Tell Me Who Your Friends Are: Using Content Sharing Behavior for News Source Veracity Detection

Maurício Gruppi*, Benjamin D. Horne†, and Sibel Adalı*
Rensselaer Polytechnic Institute*, University of Tennessee Knoxville†
gouvem@rpi.edu, bhorne6@utk.edu, adalis@rpi.edu

Abstract

Stopping the malicious spread and production of false and misleading news has become a top priority for researchers. Due to this prevalence, many automated methods for detecting low quality information have been introduced. The majority of these methods have used article-level features, such as their writing style, to detect veracity. While writing style models have been shown to work well in lab-settings, there are concerns of generalizability and robustness. In this paper, we begin to address these concerns by proposing a novel and robust news veracity detection model that uses the content sharing behavior of news sources formulated as a network. We represent these content sharing networks (CSN) using a deep walk based method for embedding graphs that accounts for similarity in both the network space and the article text space. We show that state of the art writing style and CSN features make diverse mistakes when predicting, meaning that they both play different roles in the classification task. Moreover, we show that the addition of CSN features increases the accuracy of writing style models, boosting accuracy as much as 14% when using Random Forests. Similarly, we show that the combination of hand-crafted article-level features and CSN features is robust to concept drift, performing consistently well over a 10-month time frame.

1 Introduction

"Tell me who your friends are, and I'll tell you who you are." - Unknown

The spread of false and misleading news is damaging to society (Lewandowsky et al. 2012; Lazer et al. 2018). Its harms can be felt across many parts of society, including politics (Allcott and Gentzkow 2017), education (Alvermann 2017), and health (Speed and Mannion 2017; Singh et al. 2020; Pennycook et al. 2020). Due to this cost, limiting false and misleading news has become a concern for both researchers and practitioners.

Due to the scale of this problem, many researchers have built classifiers to automatically assess the veracity of news (Kumar and Shah 2018). The vast majority of these newly-developed classifiers are based on features of the text in news articles or claims (Baly et al. 2019; Potthast et al. 2017; Popat et al. 2016; Horne, Gruppi, and Adalı 2020). These text-based methods have been shown to work well in lab-settings because unreliable news is often written in a different style than reliable news, employing many different linguistic and grammatical markers. These differences are often attributed to various factors, such as the use of moral-emotional language to gain engagement (Brady et al. 2017). Despite this success, there are still concerns about the robustness of these methods. Specifically, text-based methods are prone to performance degradation over time (often called *concept drift*) due to the dynamic attributes of the news cycle (Horne, Nørregaard, and Adali 2019b). Furthermore, text-based models may be dependent on language or over-fit to specific domains or topics, making them less generalizable.

In this paper, we present an alternative and complementary method for detecting unreliable information based on the behavior of news producers. Specifically, past work has documented that many news producers copy news stories from each other. In essence, copying is a type of amplification, making a story available to the readers of a specific source. In mainstream media, this has been attributed to meeting the demand of all-day news consumption (Boczkowski 2010). However, this behavior is very common in alternative media as well, with different motivations. These motivations include generating engagement at a low cost, increasing perceived credibility of stories, and their algorithmic visibility in social media platforms. It has been shown that when this behavior is formulated as a network, the community structures found in the network correspond to different types of news sources in the media ecosystem, including mainstream media, hyper-partisan media, and more (Starbird 2017; Horne, Nørregaard, and Adalı 2019a).

Building on the pervasive nature of content sharing among news producers, we propose a new set of source veracity features using content sharing networks, or 'CSN' for short. Our hypothesis is that the network location of sources in the CSN can provide a strong signal of source reliability. We introduce three feature sets using CSNs, one set based on well-known network properties of nodes and two sets using network embedding methods. To show the effectiveness of CSN features, we conduct a thorough study comparing them to previously studied text-based features. Furthermore, we test the stability of different models to over time. Through our comprehensive study, we show that CSN in-

formation alone outperforms the previously used text-based methods. Despite the high accuracy of CSN-only models, the combination of CSN information and text information works best, increasing accuracy by at most 14.7% over text-only models. We also show that the combination of CSN models and text-based models provide stable performance over time. Additionally, we find that text and CSN models are highly complementary: they make different types of errors in our data set. The text models make fewer errors when predicting reliable sources and the CSN models make fewer errors when predicting unreliable sources.

In short, using the content sharing behavior of news sources in veracity detection leads to highly accurate models. By adding complementary information to existing text models, we improve the overall performance and enhance model robustness.

2 Related Work

There is a large body of work on news veracity detection, particularly focused on political news articles since 2016 (Kumar and Shah 2018). These works have used a variety of machine learning techniques. These techniques include binary supervised models (Baly et al. 2018; Horne et al. 2018; Horne, Nørregaard, and Adali 2019b; Castelo et al. 2019; Cruz et al. 2019), multi-class supervised models (Baly et al. 2019), semi-supervised models (Guacho et al. 2018; Agerri 2019), unsupervised models (Hosseinimotlagh and Papalexakis 2018), and various Neural Network models (Singhania, Fernandez, and Rao 2017; Li and Goldwasser 2019; Färber, Qurdina, and Ahmedi 2019; Moreno et al. 2019). Some works have also framed the problem as a ranking problem, rather than a classification problem (Ye and Skiena 2019; Barrón-Cedeno et al. 2019). The primary features of these detection methods are based on the article text, many of which are hand-crafted feature sets. These text features range from very specific, such as the bias and emotion in an article, to very generic, such as the term frequency within an article. In general, these types of features have been shown to work well and can be used to explain algorithm decisions, but they are prone to sub-optimal performance over time and across domains. Theoretically, they are also prone to text manipulation from malicious sources (Horne, Nørregaard, and Adali 2019b), although this behavior has not yet been shown in real life.

One method of strengthening these text-based models is to augment them with features unrelated to the content of the article. To some degree, this has been done. Baly et al. add the presence of a Wikipedia page and Twitter account for each source (Baly et al. 2018) to article-related feature models. Similarly, Li and Goldwasser use both text features and Twitter social features to detect veracity (Li and Goldwasser 2019). Ye and Skiena add the number of advertisements on a page and the popularity of the source to text-based ranking models (Ye and Skiena 2019). Castelo et al. add various web markup features such as the presence of an article author, number of advertisements, and number of images (Castelo et al. 2019). However, with the exception of the number of advertisements, these additional features can be easily manipulated with little cost to the malicious news producer.

Mixing text features with source-level features has also been done in false claim and rumor detection (rather than news article or news source veracity detection). Many studies of false claims on Twitter utilize features of the users who spread the claim, such as number of followers, number of friends, age of profile, or temporal patterns of the user posts (Castillo, Mendoza, and Poblete 2011; Ruchansky, Seo, and Liu 2017; Yang et al. 2012). Other claim veracity works have used popularity as a feature (Popat et al. 2016). Again, these additional non-text features are shallow and easy to manipulate.

In this paper, we address this gap by introducing a new source-level, behavioral feature for the news source veracity prediction task, namely content sharing behavior. This behavior is costly to manipulate and highly consistent over time, which lends itself to building robust prediction models for the task. This cost stems from the additional effort malicious news producers would need to exert to produce independent false content by not copying content from their peers. Further discussion of network construction and the intuition behind using content sharing networks as signals of veracity can be found in Section 4.

3 Data

In this work, **given a news article from an unknown source, our goal is to predict if the source of the article is reliable or unreliable**. To this end, we extract news article data from the NELA-GT-2018[1] data set (Nørregaard, Horne, and Adalı 2019). The NELA-GT-2018 data set is a political news data set that contains 713K articles from 194 sources, containing all articles by these sources from February 1st, 2018 to November 30th, 2018. These sources come from a wide range of mainstream and alternative media, including many conspiracy-spreading news sources and hyper-partisan blogs. Included in the NELA-GT-2018 data set are source-level labels of credibility from several assessment platforms. Two of the assessment platforms will be used for labeling sources in this paper: Open Sources and NewsGuard[2]. Open Sources ratings have been used in many other studies. It uses a panel of experts to mark sources as one or more of these 13 categories: *reliable, blog, clickbait, rumor, fake, unreliable, biased, conspiracy, hate speech, junk science, political, satire,* and *state news*. The criteria for deciding source labels on Open Sources is available on their website. NewsGuard is an independent journalistic organization that similarly uses a group of experts to score news sources based on credibility and transparency using a stringently developed rating process. Specifically, NewsGuard rates sources on the following criteria, with each criteria having an assigned weight:

1. Does not repeatedly publish false content (22 points)
2. Gathers and presents information responsibly (18 points)
3. Regularly corrects or clarifies errors (12.5 points)
4. Handles the difference between news and opinion responsibly (12.5 points)

[1] dataverse.harvard.edu/dataverse/nela

[2] http://www.newsguardtech.com/

5. Avoids deceptive headlines (10 points)
6. Website discloses ownership and financing (7.5 points)
7. Clearly labels advertising (7.5 points)
8. Reveals who's in charge, including any possible conflicts of interest (5 points)
9. Provides information about content creators (5 points)

Using these two sets of source-level labels, we create two classes of news: `reliable` and `unreliable` as follows. We extract all articles from sources that have a credibility score above 90 according to NewsGuard to create our reliable class and sources that have a credibility score below 40 or sources that are marked as unreliable/conspiracy/fake by Open Sources to create our unreliable class. Often sources with a score below 40 by NewsGuard are also marked as unreliable/conspiracy/fake in Open Sources. To obtain a score above 90 by NewsGuard, a source would only be allowed to miss one of the last four criteria (criteria 6, 7, 8, or 9).

Based on this labeling method, we extract **184736 articles** from **52 sources**, where 25 sources are marked as reliable and 27 are marked as unreliable. These articles cover 10 months in 2018 (February through November). The sources in each class can be found in Table 1.

(R) Reliable sources	(UR) Unreliable sources
Reuters	True Pundit
NPR	Natural News
USA Today	Infowars
CNN	Veterans Today
The New York Times	Activist Post
CBS News	Mint Press News
WSJ Washington Wire	Waking Times
The Hill	Intellihub
CNBC	NODISINFO
PBS	TheAntiMedia
The Guardian	Freedom Daily
Politico	FrontPage Magazine
The Denver Post	Conservative Tree House
BBC	Shareblue
Business Insider	Bipartisan Report
Washington Examiner	Newswars
Yahoo News	Prison Planet
The Daily Beast	The Gateway Pundit
Real Clear Politics	Pamela Geller Report
National Review	Western Journal
New Yorker	The Political Insider
Fortune	The Duran
Newsweek	Instapundit
Mercury News	Palmer Report
The Atlantic	Freedom Outpost
	The Right Scoop

Table 1: Sources used in each class. Note, for BBC we only extract article from their U.S. news feed. These labels are based on external labeling. See Section 3 for more details.

4 Using Content Sharing Networks as a Signal of Reliability

Several recent studies have shown that both mainstream and alternative news sources often share (or copy) articles from each other either verbatim or in part (Starbird et al. 2018; Horne, Nørregaard, and Adalı 2019a). The motivation behind this content copying can differ greatly depending on the source. Mainstream sources copy articles from news-wire services often to meet demand or "break" news in a timely manner. Conspiracy sources may employ this tactic with malicious intent to spread false content, create uncertainty surrounding an event by amplifying alternative narratives, or to simply make money from clicks (Horne, Nørregaard, and Adalı 2019a; Starbird et al. 2018; Braun and Eklund 2019). This behavior may also indicate coordination between disinformation producers.

This article sharing behavior can be formulated as a network where each node is a news source and each directed edge $A \rightarrow B$ has weight proportional to the number of articles in B that are copied from A. This network captures various important aspects of the news ecosystem: communities of similar media sources, hubs of conspiracy news production, and bridges between the mainstream and alternative media. It is likely that these network structures, particularly community membership, provide a strong signal of veracity. It is easy to imagine that an unknown news producer, which copies articles from a well-known conspiracy news producer, is also a source of conspiracy news. This signal can be extended to more indirect cases where unknown news sources fall in a path between two known news sources, or sources that copy from both reliable and unreliable sources can be labeled as mixed veracity. It is this rich structure of information that we wish to take advantage of in detecting articles from reliable and unreliable sources.

4.1 Network Construction

Using the whole NELA-GT-2018 data set (rather than our extracted labeled data set described in Section 3), we follow the process described in (Horne, Nørregaard, and Adalı 2019a) to create a near-verbatim content sharing network (CSN) of news sources. Specifically, we compute a TF-IDF matrix of all articles in the data set and compute the cosine similarity between each article vector pair (given that each article comes from a different news source). To reduce the complexity of this process, we use a sliding 5 day window of articles. For each pair of article vectors that have a cosine similarity greater or equal to 0.85, we extract them and order them by the timestamps. This is the same cosine similarity threshold used in both (Horne, Nørregaard, and Adalı 2019a) and (Starbird et al. 2018). This process creates a directed graph $G = (V, E)$, where V is the set of news sources and E are directed weighted edges representing articles shared. Edges are directed towards publishers that copied articles (inferred by the timestamps). We normalize the weight of each edge in the network by the number of articles published in total by the source. For example, if USA Today publishes 1000 articles and copies 100 of those articles from Reuters, the edge from USA Today to Reuters would have weight 0.1 and be directed towards USA Today.

We show a visualization of this constructed network in Figure 1. We built this visualization using Gephi and used the Newman Spectral Method for directed modularity to label community membership (Newman and Girvan 2004).

Specifically, we use the default parameters from Zhiya Zuo's modularity maximization Python package[3]. This network includes the 52 sources with known labels used in this study as well as 88 sources with no labels. The presence of both labeled and unlabeled sources provides us with a rich network structure. In addition, the community structures in the network (as shown by colors in Figure 1) would likely be lost if only labeled data was used. We find that the community structure looks very similar to the structure displayed in (Horne, Nørregaard, and Adalı 2019a), as we use the same dataset and the same community detection method. To provide some intuition of where labeled sources are placed in the network, we show the number of sources from each class in each network community in Figure 2. We also show the degree distributions of labeled sources in Figure 3. Both Figures support the idea that `reliable` and `unreliable` sources are represented differently in the CSN, hence are potentially valuable in news veracity detection.

We choose to focus on near-verbatim content sharing networks in this paper due to the well-studied properties of these networks. Partial content sharing networks can also provide useful additional information, however these networks are not yet studied in the literature. Hence, we leave study of partial sharing behavior to future work.

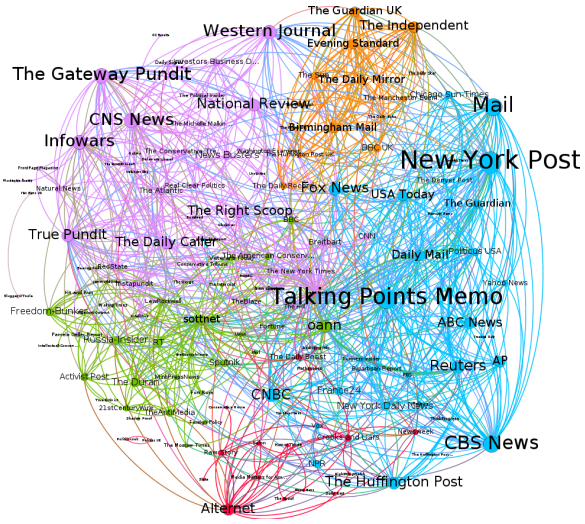


Figure 1: Visualization of CSN using Gephi (Bastian, Heymann, and Jacomy 2009). Colors represent communities using directed modularity. Edges are directed, where the outdegree of node n is how many news sources copy articles from node n. The size of each node is based on outdegree. Just as shown in (Horne, Nørregaard, and Adalı 2019a), each community contains sources from distant parts of the media landscape, often grouping sources on similar veracity. In particular, we can see many of our unreliable sources in the magenta and green communities, while our reliable sources fall mostly within the blue community.

[3]zhiyzuo.github.io/python-modularity-maximization/

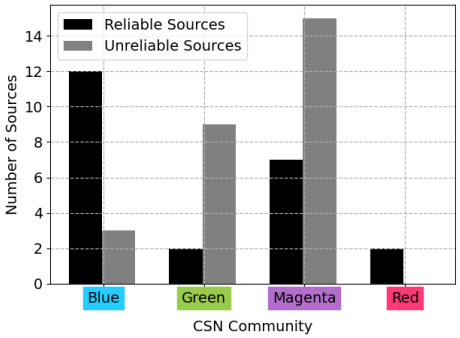


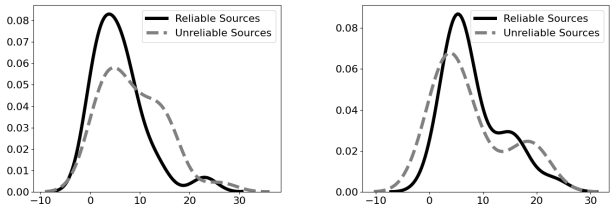
Figure 2: Number of labeled sources in each community, where the colors of the x axis labels correspond to communities in Figure 1. The high separation between `reliable` and `unreliable` labeled sources supports the intuition that the CSN network can be used to approximate veracity.

4.2 Network Representation for Classification

Hand-crafted Network Features (HCNF). One way to represent sources in the CSN is to craft a set of network features for each source. To do this, we choose several standard network measures, as well as more community-focused features. In total we compute 11 features that include:

1. Community - What community is the source in, as determined by directed modularity (Newman and Girvan 2004).
2. Weighted Outdegree - Number of articles copied from the source.
3. Weighted Indegree - Number of articles the source copies.
4. Closeness Centrality - The reciprocal of the sum of the shortest path distances from the source to all other sources.
5. Betweenness Centrality - The sum of the fraction of all-pairs shortest paths that pass through the source.
6. Eigenvector Centrality - The centrality of a source based on the centrality of its neighboring sources.
7. Community Core - Is the source a member of the k-core of its community or not, where the k-core is a maximal subgraph that contains nodes of degree k or more. We compute the the core with the largest degree.
8. Inside Source Edges - Number of articles copied from the source by sources in its community.
9. Inside Sink Edges - Number of articles the source copies from other sources in its community.
10. Importing Edges - Number of articles the source copies from sources outside of its community.
11. Exporting Edges - Number of articles copied by the source from sources outside of its community.

Node2Vec (N2V). Another, likely more complete, method to represent the CSN network is network embedding. Specifically, we use the Node2Vec (Grover and



(a) In-Degree Distribution (b) Out-Degree Distribution

Figure 3: Degree distributions `reliable` and `unreliable` labeled sources in CSN, where In-Degree (a) represents how much a source is copying articles and (b) Out-Degree represents how much a source is copied from. As discussed in literature (Horne, Nørregaard, and Adalı 2019a), unreliable sources generally copy more articles verbatim than reliable sources.

Leskovec 2016) network embedding method. As with word embedding, Node2Vec uses the skipgram model and transforms the sparse adjacency matrices of networks into a dense vector representation of nodes. This representation aims to preserve network structure and node neighborhood, clustering together those nodes with similar functionality and structure in the network, such as hubs and peripheral nodes. Additionally, the dense vector representation captures latent similarity relations within the network. Node2Vec uses the return parameter p and the in-out parameter q to control the breadth and depth of random walks on the network used to generate the embedding. In this work, we use $p = 0.5, q = 0.5$ and set the vector dimension to 40. The output are vectors representing the network nodes (news sources), we refer to these vectors as N2V features. Note that we embed all sources in our dataset, including those with no labels to fully represent the CSN. Note, we remove all articles used in the CSN construction from our training and test data in later experiments in order to avoid data leakage.

NetworkText2Vec (NT2V). Naturally, the CSN embedding can only represent sources that share content. The sharing behavior may be rare and not present in specific settings. Furthermore, not all sources may share content in verbatim, especially if they are new or are representing different topics. For example, a source may focus on breaking news, which lends itself to content sharing, while another source may focus on investigative pieces, which may not lend itself to content sharing. Although these sources can be represented as completely disconnected nodes in the network, embedding disconnected nodes with N2V would give us no relevant information with respect to node similarity. To fill this gap, we can use the similarity of sources with respect to the text that they publish, using text as side information in the embedding.

The problem of attributed network embedding was addressed by Yang et al. (Yang et al. 2015), they proposed TADW, a method that performs matrix factorization on a matrix using as input both network and text features. One limitation of this method is that it requires network and text representation of every source being embedded. In our case, this is a major drawback, as CSNs can have missing edges due to lack of content sharing information. To mitigate this issue, we propose a method based the multi-scale attributed network embedding by (Rozemberczki, Allen, and Sarkar 2019) that we refer to as NT2V . This method takes as input both the CSN and the text attributes of news articles. The text attributes are a representation of a source given by the average of its word embedding vectors. Using this information, NT2V combines two random walks based on the similarity of nodes (news sources): one over the network as in Node2Vec and the second one over the text attributes. More formally, let c_i be the i-th source from a random walk over the text corpus, the transition probabilities from c_i to c_{i+1} are obtained from the cosine similarity between c_i and its k-nearest neighbors, and normalized by the sum of weights of the edges leaving c_i. Sources with higher cosine similarities have higher chances of being picked in the random walk, thus appearing more often in contexts. We set a lower bound cutoff similarity of 0.5 to prevent selecting sources that are significantly dissimilar.

Intuitively, the process of context generation is carried out by interchanging random walks over the network space and the text space. At random walk j we decide with probability t that the walk will happen over the text space, or $1 - t$ that it will happen over the network. If the network is chosen, we perform a random walk entirely over the network, as with Node2Vec, otherwise the random walk is entirely over the text corpus space. We generate n contexts for each source. Once contexts are generated, they are used as the input to a skipgram model. In addition to the input parameters p and q of Node2Vec, NT2V requires the t parameter that controls the likelihood of performing a walk over the text and the k parameter that controls the number of nearest neighbors to consider during the text corpus walk. The output is vector representations of sources based on the generated contexts.

We set the output vector size to 40, number of walks to 1000, walk length to 80, and tune parameters p, q and t by performing a grid search over the interval $[0.2, 0.8]$ with a step size of 0.1. We select the model that yielded the best classification accuracy on a validation set. The final parameters are: $t = 0.8$, $p = 0.5$, $q = 0.4$. Code for NT2V and additional documentation are publicly available[4]. We uniformly sample 20% of the articles for each source to use in NT2V, the sampled articles are used exclusively to compute the source representation and are not used in any other scenario. This is done in order to avoid data leakage from the source representation into the article level experiments.

5 Baseline Text Models

To compare our CSN feature models to state-of-the-art text-based methods, we compute several text feature sets and discuss the details of each below.

NELA. NELA is a hand-crafted, text feature set used in whole or in part in several news veracity studies (Horne et al. 2018; Horne, Nørregaard, and Adali 2019b; Baly et al. 2018; 2019; Cruz et al. 2019; Barrón-Cedeno et al. 2019)

[4]https://github.com/mgruppi/NewsNetworkEmbedding

with available code online[5]. This feature set can be divided into five different groups:

1. Style - This group represents the general writing style of an article, including parts-of-speech used, punctuation used, and capitalization used.
2. Complexity - This group measures the complexity of writing in an article, including lexical diversity, reading difficulty, length of word, and length of sentences.
3. Bias - This group measures the bias of an article. It uses lexicons developed in (Recasens, Danescu-Niculescu-Mizil, and Jurafsky 2013) to capture various signals of bias in text, such as hedges, factives, assertives, and opinion words.
4. Affect - This group measures the emotion and sentiment of text using two well-known works in text processing: LIWC (Tausczik and Pennebaker 2010) and VADER (Hutto and Gilbert 2014). LIWC is a gold-standard, lexicon based method for discovering various social and psychological traits in text. These include various types of emotion, such as anger, anxiety, affect, and swear words. VADER is a state-of-the-art sentiment detection tool that provides measures of positive, negative, and neutral emotion in text.
5. Moral - This feature group is a lexicon based method that measures morality in text on the basis of Moral Foundation Theory (Graham et al. 2013). Examples of these features include fairness, authority, and care.

In total NELA contains 194 features, computed independently on the body text and title text of an article.

FastText (FT). Another method we can use to capture textual differences between news articles is using word embedding. Word embedding features have been used in only a few news veracity detection studies so far (Singhania, Fernandez, and Rao 2017) and are still under-explored. The potential advantage of word embedding features over hand-crafted feature sets, like NELA, is that features can be automatically captured regardless of language and domain. The disadvantage is that we cannot control the specific concepts captured in the text, which may lead to worse performance and robustness.

In this work, we use the `wiki-news-300d-1M`[6] pre-trained FastText model (Bojanowski et al. 2017) to obtaine the representation for 184736 news articles, the model was pre-trained on Wikipedia and news data, contains 1 million words and the vector dimension is 300. To obtain the representation of an entire news article we average the vectors of all the words in an article's title and content, thus, arriving at the final representation of an article, given by a 300 dimension article vector which we refer to as FT features.

Note, we also experiment with a LSTM sequence classifier and BERT embedding vectors as baseline text models, but due to the similarity of results across the text models and space restrictions, we do not display those results.

[5]https://pypi.org/project/nela-features/

[6]https://fasttext.cc/docs/en/english-vectors.html

6 Results

6.1 CSN features improve the accuracy of text-based models

Again, the goal of our classification model is to predict if the source of a news article is reliable or unreliable, given a news article and its source name as input. To this end, we train Random Forest classifiers on 80% of the sources and test on 20% of the sources. For each source, we uniformly sample 1000 articles before splitting into train and test sets to ensure each test set is balanced. Note, we are simulating a setting in which the classifier is given an individual news article from an unknown source as input and uses both article-related features and source-related features to predict. If a source is selected for testing, all 1000 of its sampled articles are removed from training. We repeat this experiment 50 times and average the performance metrics. We also repeat these experiments using a fully-connected Neural Network classifier, but find little to no improvement over the Random Forest Classifier, hence we only display the results using Random Forest.

To assess how much CSN features and text features contribute to distinguishing articles from reliable and unreliable sources, we test each individual feature group as well as combinations of article-level text features with their respective source-level CSN features. We combine text and CSN features in two ways:

1. We concatenate text and CSN vectors (represented with a plus sign, e.g. NELA+N2V) and predict using a single binary classifier, or
2. We use a feature ensemble of two binary classifiers, one trained on text features and the other trained on CSN features, using the sigmoid function to predict a probability that the given input belongs to class 0 (reliable). Those probabilities are then combined using a soft voting.

Table 2 shows the classification results for all feature group combinations and classification algorithms. As shown in Table 2, both the hand-crafted text model (NELA) and the word embedding model (FT) are improved by the CSN features (N2V and NT2V). These improvements are significant, increasing accuracy as much as 20%. Based on overall accuracy, the best model is FT+N2V, while the feature Ensemble using NELA shows the best F1 and Recall scores.

While the best performing models are all using combinations of the CSN features and the text features, we do see the CSN models alone also perform well. In fact, N2V has the best precision score among all models and has only a 3% decrease in accuracy from the best combination model, demonstrating the strong signal provided by the CSN.

6.2 Text models and CSN models often make different mistakes

It is clear that CSN features capture some signal of veracity and improve upon the text-based models. However, do CSN models make the same mistakes as the traditionally used text models? To test this, we use two methods. First, we compute the conditional probabilities that a feature correctly classifies the articles given that another feature set has failed to

	Feature Group	Accuracy	F1 Score	Precision	Recall
TX only	**NELA**	0.689	0.685	0.692	0.741
	FT	0.636	0.613	0.682	0.637
	FT+NELA	0.681	0.685	0.700	0.719
NT only	**HCNF**	0.778	0.773	0.827	0.767
	N2V	0.802	0.813	**0.860**	0.815
TX + NT	**NELA+HCNF**	0.733	0.723	0.768	0.736
	FT+HCNF	0.730	0.707	0.739	0.735
	NELA+N2V	0.789	0.791	0.808	0.789
	FT+N2V	**0.836**	0.820	0.841	0.842
	NT2V	0.802	0.803	0.790	0.860
	NELA+NT2V	0.788	0.773	0.802	0.835
	FT+NT2V	0.805	0.806	0.824	0.834
	Ensemble between NT2V & NELA	0.817	**0.823**	0.798	**0.893**
	Ensemble between NT2V & FT	0.806	0.793	0.798	0.840

Table 2: Average performance scores over 50 runs of 20% sources as a test set. NT2V params: $t = 0.8$ $p = 0.5$ $q = 0.4$. The best scores in each category are shown in bold font. See Section 6.1 for details on + and ensemble combinations.

classify it, shown in Table 3. More precisely, given feature sets A and B, we compute $P(p_B = 1|p_A = 0)$ as the *conditional accuracy*, where $p_B = 1$ is the event where feature set B correctly classifies an article, and $p_A = 0$ is the event where feature set A does not correctly classify the same article. The probabilities were computed using a classification model trained on a *leave one source out* subset of articles. Specifically, for each source s, let $\mathcal{S}$ be the articles from s in the data $\mathcal{D}$. We train a Random Forest classifier on $\mathcal{D} - \mathcal{S}$, and test the classification on $\mathcal{S}$. The conditional probability indicates how many of the mistakes of A are corrected by B, and it is given by $P(p_B = 1|p_A = 0) = \frac{P(p_B=1) \cap P(p_A=0)}{P(p_A=0)}$.

Second, we examine the distribution of errors per class for each feature group, shown in Table 4. Simply put, using the *leave one source out* method, we calculate what proportion of the wrong classifications are in each class. This analysis shows us which feature groups are better or worse at classifying one class or the other.

As shown in Table 3, the CSN models (HCNF, N2V, NT2V) made very different mistakes than the text models (NELA, FT), with at most a 83% chance of a CSN model correctly classifying an article that a text model missed. When reversing the probability, we similarly see different mistakes made, with at most a 66% chance of a text model correctly classifying an article that a CSN model missed. When looking at the specific types of mistakes made, we see several consistent cases. Generally, we see the same trend in Table 4. Specifically, we see that both NELA and FT (text models) are better at classifying the `reliable` class than the `unreliable` class, while N2V and NT2V (network models) are much better at classifying the `unreliable` class than the `reliable` class. Higher conditional probabilities imply greater distinction between the errors made by one feature group and the other.

Overall, there are very few mistakes by the CSN features, but when they do make mistakes, it is on sources in sparsely labeled areas of the network. For example, in our data set, Reuters and The Guardian, are often mis-classified by N2V

A	B	$P(p_A = 0)$	$P(p_B = 1\|p_A = 0)$
FT	**N2V**	0.38	0.79
N2V	**FT**	0.17	0.54
FT	**NT2V**	0.38	0.72
NT2V	**FT**	0.23	0.35
NELA	**N2V**	0.33	0.83
N2V	**NELA**	0.17	0.66
NELA	**NT2V**	0.33	0.73
NT2V	**NELA**	0.23	0.63
FT	**NELA**	0.38	0.43
NELA	**FT**	0.33	0.33
FT	**HCNF**	0.38	0.75
HCNF	**FT**	0.18	0.45
NELA	**HCNF**	0.33	0.75
HCNF	**NELA**	0.18	0.54
N2V	**HCNF**	0.16	0.61
HCNF	**N2V**	0.18	0.66

Table 3: Conditional probabilities of mistakes made by each feature set. $P(p_A = 0)$ is the probability of feature set A making a mistake, $P(p_B = 1|p_A = 0)$ is the conditional accuracy defined as the probability that feature set B correctly classifies samples given that feature set A failed to do so. The higher the probability, the more dissimilar the mistakes made by each feature set is. Each model uses Random Forest. We use bold font to indicate the highest dissimilarity between CSN models and text models and vice versa.

(i.e. purely CSN information). This mistake is because articles from both Reuters and The Guardian are often copied by U.K mainstream sources, which are unlabeled in our data set. Thus, when Reuters or The Guardian are left-out for testing, the model has very few, if any, examples of reliable sources in the same U.K neighborhood. However, when text is added to the model in some way, whether that is through

Feature Group	Error rate	
	Reliable	Unreliable
NELA	0.32	0.68
FT	0.14	0.86
HCNF	0.41	0.59
N2V	0.63	0.37
NT2V	0.81	0.19

Table 4: Distribution of errors per class per feature group using a Random Forest classifier. Content based models show better performance when classifying reliable sources, whereas CSN features shows better performance when classifying unreliable sources.

Feature Group	In Time	Forecast
FT	0.61	0.55
FT+NT2V	0.75	0.72
NELA	0.63	0.59
NELA+NT2V	0.71	0.64
NELA+HCNF	0.66	0.69
FT+HCNF	0.55	0.60

Table 5: Accuracy scores for each feature group when testing In Time (first month), and when forecasting (testing on a time period outside of training). The results show that the combination of text and network (NT2V) features has better forecast potential then the text-only models. A Random Forest classifier was used.

the text space in NT2V sampling or the purely text-based features, these mistakes are corrected, as there are numerous examples of reliable sources that produce articles similar to the style of Reuters and The Guardian. In this specific case, the CSN model can be improved by having more labels in the non-U.S. communities of the network. This sparse label problem is also why we see the CSN models classifying the `unreliable` class better, as the unreliable sources are more densely clustered together than the reliable sources in the network. We leave explicit tests on the impact of removing and adding nodes/labels in the CSN to future work.

Another interesting case is when both the CSN and text features incorrectly label an article, but the combination of them flips the label. For example, some articles from Business Insider, a reliable news source, are classified in this way. In the CSN space, Business Insider falls in the U.S. mainstream community, but is a peripheral node, which may lead to very few other reliable nodes being sampled in the network embedding process. In the text space, the articles are similar to other mainstream sources in the body, but the titles can sometimes be considered 'clickbait', which is often a trait of unreliable news articles. Hence, both feature models individually may not have enough information to say it is similar to a reliable source, but together they can correctly label the article.

We also note that not all text models are alike. We found that the hand-crafted text features (NELA) and the word representation features (FT) also make dissimilar mistakes. While these mistakes are not as dissimilar as those between the text-based models and the CSN models, they are notably different, with a 33% chance that a mistake made by NELA is correctly classified by FT, and a 43% vice versa. However, these differences in mistakes do not seem to be enough to help prediction performance. When qualitatively looking at these differences in mistakes, it is hard to say what specifically causes them. However, in general, it seems they are capturing different features of the text. A clear example of these differences is in the titles of articles from unreliable sources. FT often mis-classifies articles from unreliable sources with the word 'BREAKING' in the headline or with 'clickbait' headlines. NELA correctly classifies these articles. This result makes sense, as NELA has many features which focus specifically on the title of articles, including the use of words in all capital letters, while a word embedding model cannot directly capture this.

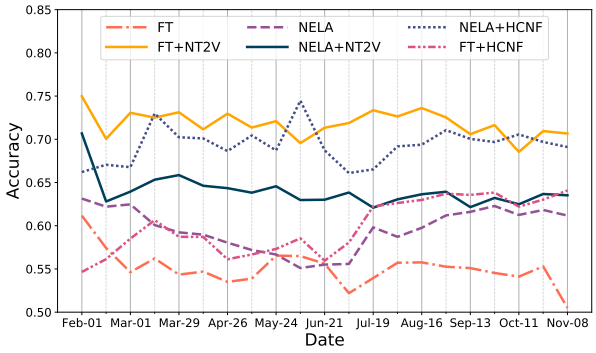


Figure 4: Classification accuracy overtime. The first month of data is used for training the classifiers, which are tested on each subsequent two week time slices. The combination of text and CSN features provide higher accuracy and stability over time, particularly with the combination of FT and NT2V features.

6.3 CSN features improve the stability of text-based models over time

In this section, we examine the stability of the performance improvements from network models over time. To test this, we train each classifier on the first month of data and test the classifier on each 2 week slice of data moving forward in time. We only test the models on sources that are unused in training and perform this train-test split over 50 runs of 20% of the sources. Again, we ensure that each source is balanced. Note, we also reconstruct the CSN network to only include information from the first month of data. This simulates a classifier that is built in February 2018 and left static for the rest of the year. These results are in Figure 4.

In addition to showing performance stability over time for each model, in Table 5, we show the classification accuracy for each feature group in two scenarios, using a Random Forest classifier: `in time` and `forecast`. The `in time` test is a prediction test on data from the same time period as

the training (i.e. February 2018), while the `forecast` test is a prediction test on the remaining time period without retraining.

As shown in Figure 4, the addition of NT2V features improves both the overall performance of the model and its consistency over time. For example, for FT, there is at most an accuracy drop of 11% over 10 months (0.61 to 0.50). However, when combined with NT2V, not only the initial accuracy is higher, but the drop is more subtle (0.75 to 0.70). However, not all NT2V combinations remain this stable. Specifically, when NT2V is combined with NELA, we similarly see a boost in overall accuracy, but see a significant initial drop in accuracy from February to March (-8%). However, the model remains very stable after the initial drop.

The results in Table 5 show that combining text and network features improve the forecast performance, but this performance increase is not always significant.

7 Discussion and Conclusion

In this study, we presented a novel feature set for the detection of articles from unreliable sources, utilizing the rich structure of news content sharing networks. To do this, we used a network embedding method that takes a deep walk approach to sample from both the CSN space and the text space. The addition of the text space to the CSN space in the sampling process makes it possible to find representations of incomplete networks by positioning sources with unknown CSN information close to those with high similarity in the text space. We show that the information provided by embedding CSN networks provides a strong signal of reliability and boosts the accuracy of text-based models. We show that text information and CSN information make dissimilar mistakes, illustrating complementary signals between the two types of models. Saliently, these CSN features also stabilize the performance of text-based models over time, performing consistently over a 10 month time frame without retraining. This stabilization is likely due to the fact that the CSN structure remains largely unchanged over time, while text features are vulnerable to recurrent topic changes.

There may be additional advantages to using the CSN embedding model that can be explored in future work. First, both the CSN construction and word embedding are language-agnostic, unlike the hand-crafted text features (NELA). Assuming reliable and unreliable media operate distinctly in other languages and cultures, the NT2V embedding can be used out-of-the-box to detect these differences. In fact, this method could be extended to many other types of information spaces beyond political news, as it is common for sources to amplify their message by creating copies (e.g. bot-generated retweets on Twitter). Second, CSN features may also work in distinguishing different granularity of labels, due to the tightly-formed communities in the network. For example, if we have labels of political-leaning or other characteristics of sources, it is possible that we can separate them in the network space.

In conclusion, using the behavior of information producers provides valuable signal in news veracity classification. This result points to a bigger picture need to explore and understand tactics used by disinformation producers not only for social interventions, but for automated support tools. If we can continue to structure information producer behaviors and tactics clearly, they can be used to aid our automated methods, which in turn can further our understanding of the news ecosystem.

References

Agerri, R. 2019. Doris martin at semeval-2019 task 4: Hyperpartisan news detection with generic semi-supervised features. In *Proceedings of the 13th SemEval*, 944–948.

Allcott, H., and Gentzkow, M. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives* 31(2):211–36.

Alvermann, D. E. 2017. Social media texts and critical inquiry in a post-factual era. *Journal of Adolescent & Adult Literacy* 61(3):335–338.

Baly, R.; Karadzhov, G.; Alexandrov, D.; Glass, J.; and Nakov, P. 2018. Predicting factuality of reporting and bias of news media sources. In *Proceedings of 2018 EMNLP*.

Baly, R.; Karadzhov, G.; Saleh, A.; Glass, J.; and Nakov, P. 2019. Multi-task ordinal regression for jointly predicting the trustworthiness and the leading political ideology of news media. *arXiv preprint arXiv:1904.00542*.

Barrón-Cedeno, A.; Jaradat, I.; Da San Martino, G.; and Nakov, P. 2019. Proppy: Organizing the news based on their propagandistic content. *Info. Proc. & Manage.*

Bastian, M.; Heymann, S.; and Jacomy, M. 2009. Gephi: An open source software for exploring and manipulating networks.

Boczkowski, P. J. 2010. *News at work: Imitation in an age of information abundance*. University of Chicago Press.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the ACL* 5:135–146.

Brady, W. J.; Wills, J. A.; Jost, J. T.; Tucker, J. A.; and Van Bavel, J. J. 2017. Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences* 114(28):7313–7318.

Braun, J. A., and Eklund, J. L. 2019. Fake news, real money: Ad tech platforms, profit-driven hoaxes, and the business of journalism. *Digital Journalism* 7(1):1–21.

Castelo, S.; Almeida, T.; Elghafari, A.; Santos, A.; Pham, K.; Nakamura, E.; and Freire, J. 2019. A topic-agnostic approach for identifying fake news pages. In *WWW Companion*, 975–980. ACM.

Castillo, C.; Mendoza, M.; and Poblete, B. 2011. Information credibility on twitter. In *Proceedings of the 20th WWW*, 675–684. ACM.

Cruz, A.; Rocha, G.; Sousa-Silva, R.; and Cardoso, H. L. 2019. Team fernando-pessa at semeval-2019 task 4: Back to basics in hyperpartisan news detection. In *Proceedings of the 13th SemEval*, 999–1003.

Färber, M.; Qurdina, A.; and Ahmedi, L. 2019. Team peter brinkmann at semeval-2019 task 4: Detecting biased news articles using convolutional neural networks. In *Proceedings of the 13th SemEval*, 1032–1036.

Graham, J.; Haidt, J.; Koleva, S.; Motyl, M.; Iyer, R.; Wojcik, S. P.; and Ditto, P. H. 2013. Moral foundations theory: The pragmatic validity of moral pluralism. In *Advances in experimental social psychology*, volume 47. Elsevier. 55–130.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864. ACM.

Guacho, G. B.; Abdali, S.; Shah, N.; and Papalexakis, E. E. 2018. Semi-supervised content-based detection of misinformation via tensor embeddings. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 322–325. IEEE.

Horne, B. D.; Dron, W.; Khedr, S.; and Adalı, S. 2018. Assessing the news landscape: A multi-module toolkit for evaluating the credibility of news. In *WWW Companion*.

Horne, B. D.; Gruppi, M.; and Adalı, S. 2020. Do all good actors look the same? exploring news veracity detection across the us and the uk. *ICWSM ICWSM Data Challenge on Safety*.

Horne, B. D.; Nørregaard, J.; and Adalı, S. 2019a. Different spirals of sameness: A study of content sharing in mainstream and alternative media. In *Proceedings of ICWSM*, volume 13, 257–266.

Horne, B. D.; Nørregaard, J.; and Adali, S. 2019b. Robust fake news detection over time and attack. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11(1):1–23.

Hosseinimotlagh, S., and Papalexakis, E. E. 2018. Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. *MIS2*.

Hutto, C. J., and Gilbert, E. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth ICWSM*.

Kumar, S., and Shah, N. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*.

Lazer, D. M.; Baum, M. A.; Benkler, Y.; Berinsky, A. J.; Greenhill, K. M.; Menczer, F.; Metzger, M. J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; et al. 2018. The science of fake news. *Science* 359(6380):1094–1096.

Lewandowsky, S.; Ecker, U. K.; Seifert, C. M.; Schwarz, N.; and Cook, J. 2012. Misinformation and its correction: Continued influence and successful debiasing. *Psychological Science in the Public Interest* 13(3).

Li, C., and Goldwasser, D. 2019. Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *Proceedings of ACL*, 2594–2604.

Moreno, J. G.; Pitarch, Y.; Pinel-Sauvagnat, K.; and Hubert, G. 2019. Rouletabille at semeval-2019 task 4: Neural network baseline for identification of hyperpartisan publishers. In *Proceedings of the 13th SemEval*, 981–984.

Newman, M. E., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E* 69(2):026113.

Nørregaard, J.; Horne, B. D.; and Adalı, S. 2019. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings ofICWSM*, volume 13, 630–638.

Pennycook, G.; McPhetres, J.; Zhang, Y.; Lu, J. G.; and Rand, D. G. 2020. Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological science* 31(7):770–780.

Popat, K.; Mukherjee, S.; Strötgen, J.; and Weikum, G. 2016. Credibility assessment of textual claims on the web. In *Proceedings of ACM CIKM*, 2173–2178. ACM.

Potthast, M.; Kiesel, J.; Reinartz, K.; Bevendorff, J.; and Stein, B. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*.

Recasens, M.; Danescu-Niculescu-Mizil, C.; and Jurafsky, D. 2013. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the ACL (Volume 1: Long Papers)*, volume 1, 1650–1659.

Rozemberczki, B.; Allen, C.; and Sarkar, R. 2019. Multi-scale attributed node embedding. *arXiv preprint arXiv:1909.13021*.

Ruchansky, N.; Seo, S.; and Liu, Y. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 797–806. ACM.

Singh, L.; Bansal, S.; Bode, L.; Budak, C.; Chi, G.; Kawintiranon, K.; Padden, C.; Vanarsdall, R.; Vraga, E.; and Wang, Y. 2020. A first look at covid-19 information and misinformation sharing on twitter. *arXiv preprint arXiv:2003.13907*.

Singhania, S.; Fernandez, N.; and Rao, S. 2017. 3han: A deep neural network for fake news detection. In *NeuIPS*, 572–581. Springer.

Speed, E., and Mannion, R. 2017. The rise of post-truth populism in pluralist liberal democracies: challenges for health policy. *International journal of health policy and management* 6(5):249.

Starbird, K.; Arif, A.; Wilson, T.; Van Koevering, K.; Yefimova, K.; and Scarnecchia, D. 2018. Ecosystem or echosystem? exploring content sharing across alternative media domains.

Starbird, K. 2017. Examining the alternative media ecosystem through the production of alternative narratives of mass shooting events on twitter. In *ICWSM*, 230–239.

Tausczik, Y. R., and Pennebaker, J. W. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology* 29(1):24–54.

Yang, F.; Liu, Y.; Yu, X.; and Yang, M. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, 13.

Yang, C.; Liu, Z.; Zhao, D.; Sun, M.; and Chang, E. 2015. Network representation learning with rich text information. In *Twenty-Fourth IJCAI*.

Ye, J., and Skiena, S. 2019. Mediarank: Computational ranking of online news sources. *arXiv preprint arXiv:1903.07581*.