Is This Thesis Fake News?
Linguistic Methods for Categorizing News

Edie Reimink
Advisor: Claire Bowern

Submitted to the faculty of the Department of Linguistics in partial fulfillment of the requirements for the degree of Bachelor of Arts

Yale University
May 2018

Abstract

Just as more and more people turn to online articles and websites to keep up-to-date on current events and developing news stories, the number of websites and news articles of questionable credibility has also grown in recent years. This paper seeks to define natural categories that news can be divided into in order to develop a more nuanced understanding of 'fake' and 'real' news, using computational clustering methods to identify these groups. First, a corpus is compiled which takes news articles from mainstream, satire, and 'fake' web pages, with frequency counts conducted based on parts of speech and LIWC psycholinguistic features. Then the data is analyzed using self-organized mapping to cluster similar articles together. These clusters are then analyzed to see whether or not they correspond with any existing teleological groupings.

Acknowledgments

I owe my deepest gratitude first and foremost to my advisor, Claire Bowern, for her patience and guidance. I would also like to extend my thanks to Bob Frank, Raffaella Zanuttini, and the other seniors for their feedback and advice, and to Larry Horn, for agreeing to be my second reader. Last, but most certainly not least, I would like to thank Ali Yawar, Josh Phillips, Walter Bircher, Neelima Sharma, and Nihav Dhawale for their endless moral support, for providing me with copious amounts of coffee, and for listening to me ramble on and on about this thesis.

Contents

1 Introduction

In recent years, as media has become more and more pervasive and news websites and blogs have become a dime a dozen, a number of web pages have come into being that claim to report on current events, but whose credibility has been called into question. Debate over such websites has become so prevalent that the topic of 'fake news' is itself an integral part of the news cycle. What constitutes 'fake news,' however, has only become less clear as the topic has become more common, with mainstream news sources such as CNN and FoxNews coming under the firing line. In response to the increased presence of fake news there has been an exploration of computational methods for detecting such news. Some models focus on the ways in which fake news articles are shared via social media. Other proposed models focus on using corpus analysis to pick out fake news articles.

What seems to be missing from the discussion of fake news, or at least what is given lower priority in such a discussion, is how to define fake news. The discussion of 'fake' and 'real' news implies a binary, in which there seems to be an assumption that the label of fake news can only apply to that which is factually inaccurate, and the label of real news can only apply to that which is wholly factually correct. However, this leaves out articles which might convey factually accurate information, but present such information out of context so as to encourage the misinterpretation of the information (often called propaganda). It also neglects satire, which is intended to parody current events for humorous effect, but can be misinterpreted or mistaken for real news.

Some computational models of deception detection attempt to distinguish simply between the binary of fake and real news. Other models only attempt to distinguish between satire and real news. Some other models attempt to distinguish between fake news, satire, propaganda, and real news. The lack of standardization of these categories across models calls into question just how sound these categories are and begs the question of whether some categories are more robust or significant than others.

In section 2 of this paper, I'll discuss in more depth some of the previous computational work that has been done on deception detection as well as the categories and definitions of news that have been utilized in these analyses. In section 3, I will put forth my hypothesis and introduce the tools I plan to use for my own analysis. In section 4, I present my methods and results. In section 5 I discuss these results, and section 6 concludes the paper.

2 The Landscape of Deception Detection

In an attempt to create algorithms which can automatically detect fake news, various attempts have been made to train models based on corpus analysis. These models are built on corpora made up of articles from various news sources which are marked for various features. The news sources and the features used to mark the corpora are often different across models, and range between features which relate to the style of reporting and features which relate to the content of the reporting. The computational methods and categories of news used for the analysis also tend to differ across models.

2.1 Predictive Models and Features

Yang et al. Yang et al. 2017 attempts to determine features that are effective for identifying satirical news, and proposes a model to detect satirical news. The model proposed in Yang et al. 2017 is a 4-level hierarchical neural network model with linguistic features embedded. The four different levels which make up this hierarchy are based upon the structure of the document itself: the character level, the word level, the paragraph level, and the document in its entirety. Yang et al. proposes this hierarchical model based on the theory that paragraphs within documents are significantly different from one another; specifically with regards to satire, this is intended to address the fact that satire articles often features paragraphs which are purely functional and meant to set up the satire, thus display no obvious features of satire themselves. The model takes four different types of linguistic features into account: psycholinguistic, writing stylistic, readability, and structural. The corpus used for analysis was created by collecting articles from 14 websites that self-declare that what they publish is satire, omitting headline, creation time, and author information.

In order to determine the optimal method of satirical news detection, various models of satire detection were used to asses the corpora, and the results of each model were compared. One category of models simply used method learning and different combination of word n-grams and linguistic features. Another category of models used the 4-level hierarchical neural network with linguistic features embedded at different levels. The model with highest accuracy, recall, and F1 statistic was the the 4-level hierarchical neural network which included linguistic features at both the paragraph level and document level.

Once the 4-level hierarchical neural network model with linguistic features embedded at both paragraph and document level was determined to be the most useful model, individual linguistic features were examined within each category by calculating the weight for each feature at the paragraph level and the document level for both satirical news and true news. Readability was determined to be the most important category of linguistic features at the document level, while psycholinguistic features, writing stylistic features, and structural features were determined to be more important at the paragraph level.

While Yang et. al 2017 does not address degree estimation within their model, they suggest that satire might be defined by different degrees of sarcasm, irony, and humor, and thus suggest the next step for their project: using similar methods to determine what role those three categories play in the detection of satirical news.

Rashkin et al. Rashkin et al. 2017 attempts to provide a more nuanced model for predicting news type by introducing more categories of news. They argue that not all types of fake news articles have the same intent: some are meant to be humorous and are not meant to be taken as truth, while others are meant to persuade readers of the truthfulness of their contents. They use the intent of the author as well as the actual veracity of the article to define four categories of news: satire, hoax, propaganda, and trusted news, as seen in Figure 1 below. They define trusted news as that in which the author intends to tell the truth, and uses trustworthy information in their article; propaganda is that which intends to deceive but includes some trustworthy information; hoax is that which intends to deceive, and uses false information; satire is that which uses false information, but has no intent to deceive readers. Based on these definitions, Rashkin et al. attempt to find stylistic predictors of each category.

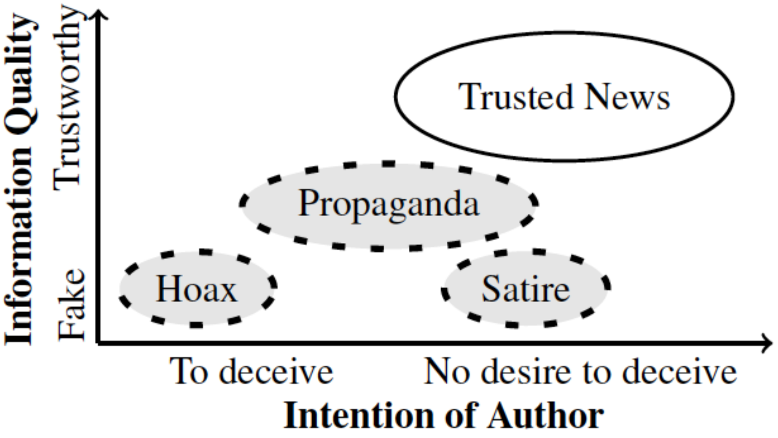


Figure 1: Categories of news based on author's intent and the veracity of the content (Rashkin et al. 2017)

A stylistic lexical analysis of each category was performed on a corpus which collected news articles from a variety of different sources. All trusted news articles were collected from the English Gigaword corpus. The Gigaword corpus, compiled by the Linguistic Data Consortium, contains articles from four different English-language newspapers: Agence France Press English Service, Associated Press Worldstream English Service, The New York Times Newswire Service, and The Xinhua News Agency English Service. Corpora were created for satire by sampling articles from The Onion, The Borowitz Report, and Clickhole, for hoax by sampling from American News and DC Gazette, and for propaganda by sampling from The Natural News and Activist Report. Analysis centered on the frequency of different lexical categories as they appeared within each genre of news. These categories consisted of Swear, 2nd person, Modal Adverb, Action Adverb, 1st person singular, Manner Adverb, Sexual, See, Negation, Strong Subjective, Hedge, Superlatives, Weak Subjective, Number, Hear, Money, Assertive, and Comparatives. Ratios were calculated of how frequently words in each category appeared in unreliable news articles as compared with how frequently they appeared in trustworthy news articles. It was also noted which type of unreliable news article each lexicon marker category appeared in the most.

Because ratios were only calculated as a comparison of trusted news sources and unreliable news sources, and not as a comparison between different categories of unreliable news, the lexicon markers can only be used as predictors for distinguishing between trustworthy and unreliable news, and not for distinguishing between satire, hoax, and propaganda. Also, while it is noted which type of unreliable news uses words from a category with the highest frequency, because no ratio is calculated between categories of unreliable news, there is no way to determine whether a category of lexicon marker is a significant predictor of satire, hoax, or propaganda.

While Rashkin et al. 2017 demonstrates that stylistic lexical markers can be used as significant predictors of reliable versus unreliable news, they do little to address the differences between categories of unreliable news. To achieve a better understanding of "fake news," it would be helpful to conduct a frequency analysis of these lexical markers as they appear between categories.

2.2 Existing Categories of News

Little work has been done that focuses specifically on dividing news into different categories. As mentioned earlier, Yang et al. 2017 focuses on singling out satire from 'trusted' news, and build their corpus from websites which explicitly claim to deliver satire articles. Burfoot & Baldwin 2009 also focuses on two categories, satire and 'true' news.

Rubin et al. 2015 lays the groundwork for the categories used by Rashkin et al. They propose three categories of deceptive news: serious fabrications, which broadly corresponds to the Rashkin et al. category of propaganda; large-scale hoaxes, which broadly corresponds to the Rashkin et al. category of hoax; and humorous fakes, which broadly corresponds to the Rashkin et al. category of satire. These categories again depend on both author intent and the factual information contained within the article, as represented by Figure 1 above. While they examine three types of deceptive news, they ultimately lump these types into one category, and run the main analysis on just two categories of news: trusted and fake.

Potthast et al. focuses singularly on the impact that politics has on the media, and gathers news based on whether it is from a 'mainstream,' 'left-wing,' or 'right-wing' news source. Once they fact-check the articles, however, they again build their model based on the dichotomy of fake and true news.

3 Eyeing Natural Categories

In spite of the fact that those who have built deception detection models have been willing to acknowledge the nuance that exits between different categories of news, much of the work that has been done on deception detection has still ultimately been based on a dichotomy between news which is categorized as being either clearly "fake" or "true." I hypothesize that there are more fine-grained categories which exist for classifying news articles to the extent that a self-organizing map would be able to organize or cluster data into robust natural categories based on stylistic features, which would correlate to teleological categories (such as hoax, propaganda, satire, etc).

3.1 Unsupervised Learning: Self-Organized Maps

Many of the attempts that have been made at deception detection have used various supervised learning methods to develop their models. In these cases, the model is trained on one part of the corpus, and tested on the rest of the corpus. In doing so, the resulting model is constrained by the categories imposed upon it beforehand.

Finding natural categories depends on finding natural clusters among the articles, based on feature values. A number of different clustering methods exist which could be applicable in this case. Given the high dimensionality of the data and the exploratory nature of the analysis, I placed more consideration on dimension-reducing methods of clustering such as principal component analysis, multi-dimensional scaling, and self-organizing maps. Organizing the data in a two-dimensional space allows for the consideration of resulting clusters on a scale - in this case, theoretically from more fake to more true. Multi-dimensional scaling attempts to plot an estimate of the original distance matrix of the high dimensionality data in a continuous space, which places an emphasis on the dissimilarities in data. Self-organizing maps instead map data objects onto a grid of units, which places an emphasis on similarities in the data. Given that I am interested in forming clusters based on which articles are more similar to one another, I chose self-organizing maps as my tool of analysis.

Self-organizing maps (SOMs from here on out) are a useful tool for visualizing complex data, which can take data with multiple dimensions and compress the values to produce a 2D representation of the data with data objects grouped together based on value similarity. Rather than mapping objects together in a continuous space, the SOM is organized around a grid of units, often referred to as nodes. The value of these nodes is randomized at the beginning of the analysis. Then the data is presented to the grid one object at a time; the object is compared against all of the nodes in the grid, and then is assigned to the "winning" node (that node which is most similar to the data object). The winning node, and its neighbor nodes, are then weighted to become more similar to the value of the object data assigned to the node. An example of this process can be seen in Figure 2 below. In this case, blue, green, and red objects are presented to the grid; objects which are blue are matched to the 'blue' nodes, and in turn make those nodes more 'blue.'

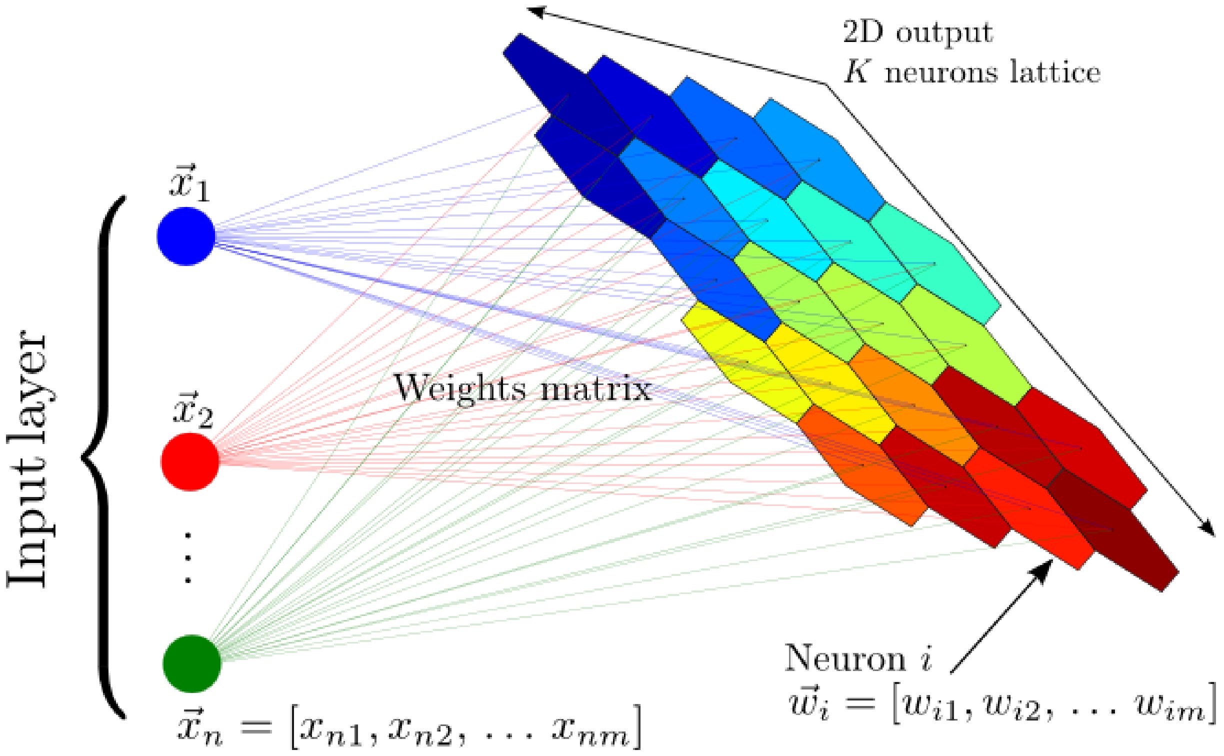


Figure 2: How data are mapped to the nodes (Carrasco Kind 2015)

3.2 Linguistic Features

One of the most important aspects of identifying potential natural categories is to determine what features of language will be used as metrics. As Yang et al. 2017, Potthast et al. 2017, and Rashkin et al. 2017 demonstrate, parts of speech have previously been significant predictors of type of news. Yang et. al 2017 and Rashkin et al. 2017 also demonstrate the significance specific features of the Linguistic Inquiry and Word Count (LIWC) dictionary. While other features, such as readability, have also been shown to be significant predictors of type of news, choosing parts of speech and the LIWC software offer the opportunity to examine both style and content of the corpus, and are easy to apply to my corpus using pre-made software. In this instance, I am able to mark my corpus for parts of speech with the Penn English Treebank tagset through the software *TreeTagger* (Schmid 1994) and *R*, while I am able to mark my corpus for the Linguistic Inquiry and Word Count dictionary using the LIWC software.

The Penn English Treebank dictionary consists of 36 different categories corresponding to parts-of-speech (see Table 1 below). Unlike the Universal tagset, which is intended to work across languages and features 12 basic parts of speech (adjectives, adpositions, adverbs, conjunctions, determiners, nouns, cardinal numbers, particles, pronouns, punctuation, verbs, other), the Penn tagset is tailored to English and offers a slightly more complex dictionary of parts of speech. The Penn tagset offers some distinction within part-of-speech categories, for instance distinguishing between comparative, superlative, and all other adjectives. While an even more nuanced tagset like the Brown tagset, which contains 87 different part-of-speech tags, might offer an even more nuanced analysis of the corpus, I could not find a program to tag my corpus with the Brown tagset, and thus settled on the Penn English Treebank tagset.

Table 1: Penn English Treebank

POS Tag	**Description**	**Example**
CC	Coordinating conjunction	and
CD	Cardinal number	1, one
DT	Determiner	the
EX	Existential there	there is

POS Tag	**Description**	**Example**
FW	Foreign word	d'hoevre
IN	Preposition or subordinating conjunction	in, of
JJ	Adjective	green
JJR	Adjective, Comparative	greener
JJS	Adjective, Superlative	greenest
LS	List item marker	1)
MD	Modal	could
NN	Noun, singular or mass	table
NNS	Noun, plural	tables
NP	Proper noun, singular	John
NPS	Proper noun, plural	Vikings
PDT	Predeterminer	all
POS	Possessive ending	friend's
PP	Personal pronoun	I
PP.	Possessive pronoun	my
RB	Adverb	naturally
RBR	Adverb, comparative	better
RBS	Adverb, superlative	best
RP	Particle	give up
SYM	Symbol	/
TO	infinitive to	to go
UH	Interjection	oh
VB	Verb, base form	be
VBD	Verb, past tense	was
VBG	Verb, gerund or present participle	being

POS Tag	**Description**	**Example**
VBN	Verb, past participle	been
VBP	Verb, non-3rd person singular present	am
VBZ	Verb, 3rd person singular present	is
WDT	Wh-determiner	which
WP	Wh-pronoun	who
WP.	Possessive wh-pronoun	whose
WRB	Wh-adverb	where

The Linguistic Inquiry and Word Count dictionary incorporates results from a number of different psychological studies to create 90 categories for analyzing text. Categories which receive special attention in a natural category analysis are: pronouns (first-person singular, first-person plural, second person, third-person singular, third-person plural, and indefinite); verb tense (past, present, and future); negative emotion; differentiation; and motion.

4 Methods

4.1 Compiling a Corpus

The corpus features articles from three different categories: satire, questionable, and mainstream news sources. I used two different methods in order to generate urls of the websites: the first was using the program *Lynx* (Dickey 2017) to collect website urls; the second was using the *R* program *RCrawler* (Kahil 2017). The method used depended upon the formatting of the urls. The code used to run all of the *R* programs mentioned from here on out appears in the appendix to this paper.

Once the urls were generated, the program BootCaT (Baroni & Barondini 2004) was used to download the body of text from each url. I then randomly selected 100 articles from the corpus compiled by BootCat and manually cleaned them; errors (such as the inclusion of advertisements or article title) were removed. Then I used the program *TreeTagger* through it's *R* wrapper, *koRpus*, to tag each word in each article with it's corresponding part of speech. The output of the *TreeTagger* program was a data frame with columns for the document name, the word being tagged, and the tag itself, as seen in Table 2:

Table 2: Sample NYT TreeTagger Output

doc id	**token**	**tag**
1.txt	on	IN
1.txt	monday	NN
1.txt	democratic	JJ
1.txt	leaders	NNS
1.txt	gathered	VBD
...	...	...

The proportional frequency of each part-of-speech tag was then calculated for each article by dividing the frequency of the part of speech by the total number of words in the respective article. These frequencies were compiled into a data frame organized with parts of speech as columns and article number as rows. An example of this organization can be seen in Table 3 below.

Table 3: Sample Onion POS Frequency Table

doc id	**CC**	**CD**	**DT**	**...**
on.1	1.323	1.022	2.101	...
on.2	0.000	1.180	1.879	...
on.3	1.204	1.204	1.857	...

doc id	**CC**	**CD**	**DT**	**...**
on.4	1.383	0.906	1.383	...
on.5	0.995	1.472	1.949	...
...	...	...	...	...

Once the part of speech data frame was finished, I created another data frame based on the LIWC dictionary. To create this second data frame, each article was analyzed using the LIWC software. The articles were tagged by the software and a table was generated as the output. The rows of the table were individual articles, while the columns corresponded to each of the 90 LIWC categories. The values of each row were the proportional frequency of words belonging to that category within each article, as can be seen in Table 4 below. From this large table, a smaller table was created containing only the categories mentioned earlier: pronouns, verb tense, negative emotion, differentiation, and motion.

Table 4: Sample Onion LIWC Frequency Table

doc id	**Word Count**	**Analytic**	**Clout**	**...**
on.1	159	90.04	75.56	...
on.2	80	93.98	59.88	...
on.3	206	80.91	61.47	...
on.4	201	87.38	80.10	...
...	...	...	...	...

4.2 Implementing a Self-Organized Map

After the data frames were created they could be analyzed using the self-organizing mapping method described earlier. The SOM package *kohonen* (Wehrens & Buydens 2007) was used in *R* to run the analysis. The function *topology* was used to optimize the number of cells in the map for each data frame, such that upon completing the analysis there would not be too many empty cells. The map was then initialized using this optimized number of cells by establishing the dimensions of the x-axis and y-axis and the shape of the grid (either hexagonal or rectangular). In this case, each map was initialized with hexagonal topology, as this allows for every neighbor of a neuron to be equidistant from that neuron. Once the neurons were initialized, the data frame was converted to a matrix and the model was created using the function *som*. The resulting model is a list object in the *R* environment, which contains information about the members occupying each neuron of the map, as well as the distances between neurons. This list can then be visualized and analyzed in a few different ways in order to examine the relationships between neurons and their members.

One way to determine optimal clustering of the map is to perform a hierarchical clustering of the data, based on the distances between neurons. The result of this hierarchical clustering for the parts of speech data frame can be seen in the dendrogram in Figure 3 below.

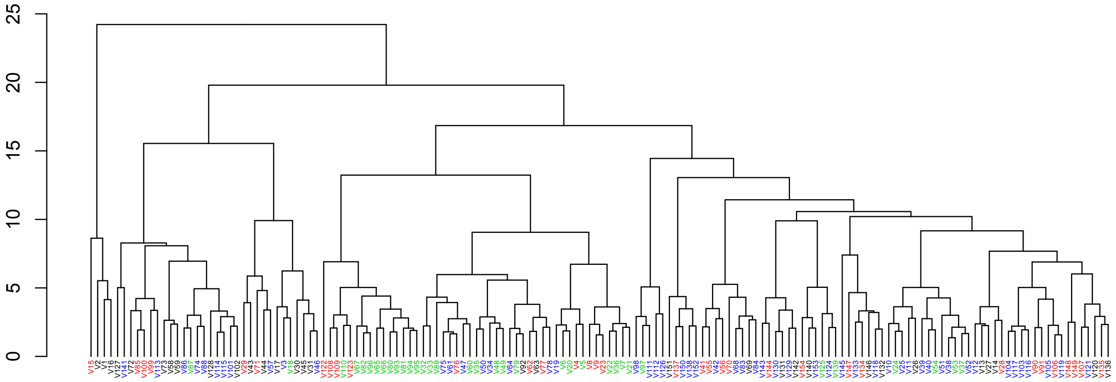


Figure 3: POS Dendrogram

Part of the output produced by the SOM model is a unified distance matrix (as seen in Figure 4 below), which represents the distances between neurons of the map. This is useful for determining clusters within a self-organizing map: places on the map which exhibit greater distances between neurons can be useful indicators of clusters. The unified distance matrix presented below has three clusters from the hierarchical clustering superimposed over the matrix. This juxtaposition of the two clustering methods demonstrates that the members of the first and smallest cluster in the bottom left corner are similar to one another, but very distinct from the rest of the corpus. While the second and third clusters are shown to be distinct from one another in the hierarchical clustering, they are shown to be not as strongly distinct from one another as they are from the first cluster.

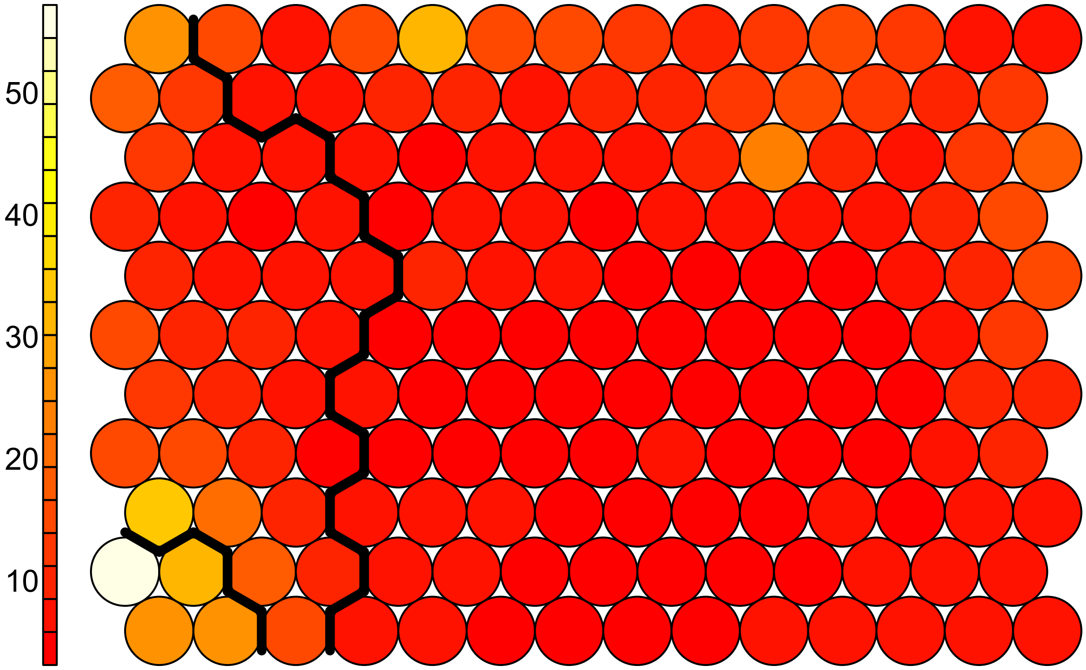


Figure 4: Unified Distance Matrix with cluster boundaries superimposed over the matrix.

Once I decided to use three clusters for my analysis, I was then able to use the plotting tools within the *kohonen* package to create a visual representations of the neurons and the groupings they belonged to. Figure 5 below represents the clusters rendered by the SOM model of the articles tagged for part of speech.

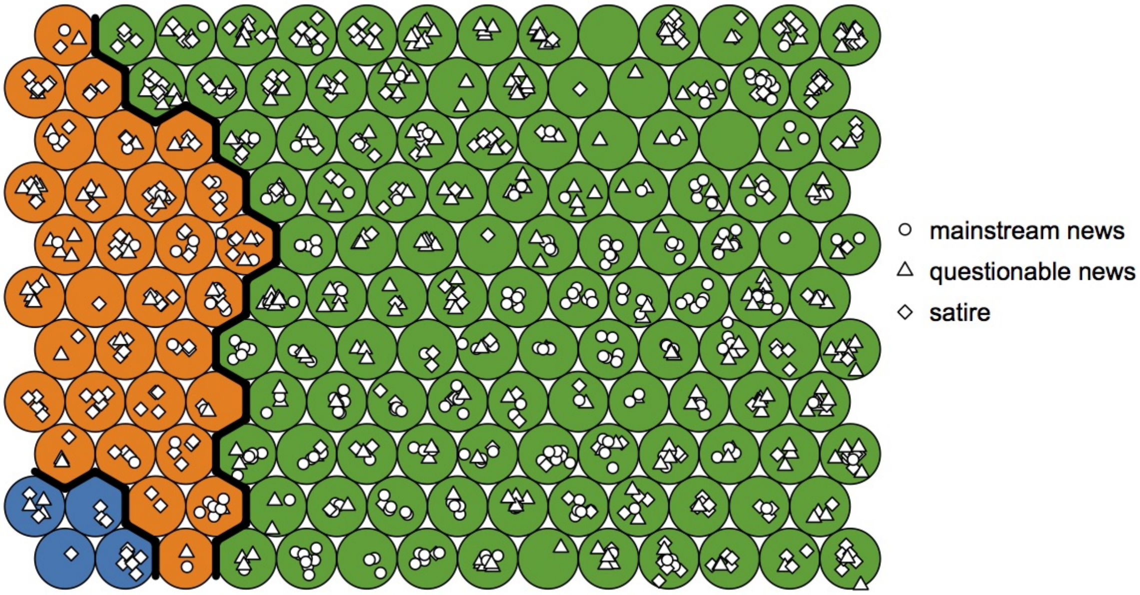


Figure 5: Plot of the articles clustered based on parts of speech. Nodes are colored according to the cluster they belong to.

From a quick examination of this map, it might seem as if some significant categories have emerged. The first cluster, colored blue in the map above, contains 17 articles; this cluster is made up of predominantly satire articles, but also includes a few articles from questionable news sources. The second cluster, colored orange, contains 149 articles; this cluster is again predominantly made up of satire articles, but includes a fair amount of articles from both questionable news sources and mainstream news sources. The third cluster, colored green, contains the remaining 734 articles and includes articles from all three types of news sources.

For a more thorough examination of the members of each cluster, I examined the the subsections of the dendrogram that belonged to each cluster. Each label of the dendrogram represents a single neuron in the self-organizing map, and was color coded based on the type of article which is most representative of the neuron.

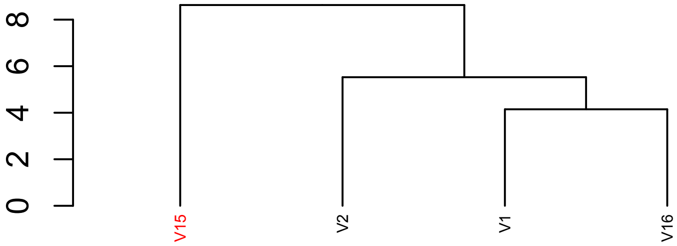


Figure 6: Dendrogram of the first cluster. Each label corresponds to a cell of the self-organizing map. Black labels represent cells whose members are majority satire articles. Red labels represent cells whose members are majority questionable articles.

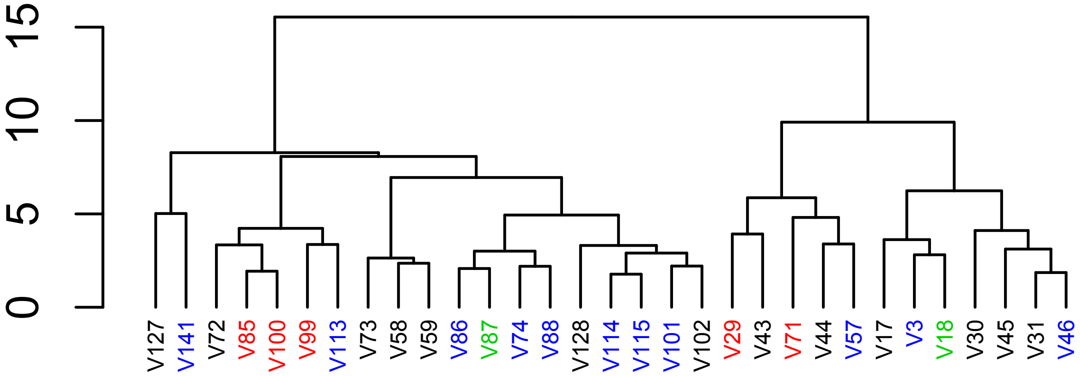
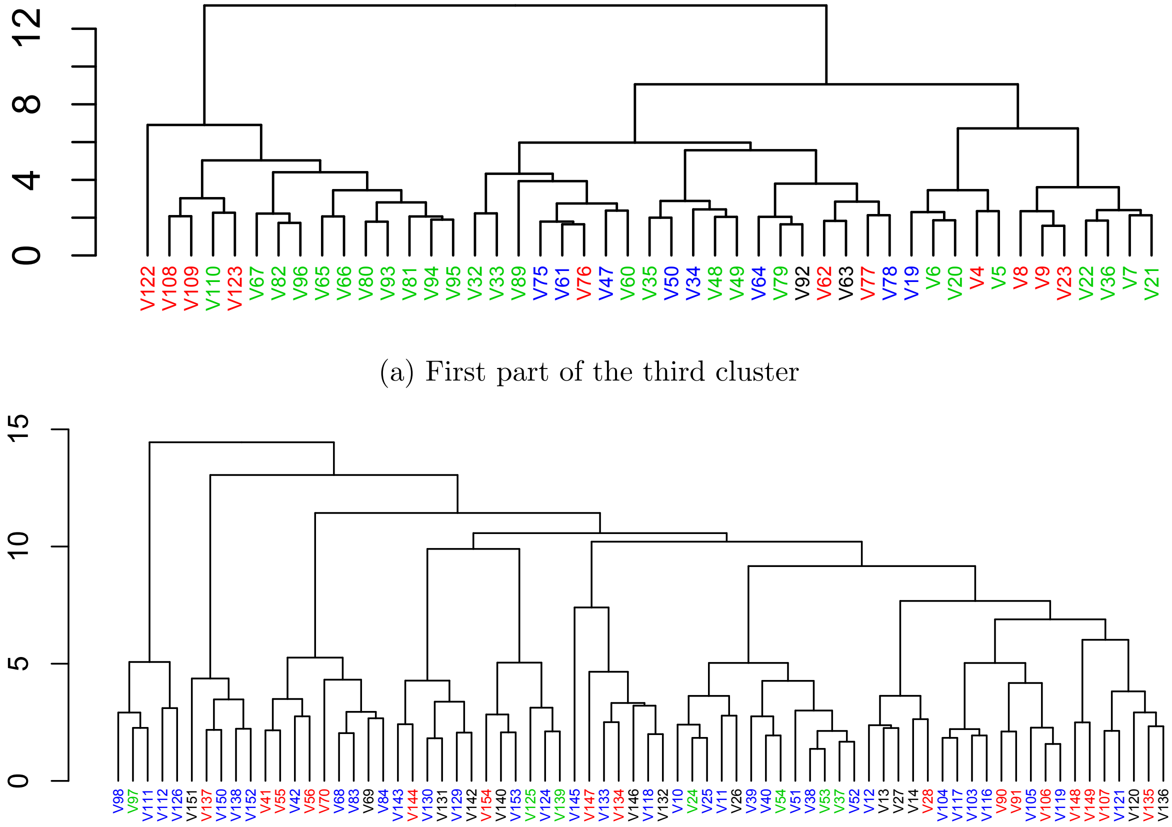


Figure 7: Dendrogram of the second cluster. Black labels represent cells whose members are majority satire articles. Red labels represent cells whose members are majority questionable articles. Green labels represent cells whose members are majority mainstream articles. Blue labels represent cells whose members do not come from one specific type of news.

Figures 6 and 7 above show the dendrogram subsections of the first two clusters. Again it is clear that the first cluster is made up predominantly of satire articles. From this visualization, it is much easier to see that while the second cluster does have a large amount of neurons which contain satire articles, there is also a large number of neurons which do not clearly represent one type of new article over another.

Because the third cluster, represented in Figure 8 below, was so large, I split it into two subsections. In part (a) below, it appears as if there is a very small cluster of articles from mainstream news sources within the larger cluster (V67 through V95). However, the rest of subsection (a) shows no other groupings, and subsection (b) also shows no predominate news type. In cluster three, it is again clear that there are many neurons whose member articles do not clearly belong to one type of news over another.



(a) First part of the third cluster

(b) Second part of the third cluster

Figure 8: Dendrogram of third cluster colored by type of member articles. Black labels represent cells whose members are majority satire articles. Red labels represent cells whose members are majority questionable articles. Green labels represent cells whose members are majority mainstream articles. Blue labels represent cells whose members do not come from one specific type of news.

To test the efficacy of these clustering methods and determine whether or not the clusters represented above were significant, I first performed a test of robusticity on the clusters. To do this, I randomly selected ten percent of the articles from each cluster to remove from the data frame. I removed 2 articles from the first cluster, 15 articles from the second cluster, and 74 articles from the third cluster. I then ran the self-organizing map again using this reduced data frame. The resulting self-organizing map produced similar-sized clusters, but the clusters were not robust from the first run to the second. Were the clusters to be robust, I would expect that a high percentage of the articles which appeared in a cluster in the first run would then also appear in the corresponding cluster in the second run. However, this was not the case: for the first cluster, none of the articles were consistent between the first run and the second run; for the second cluster, only 12.2% of articles were consistent between both runs; for the third cluster, only 31.9% of articles were consistent between runs.

Another self-organizing map was modeled using the corpus as it was tagged for the Linguistic Inquiry and Word Count features, as represented in Figure 9 below.

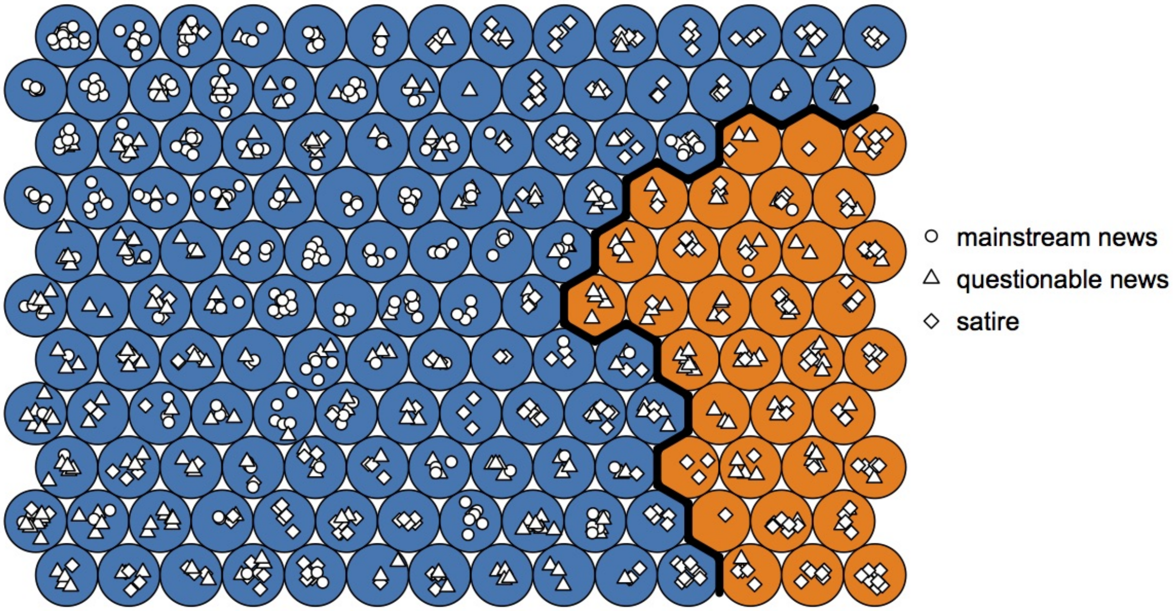


Figure 9: Articles clustered based on LIWC features

Unlike with the parts of speech model, in the LIWC model, there did not seem to be any potentially significant clusters within the map. I again inspected dendrograms of the clusters for a closer examination, as seen in Figures 10 and 11 below.

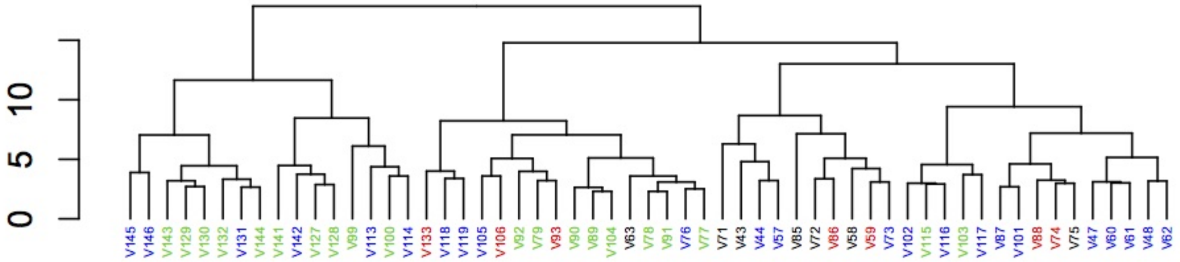


Figure 10: First cluster based on LIWC features

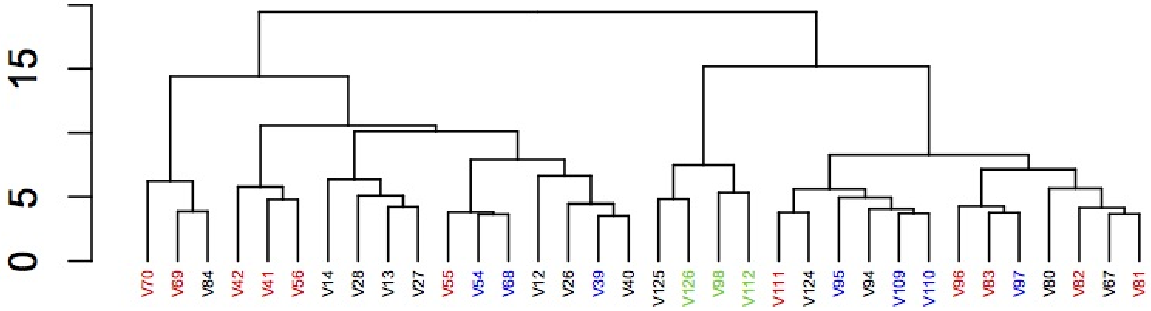


Figure 11: Second cluster based on LIWC features

While the first cluster does seem to contain more neurons which are occupied by articles from mainstream news sources, and the second cluster seems to contain more neurons which are occupied by articles from questionable and satirical news sources, both clusters contain many neurons representing each type of news. Moreover, both clusters also contain many neurons whose members come from all three types of news.

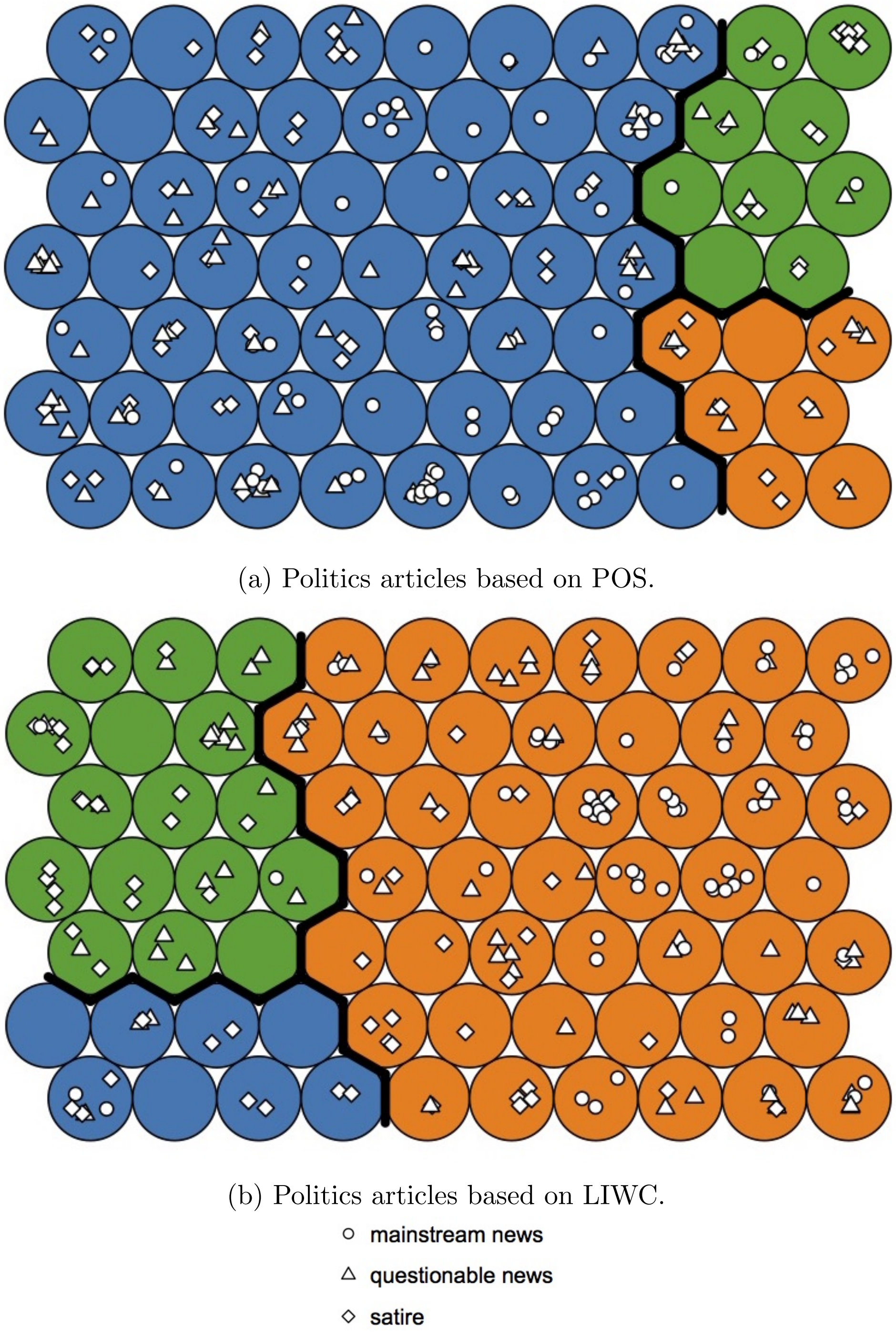
A robusticity test was also performed on these clusters by removing 10 percent of the data and modeling a new self-organizing map based on the reduced data frame. In this case the clusters again proved to lack robustness, with the first cluster only containing 37.98% of the same articles between the first model and the second, and with the second cluster containing only 1.06% of the same articles between the two models.

Overall, both the model based on parts of speech and the model based on the LIWC features failed to produce any significant, robust categories of news type.

4.3 Re-examining the Categories

Due to an inability to find any natural categories, I was also unable to run any analysis which would determine the most important predictive features of these categories. Instead, I decided to expend more effort refining the data sets. While compiling the corpora, I noticed that there were a few common topics that frequently recurred, and so I decided to re-sort the corpora into groups based on topic. One such topic was politics; another was pop-culture. Sorting the groups based on such topics was done in the hopes that eliminating variation among article topics might also eliminate any noise that was preventing patterns in the data from emerging. I went through all 900 articles manually, and categorized them as 'politics,' 'pop-culture,' 'international,' 'business,' 'science,' or 'other' based on briefly skimming the first few sentences of the article. Dividing the articles into groups based on topic required a fair amount of executive decision on my part, as many of the articles could have been labeled as belonging to two or more of the above categories. In these cases, I placed the article into the category which seemed most relevant; for example, an article pertaining to a celebrity response to a political event would have been categorized as 'pop-culture,' because although it might have pertained to politics, the focus of the article was on the celebrity.

In the end, the only category with enough articles to run the analysis on was 'politics.' I ended up with 23 articles pertaining to politics from each of the nine publications, for a total of 207 articles. Using both parts of speech and the LIWC features, I created two self-organizing map models, as seen in Figure 12 on the next page. While the clusters seem to show a more distinct separation of questionable and satirical news articles from mainstream news articles than previously observed, these clusters also lacked sufficient robusticity.



(a) Politics articles based on POS.

(b) Politics articles based on LIWC.

Figure 12: Self-organizing maps of articles pertaining to politics, based on both parts of speech and LIWC features.

5 Discussion

The failure to find any natural categories - even to find a basic distinction between 'real news,' 'satire,' and 'fake news' - is telling. On one hand, such a failure might mean that there is, in fact, no discernible stylistic distinction between the three categories. On the other hand, this might also reveal a lack of consistency within publications; that is, while a publication might publish 'fake' news, it might also publish articles which report factual events in an unbiased or non-sensationalized manner. In order to determine which of the two explanations better explains the lack of distinction between the two categories, a more refined corpus might be necessary.

When designing my methods of analysis, I looked to research that had been done before (as seen in Section 2); for most of this research, analysis depended upon large corpora built by taking many or almost all articles available from a small number of publications. As I did in my analysis, previous research has thus assumed that all articles from a publication can be classified as a single category of news. Perhaps a better method of building a corpus would have instead been to curate articles by manually reading each article and individually assessing the veracity of the articles and the intention of the author. This would make it more feasible to take articles from a much broader range of sources while still upholding the integrity of the corpus. This would also make it easier for articles from the same publication to belong to different categories, and thus make it easier to discern whether the lack of natural categories simply stems from noise in the data, or from an actual lack of stylistic distinction between categories of news.

Another barrier to finding any distinct categories might lie in the news articles themselves. The use of stylistic features to determine 'fake news' from 'true news' often anticipates that 'true news' will be more formally and objectively written, while 'fake news' will be more informally and subjectively written. This expectation perhaps underlies the fact that often ‘real news’ is not only considered to be tied to the veracity of the information it contains, but is also tied to the institutions which publish the articles. News in the United States is constrained by few professional guidelines - there are no standardized exams that one must pass to become a journalist, and no organizations regulate the media to ensure that published news meets a specific set of standards. Instead, news is defined by “a set of cultural practices, informal and often implicit agreements about proper conduct, style, and form” (Baym 2005:261). Thus the journalism sponsored by established news organizations might be more likely to adhere to these cultural practices, while journalism sponsored by organizations which have come into being more recently, and which reach a smaller audience, might be expected to produce less formal or rigorous content that does not meet the cultural standards of news reporting. This underlying supposition about news type and the institutions they are related to can be seen in almost all of the corpus analyses that have previously been done on deception detection. Many of the ‘trusted’ news sources for these corpora were well-established, mainstream media outlets - The Washington Post, The New York Times, etc. While this may be a presupposition to previous analyses, that does not mean that it necessarily holds any water. After all, there is nothing to prevent less-established media outlets or blogs from adhering to these cultural practices, and there is very little to prevent mainstream or well-established media outlets from turning their backs on these practices, particularly in an attempt to maintain or regain leadership.

Another factor to consider is the prevalence of wire services, such as the Associated Press, which allow all news sources to use articles and content not originally written for the publication which ends up using it, with very little adaptation of the article being necessary. As a result, such content might not match the style of the publication that uses it, or might be used by multiple publications, causing noise in a corpus such as this (and those of previous work mentioned earlier in this paper), where articles are selected at random from web pages and receive very little manual review.

In acknowledging the limitations of this analysis, it is also important to recognize that a corpus analysis can only take into consideration that which is present within the corpus. While this sounds obvious, it is often all to easy to forget about what Duguid and Partington (2018:40) refer to as "the somewhat paradoxical absence of something because it is too obvious to mention and taken for granted." This absence generally refers to topics or connotations which are such an integral part of discourse about a topic that they do not need to be specifically referenced to be understood. While an absence of specific words or topics might not have as large of an impact on a stylistic corpus analysis, such as that performed by using parts of speech to tag the corpus, it should still be taken into consideration as a possible source of noise.

6 Conclusion

While deception detection is a topic which is complex and nuanced, as news sources of questionable credibility become a larger part of online media, it becomes a topic which garners greater interest. As my analysis has demonstrated, the lines separating different categories of news are not as clear as they have been assumed to be in previous analyses.

My lack of results does not necessarily mean that corpus-based methods of deception detection are impossible, but my hope going forward is that in the very least greater care will be taken when constructing the corpora for such analyses. I believe that those designing and testing such models need to take greater care in meticulously defining the categories of news they use in their analyses, and that more effort should be put into addressing the nuances that exist both between and with different categories of news.

Appendix

```
#Using Rcrawler to generate website URLS
##once URLS were generated, I saved them as a data frame,
##formatted such that it could be used as an input into the
##BootCaT program.
Rcrawler(Website =”https://www.theonion.com”,
          no_cores = 4, no_conn = 4,
          DIR = ”./onion”)
onion.urls = INDEX[[2]]
onion.urls.2= as.data.frame(onion.urls)
write.table(onion.urls.2, file = ”onion.urls”,
         quote = FALSE,
         sep = ’\t’,
         col.names = FALSE,
         row.names = FALSE)

#creating POS data frame from corpus
##splitting corpus by article
ynw.file = ”/Users/Edie/Documents/Thesis/Corpora/ynwcorpus.txt”
ynw = scan(ynw.file, what = ”character”, sep = ’\n’, comment.char = ’#’)
found_cur_url_logic = grepl(’CURRENT_URL’, ynw)
bad_lines = ynw[found_cur_url_logic]
not_found_cur_url_logic = !found_cur_url_logic
good_lines = ynw[not_found_cur_url_logic]
```

```
break_indexes = grep(’CURRENT_URL’, ynw)
break_indexes_dummy = break_indexes_with_last
for_loop_indexes = 2:length(break_indexes_dummy)
filename_counter = 0
test = 0
for (loop_idx_n in for_loop_indexes) {
  index_start = break_indexes_dummy[loop_idx_n-1] + 1
  index_end = break_indexes_dummy[loop_idx_n] - 1
  filename_counter = filename_counter + 1
  filename = paste(filename_counter ,’txt’, sep=’.’)
  setwd(”/Users/Edie/Documents/Thesis/Corpora/ynw/”)
  test = tolower(ynw[index_start:index_end])
  write(test, file = filename)
}

##using TreeTagger
setwd(”/Users/Edie/Documents/Thesis/Corpora/ynw”)
x = list.files(”/Users/Edie/Documents/Thesis/Corpora/ynw/”)
filename_counter = 0
for (item in x) {
  tagged.text = treetag(
    item,
    treetagger = ”/Users/Edie/TreeTagger/cmd/tree-tagger-english”,
    lang=”en”,
    doc_id= item
  )
  data = taggedText(tagged.text)
  filename_counter = filename_counter + 1
  tagged = paste(”tagged”, filename_counter, sep = ’_’)
  folder = paste(”Tagged”, tagged, sep = ’/’)
  filename = paste(folder ,’txt’, sep=’.’)
  write.table(data, file = filename, quote = FALSE, sep = ’\t’)
}

##generating frequency counts of POS for each article
setwd(”/Users/Edie/Documents/Thesis/Corpora/ynw/Tagged”)
tglist = list.files(”/Users/Edie/Documents/Thesis/Corpora/ynw/Tagged/”)
filename_counter = 0
for (fl in tglist) {
  a = read.csv(fl, header = TRUE, sep = ’\t’, row.names = NULL)
  b = as.vector(a$tag)
  c = table(b)
  d = as.data.frame(c)
  sum.freq = sum(d$Freq)
  freq = d$Freq
  d$freq.m = NA
  d$freq.m = (log10(freq/sum.freq)) + 3
  df = d[-2]
  t = t(df)
  df.f = as.data.frame(t)
  filename_counter = filename_counter + 1
  freq = paste(”freq”, filename_counter, sep = ’_’)
  filename = paste(freq ,’txt’, sep=’.’)
  write.table(df.f, file = filename, quote = FALSE, sep = ’\t’,
        col.names = FALSE)
}

##creating frequency data frame
files = list.files(pattern=”freq”)
files = mixedsort(files)
filelist <- lapply(files, read.csv, header = TRUE, sep = ’\t’)
filenames = c()
for(i in (1:100)) {
  a = paste(”ynw”, i, sep = ’_’)
  filenames = c(filenames, a)
}
names(filelist) = filenames
myfiles = ldply(filelist)



#implementing the SOM
##importing the data frame
setwd(”/Users/Edie/Documents/Thesis/Corpora”)
data.list = list.files(pattern = ”data_frame”)
dl = data.list[-1]
filelist <- lapply(dl, read.csv, header = TRUE, sep = ’\t’)
names(filelist) = dl
myfiles = ldply(filelist)
```

```
myfiles[is.na(myfiles)] = 0


##generating the SOM
som_d = myfiles[-1]
som_d_m = as.matrix(scale(som_d))
set.seed(5)
som_grid <- somgrid(xdim = 14, ydim= 11, topo="hexagonal")
som_model <- supersom(som_d_m,
                      grid=som_grid,
                      rlen=100,
                      alpha=c(0.05,0.01),
                      keep.data = TRUE)



#performing hierarchical clustering
som_dis = dist(som_model$codes[[1]])
som_hclust = hclust(som_dis, method="ward.D2")
kclust = cutree(som_hclust, 3)



#plotting/mapping som objects
pretty_palette <- c("#1f77b4","#ff7f0e","#2ca02c", "#d62728",
          "#9467bd","#8c564b","#e377c2")
shapes = c(21, 24, 23, 22, 25, 4)
x = myfiles
x$type = NA
```

```
x$type[1:100] = "satire"
x$type[101:200] = "questionable news"
x$type[201:300] = "questionable news"
x$type[301:400] = "mainstream news"
x$type[401:500] = "satire"
x$type[501:600] = "mainstream news"
x$type[601:700] = "satire"
x$type[701:800] = "mainstream news"
x$type[801:900] = "questionable news"
x$type = as.factor(x$type)

dev.new(width = 11, height = 5)
plot(som_model, type="mapping",
        pchs = shapes[match(x$type, levels(x$type))],
        col = "black", bg = "white", bgcol = pretty_palette[kclust],
        main = "POS Clusters")
add.cluster.boundaries(som_model, kclust)
legend(15,7, pch = shapes, legend = levels(x$type),
        text.col = "black", bty = "n")

plot(som_model, type = "dist.neighbours", main = "U-Matrix")
add.cluster.boundaries(som_model, kclust)


#subsetting clusters
ind.group <- kclust[som_model$unit.classif]
```

```
print(ind.group)
clusters = as.data.frame(ind.group)
obj_id = as.data.frame(mixedsort(as.vector(myfiles$.id)))
type = x["type"]
groups = cbind(obj_id, clusters, type)
cluster1 = subset(groups, ind.group == 1)
cluster2 = subset(groups, ind.group == 2)
cluster3 = subset(groups, ind.group == 3)


#creating the politics data frame
##indexes of articles pertaining to politics
pol.ch = ch[c(4,6,10,14,22:34,36,41,47,50,51,55,61:66,68,77),]
pol.dw = dw[c(1:6,8:11,12:18,20:30,32,34,35,36,38,39,41:45,47,
           49,53,54,57,58,60,61,63,64,66,69,71:73,75:78,80,
           82,83,84,86,87,88,90,94,97:100),]
pol.ls = ls[c(1:20,24:30,33:37,39:41,44,49:52,54,55,58,60,61,
           63,64,66,68,69,72,74,76,78,80,81,83,84,85,87,88,
           90,93,94,95,96,97,98,100),]
pol.nyt = nyt[c(1:4,6,8,9,11,13,14,16:19,21,23,25,26,27,30,31:35,
            37,39,40,42,44:58,60:64,66:77,79:89,91,92,94:97,
            99,100),]
pol.on = on[c(2,6,9,17,21,33,34,40:42,44,46,48,52,60:62,83,
           86:97),]
pol.re = re[c(5:17,23:25,28:36,45,46,52,58:61,63,64,66:74,77,88,
           89,96,100),]
```

```
pol.sw = sw[c(1,2,4,16,17,19,25,38,60,63,68,70,71,74,76,80,
           83,85,86,89,91,92,93),]
pol.wapo = wapo[c(2,3,4,11,17,21:23,27:29,36,37,39,40,41,42,44,
             45,46,47,48,64,65,67,75,82,84),]
pol.ynw = ynw[c(2,6,8,11,14,15,18,21,22,28,30,32,33,36,40,42,43:46,
            50:52,58,61:63,65,70,72,76,79,80,85,89,92,93,
            95,96,99),]

##randomly sampling to generate data frame
s.pol.ch = sample_n(ch,23)
s.pol.dw = sample_n(dw,23)
s.pol.ls = sample_n(ls,23)
s.pol.nyt = sample_n(nyt,23)
s.pol.on = sample_n(on,23)
s.pol.re = sample_n(re,23)
s.pol.sw = sample_n(sw,23)
s.pol.wapo = sample_n(wapo,23)
s.pol.ynw = sample_n(ynw,23)

f.pol = rbind(s.pol.ch, s.pol.dw, s.pol.ls, s.pol.nyt, s.pol.on,
              s.pol.re, s.pol.sw, s.pol.wapo, s.pol.ynw)
write.table(f.pol, file = ”pol_df.txt”, quote = FALSE, sep = '\t',
            row.names = FALSE, col.names = TRUE)


#robusticity tests
```

```
##randomly selecting 90\% of the data from each cluster
r2.1 = sample_n(cluster1, 15)
r2.2 = sample_n(cluster2, 134)
r2.3 = sample_n(cluster3, 660)

r = rbind(r2.1, r2.2, r2.3)
r = r[1]
colnames(r) = ".id"

##using ids of reduced data to subset from the full data frame
r = inner_join(r, myfiles, by = ".id")

##identifying objects that occur in the same cluster
##from one model to the next
r.c.1 = cluster1_r[1]
colnames(r.c.1) = ".id"
r.c.2 = cluster2_r[1]
colnames(r.c.2) = ".id"

c.1 = cluster1[1]
colnames(c.1) = ".id"
c.2 = cluster2[1]
colnames(c.2) = ".id"

s.clust.1 = inner_join(r.c.1, c.1, by = ".id")
s.clust.2 = inner_join(r.c.2, c.2, by = ".id")
```

References

Baroni, Marco & Silvia Bernardini. 2004. Bootcat: Bootstrapping corpora and terms from the web. In *Proceedings of lrec 2004*, 1313–1316.

Baym, Geoffrey. 2005. The daily show: Discursive integration and the reinvention of political journalism. *Political Communication* 22(3). 259–276.

Dickey, Thomas E. 2017. Lynx. lynx2.8.8.

Duguid, Alison & Alan Partington. 2018. *Corpus approaches to discourse: A critical review* chap. Absence: You don't know what you're missing. Or do you?, 38–59. New York: Routledge.

Kahil, Salim. 2017. Rcrawler: Web crawler and scraper. R package version 0.1.7-0.

Kind, Marrias Carrasco & Robert J. Brunner. 2014. Somz: Photometric redshift pdfs with self-organizing maps and random atlas. *Monthly notices of the royal astronomical society* 438(4). 3409–3421.

Michalke, Meik. 2018. *korpus: An r package for text analysis.* Version 0.11-2.

Potthast, Martin, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff & Benno Stein. 2017. A stylometric inquiry into hyperpartisan and fake news. *CoRR* .

Rashkin, Hannah, Eunsol Choi, Jin Yea Jang, Svitlana Volkova & Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2921–2927. Copenhagen, Denmark: Association for Computational Linguistics.

Rubin, Victoria L., Yimin Chen & Niall J. Conroy. 2015. Deception detection for news: Three types of fakes. In *Proceedings of the 78th association for information science and* *technology annual meeting*, Saint Louis, Missouri: Association for Information Science and Technology.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, Manchester, UK.

Wehrens, Ron & L.M.C. Buydens. 2007. Self- and super-organising maps in r: the kohonen package. *J. Stat. Softw.* 21(5).

Yang, Fan, Arjun Mukherjee & Eduard Gragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2921–2927. Copenhagen, Denmark: Association for Computational Linguistics.