork or chicken, boiled and simmered in a mixture of vinegar, soy sauce and other spices. When writing stories for this blog, we let the news sizzle and simmer in our mind in a mixture of fact and fiction, then we spice it up with figments of our imagination. var CasaleArgs = new Object(); CasaleArgs.version = 2; CasaleArgs.adUnits = "4"; CasaleArgs.casaleID = 159339; ...

   Analysis: The formatting at the beginning, which features separators such as "Claim," "FALSE," "Example," "Origins," likely caused our classifier to miss the body's direct statement of disagreement with the headline. Furthermore, the binary feature `headlineInBody` is true here (at least after our preprocessing stage), which likely confused our classifier into thinking the label must either be "discuss" or "agree." We would thus need to improve our syntactic and discourse analysis in order to determine that the "FALSE" statement in the body refers to the claim in the headline that is repeated in the body.

4. **Agree Mislabeled as Disagree**

   Headline: That Story About a Catholic Priest Dying, Seeing God as a Woman, and Coming Back to Life? Its Almost Definitely Fake

   Body: Its a eyeball-grabbing headline: God is a woman, priest who died for 48 minutes

claims. But its almost certainly not true. A screengrab of the likely untrue story of a priest who supposedly died and saw a female God. The picture of the priest appears to be a stock photo, not a picture of an actual priest. (Image via monitor.co.ug) The story of the Catholic priest who supposedly died briefly and saw a female God has gained massive traction online, sparking debates on Reddit and getting picked up by everyone from the Dallas/Ft. Worth radio station KVIL-FM to viral news sites such as Inquisitr. Yet the story does not appear to spring from any news outlets close to Boston, Massachusetts, where the priest was reportedly working. Instead, the story was first reported by the Daily Monitor, a newspaper based in the African nation of Uganda. Why would a remote African nation be first to an American faith story? Because it pulled the tale from the World News Daily Report, a satirical website. To add insult to injury, the images of Father ONeal that accompany the stories appear to be stock photos (as Reddit user NewdAccount pointed out), not pictures of an actual priest. It seems this story is another bogus tale in need of debunking, like the Sandy Hook conspiracy theory or the fake report of a military coup against the Obama administration over the revelation of an alleged Colin Powell affair. Spokespersons for the archdiocese of Boston were not immediately available for comment Sunday morning. Follow Zach Noble (@thezachnoble) on Twitter

Analysis: This failure may have occurred because we did not perform negation detection, which would have matched "fake" in the headline with "not true" in the body.

5. **Unrelated mislabeled as Discuss**

Headline: Samsung is reportedly making a secret new chip in the Apple Watch

Body: When Apple unveiled its Apple Watch, the company said the starting price would come in at US$349, but insider sources are claiming the stainless steel version will start at $500. Those sources also say the high-end gold Apple Watch Edition will start between $4,000 and $5,000, which is lower than many have been expecting. Apple's mid-range smartwatch will likely start at $500Apple's mid-range smartwatch will likely start at $500 Assuming the sources are correct, that pegs the anodized aluminum Apple Watch Sport as the lower-priced $349 model. The anonymous sources, speaking with the French site iGen (English translation), also said Apple will ship its smartwatch in time for Valentine's Day in February. While a February release would be nice because it would get Apple Watch on consumer's wrists earlier, doesn't fit with Apple Senior Vice President of Retail Angela Ahrendts comments saying the smartwatch will ship in Spring 2015. Apple Watch is a smartwatch device with sensors for tracking fitness activities and heart rate. It links to your iPhone to display alerts and messages, lets you reply to messages, shows turn-by-turn directions, learns your fitness routine and offers suggestions for improvement, and more. Three models in two sizes with multiple band options will be available when Apple Watch ships. Hopefully Apple will release Apple Watch earlier in the year instead of some time in Spring, because even though products like Microsoft's new Band and Fitbit's just announced Surge aren't direct competitors, they are enticing enough to draw away some sales. Waiting until Spring could cost Apple some momentum, too. iGen has been accurate in the past with Apple information, so they may be right on the pricing, but we're going with Angela Ahrendts on the release time frame because she's clearly in a good position to know when her own company's products will ship.

Analysis: The topic of the headline is the "secret new chip," but this is never mentioned in the body of the article. However, our classifier is not fine-grained enough in its understanding of sentence structure and semantics to determine that, although both the headline and body relate to Apple and watches, their foci are completely unrelated.

8 Conclusions

8.1 Summary

We found that a Gradient Boosting Classifier trained on a combination of the FNC-1 baseline features and our original features outperformed the standard set by the Fake New Challenge

creators. We learned that the new features we introduced were most helpful in the multi-class case, likely because many of them (SentiWordNet scores, frequency of adjectives, personal pronoun usage) were predictive of opinion-making behavior (i.e., "agree"/"disagree" relationships between headlines and bodies). After analyzing our classifier's errors, we learned that achieving the highest levels of accuracy will require us to look at not only local signals in our texts, but also document-level themes that can only be gleaned through proper syntactic, semantic, and discourse analysis, as opposed to faster or more efficient (but ultimately noisy) heuristics. Altogether, our results suggest that relationships between texts lie not only in the content of each piece (as determined by processes like automatic summarization) but also in their sentiments and styles (Paivio Meaningfulness, Familiarity, length, use of social media tokens, etc.).

8.2 Future Work

If we had more time with this project, we would focus on improving our sentiment analysis techniques. Also, because our Gensim summarizer is based on the TextRank algorithm, it could be beneficial to experiment with different window sizes when determining centrality. We would also look into performing discourse analysis, as well as using FrameNet more creatively than we have thus far. Moreover, we might look into a feature that matches exact number values, rather than the number of cardinals. Finally, it would be helpful to speed up feature extraction so that we could test our implementation efficiently and with a larger corpus of headline-body pairs.

8.3 Extended Applications

We originally hoped to apply these same techniques to news in a foreign language. More specifically, given the similar political climates of the United States and France, we attempted to transform our code to work with French headlines and body texts.

As it turned out, however, this approach was more daunting than expected, largely due to the syntactic differences between the languages. For example, negations are more complex and varied in French than in English. They tend to entail two words surrounding the verb that is negated, so they cannot be identified with a simple search for a single word. Similarly, French does not have simple modals in the same way that English does.

Stopwords also present related challenges. For example, the translated equivalent of many English stopwords span multiple words in French. French also grapples with verb conjugation and gender agreement, meaning that a basic list of French stopwords could be considerably more complex than one in English.

Along the same vein, the myriad combinations of conjugations, subject numbers, and genders may render the tactics that rely on repetition relatively futile—at least for corpora of a reasonable size. That is, because verbs have over 20 possible tenses, each of which features up to six different conjugations, software is far less likely to see many duplicate phrases in a reasonable amount of data. (If we hoped to this investigate further, we would perhaps begin by searching for software that reduces a French verb to its stem.)

Altogether, these and other challenges—as well as the dearth of foreign languages that we encountered in the aforementioned research—suggest that current innovations in stance detection are not easily transferred from one language to another. While slightly disappointing, this does point to exciting areas for future exploration, should interest in automatic fake news detection spread to non-English speaking countries. Moreover, it indicates just how difficult it will be to counteract fake news on a truly global basis.

9 Honor Code

This paper represents our own work in accordance with University regulations.

Elise Georis and Isabelle Ingato

References

[1] I. Augenstein *et al*, "Stance Detection with Bidirectional Conditional Encoding," *CoRR*, vol. abs/1606.05464, 2016. [Online]. Available: http://arxiv.org/abs/1606.05464

[2] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly, 2009. [Online]. Available: http://victoria.lviv.ua/html/fl5/NaturalLanguageProcessingWithPython.pdf

[3] J. Constine, "Facebook chose to fight fake news with AI, not just user reports," 2016. [Online]. Available: https://techcrunch.com/2016/11/14/facebook-fake-news/

[4] Fake News Challenge, "Fake News Challenge Stage 1 (FNC-I): Stance Detection," n.d. [Online]. Available: http://www.fakenewschallenge.org/

[5] FakeNewsChallenge, "fnc-1-baseline," 2017, Github repository. [Online]. Available: https://github.com/FakeNewsChallenge/fnc-1-baseline

[6] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling," *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics*, pp. 363–370, 2005, Computer Science Department, Stanford University. [Online]. Available: https://nlp.stanford.edu/ manning/papers/gibbscrf3.pdf

[7] P. Krejzl, B. Hourova, and J. Steinberger, "Stance detection in online discussions," *CoRR*, vol. abs/1701.00504, 2017. [Online]. Available: http://arxiv.org/abs/1701.00504

[8] "MRC Psycholinguistic Database," 2017, department of Psychology, University of West Alabama. [Online]. Available: http://websites.psychology.uwa.edu.au/school/MRCDatabase/uwa_mrc.htm

[9] NumPy developers, "NumPy." [Online]. Available: htttp://www.numpy.org/

[10] B. G. Patra, D. Das, and S. Bandyopadhyay, "JU_NLP at SemEval-2016 Task 6: Detecting Stance in Tweets using Support Vector Machines," 2016, department of Computer Science and Engineering, Jadavpur University. [Online]. Available: https://www.aclweb.org/anthology/S/S16/S16-1071.pdf

[11] Princeton University, "What is WordNet?" 2015. [Online]. Available: https://wordnet.princeton.edu/

[12] R. Rehurek, "summarization.summarizer – TextRank Summariser," 2017. [Online]. Available: https:// radimrehurek.com/gensim/summarization/summariser.html

[13] J. Ruppenhofer, *et al.*, "FrameNet II: Extended Theory and Practice," 2010. [Online]. Available: https://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf https://framenet.icsi.berkeley.edu/fndrupal/WhatIsFrameNet

[14] scikit-learn developers, "scikit-learn: Machine Learning in Python," 2016. [Online]. Available: http://scikit-http://scikit-learn.org/stable/

[15] "sklearn.ensemble.GradientBoostingClassifier," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

[16] "sklearn.linear_model.LogisticRegressionCV," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html

[17] "sklearn.model_selection.StratifiedKFold," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

[18] "sklearn.neighbors.KNeighborsClassifier," 2016. [Online]. Available: http://scikit-learn.org/ stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[19] "sklearn.svm.SVC," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[20] "sklearn.linear_model.LogisticRegressionCV," 2016. [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[21] "SVM-Anova: SVM with univariate feature selection," 2016. [Online]. Available: http:// scikit-learn.org/stable/auto_examples/svm/plot_svm_anova.html

[22] "SentiWordNet," 2010. [Online]. Available: http://sentiwordnet.isti.cnr.it/

[23] P. Sobhani, S. M. Mohammad, and S. Kiritchenko, "Detecting Stance in Tweets and Analyzing its Interaction with Sentiment," *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pp. 159–169, 2016. [Online]. Available: https://aclweb.org/anthology/S/S16/S16-2021.pdf

[24] spaCy (Explosion AI), "Industrial-Strength Natural Language Processing in Python," 2017. [Online]. Available: https://spacy.io/

[25] S. Tavernise, "As Fake News Spreads Lies, More Readers Shrug at the Truth," 2016. [Online]. Available: https://www.nytimes.com/2016/12/06/us/fake-news-partisan-republican-democrat.html

[26] "What is FrameNEt?," n.d. [Online]. Available: https://framenet.icsi.berkeley.edu/fndrupal/WhatIsFrameNet

[27] M. Wojatzki and T. Zesch, "ltl.uni-due: Stance Detection in Social Media Using Stacked Classif," 2016, Language Technology Lab, University of Duisburg-Essen. [Online]. Available: http://www.ltl.uni-due.de/wp-content/uploads/SemEval2016.pdf