Context Map Analysis of Fake News in Social Media:
A Contextualized Visualization Approach

Onur Şeref
Virginia Polytechnic Institute
and State University
seref@vt.edu

Michelle M. H. Şeref
Virginia Polytechnic Institute
and State University
mmhseref@vt.edu

Sukhwa Hong
University of Hawai'i at Hilo
sukhwa@hawaii.edu

Abstract

Visualization tools in text analytics are typically based on content analysis, using n-gram frequencies or topic models which output commonly used words, phrases, or topics in a text corpus. However, the interpretation of these visual output and summary labels can be incomplete or misleading when words or phrases are taken out of context. We use a novel Context Map approach to create a connected network of n-grams by considering the frequency in which they are used together in the same context. We combine network optimization techniques with embedded representation models to generate an visualization interface with clearer and more accurate interpretation potential. In this paper, we apply our Context Map method to analyze fake news in social media. We compare news article veracity (true versus false news) with orientation (left, mainstream, or right). Our approach provides a rich context analysis of the language used in true versus fake news.

1. Introduction

Traditional text analytics depends mostly on the *content* in written text, where bag-of-words approaches are used to calculate prevalence or co-occurrence of words in documents. However, such approaches ignore the *context* in which words are used. Generative methods extend bag-of-words approaches to topic models such as latent Dirichlet allocation (LDA), which provide hidden collections (topics) of terms that tend to occur together. However, these collections require further inspection in order to understand the topics and extract meaningful information. Visualization methods are limited and are usually hard to interpret. LDA results can be represented as separate word clouds with emphasis on frequent words, however they do not go beyond a summarized display of content frequency. More temporary approaches like distributed representations of words take context into consideration; however, the resulting 2D representation of words as points on a plane are not easy to interpret either.

Text mining and visualization are often used in the analysis of user-generated content. Netzer et al.[1] create a graph to represent market structure based on co-occurrence and lift scores between brand-related terms in written text. They use a multi-dimensional scaling (MDS) method to project their spring-embedded graph onto 2D to visualize clusters of brands with applications to car models and drug brands. France and Ghose [2] follow a similar approach to find an optimal submarket partition for brand substitution and visualize it using MDS. Li et al. [3] combine a knowledge base with semantic, news-related and transactional features to create a graph that connects these features with news articles. Word clouds are used frequently for visualization of prevalent terms [4], especially in conjunction with LDA-based approaches [5, 6, 7]. Among the most recent developments, Timoshenko and Hausera [8] use state-of-the-art deep learning techniques to extract informative sentences from a small set of manually labeled ones . They use clustering techniques on sentence embeddings that are averaged from word embeddings, and finally plot them on a 2D plane using principal component analysis. More general literature on information visualization and visual analytics can be in computer graphics literature [9, 10].

Our Context Map approach is unique in both the formation of a connected n-gram network as well as the transformation of network component analysis into an interactive visualization tool for understanding context. In this paper, we apply our method to analyze fake news in social media. We compare the context clusters used in true news versus false news across media sources that are left-wing, mainstream, or right-wing oriented.

Several recent developments have been made in information systems research analyzing fake news and disinformation. Some studies look at the social network of online information sharing. Zhang et al. [11] create FakeDetector to build a deep diffusive network model to study the representations of news articles, analyzing

HᶠCSS

authors and subjects simultaneously. They study tweets posted by PolitiFact to better understand how fake news is distributed. Wu and Lie [12] propose a TraceMiner tool to study social network structures and propagation pathways of disinformation. Ruchansky et al. [13] use a recurrent neural network to study temporal engagements with news articles and user reactions. Other studies have examined the content of fake news posts. Shu et al. [14] include auxiliary information of users social behavior with fake news posts, likes, and shares.

While these papers compare different methods for the classification of fake news, none describe nor discuss the textual features of their classifiers nor provide any visual output for interpretation. Context Maps provide potential for a more in-depth analysis of fake news by providing clear contextual components for news article veracity and orientation.

2. Context Maps

Our visualization framework is composed of a number of novel graph-based methods combined with distributed representations that create a highly interpretable map, unique in its context driven composition. The components involved in our framework are summarized in Figure 1 in two parallel pipelines: the graph-based methods comprise the pipeline on the left and the distributed representation components comprise the pipeline on the right.

2.1. Preprocessing

The input for our framework is a *corpus* of documents that share a common subject. We consider that the documents in the corpus are partitioned into a *positive* and a *negative* class. Documents are composed of sentences. We define a sentence to be an ordered set of words as the fundamental unit of meaning. A preprocessing step splits documents into sentences, which are then tokenized into lemmatized and indexed words. We connect documents, sentences and words based on a linear positioning of the tokens from the beginning to the end of the corpus.

2.2. Graph-based Pipeline

An n-gram is a set of n consecutive words in a corpus. We create a connected n-gram network where each node represents an n-gram and each arc represents an $n + 1$-gram. There are therefore two connected n-grams in an $n + 1$-gram. We call the former n-gram the *tail* node and the latter the *head* node, where the tail node and head node have $n - 1$ overlapping words. For example, in the 4-gram {*president sign executive*

Figure 1. Computational framework for context maps.

order} the tail 3-gram is {*president sign executive*} and the head 3-gram is {*sign executive order*}. We represent the corpus using a set N of unique n-grams. The corpus is partitioned into positive and negative document classes. Let f_i^+ and f_i^- denote the cumulative frequencies of n-gram $i = 1, \ldots, |N|$ in the positive and negative documents, respectively. We eliminate n-grams such that $f_i^+ + f_i^- < th_f$, where th_f is a frequency threshold (typically $5 \leq th_f \leq 10$). Lower values of th_f might include arbitrary terms, increase

noise, and have a prohibitively high computational cost, whereas higher values of it might exclude important terms and produce sparse results. For the remaining n-grams, we calculate the *log-score* for each n-gram as $s_i = \log f_i^+ - \log f_i^-$. We further eliminate n-grams such that $|s_i| < th_c$, where th_c is a contrast threshold (typically $0.5 \leq th_c \leq 1.5$) to create a contrast between n-grams that occur more frequently in one class compared to the other. The remaining n-grams can be grouped into a positive set N^+ and a negative set N^-. Note that $N^+ \cap N^- = \emptyset$. We consider N^+ as a set of nodes that represent their respective n-grams, and we define a set of arcs A^+ to represent $n+1$-grams in the corpus whose tail and head n-grams belong to N^+. We call the combined set of nodes and arcs $G^+ = (N^+, A^+)$ as the *positive n-graph*; we construct a *negative n-graph* G^- similarly. The positive and negative n-graphs are typically composed of a significantly large component and many smaller components. The largest components of G^+ and G^- are shown in Figure 2 using a force-directed layout.
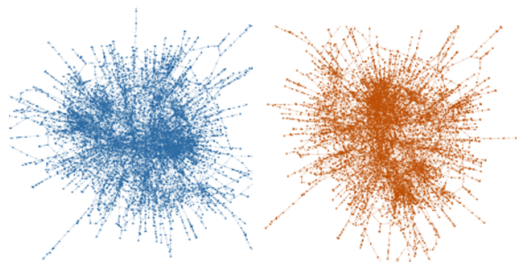

Figure 2. Largest components of the n-Gram Network for Positive (Blue) and Negative (Orange) Document Classes.

Both n-graphs are sparse graphs representing frequent connections between n-grams that trace variations of expressions within different contexts relative to their locality in the graph. Without loss of generality, we partition G^+ into smaller subsets of a maximum size t_{max} (typically $10 \leq t_{max} \leq 20$) to capture the local context as follows. Ignoring directionality of arcs in A^+, we compute a maximal spanning graph using Kruskal's algorithm, using the frequencies of the $n+1$-grams (arcs) as arc weights, which results in a forest of trees $F = \{T_1, \ldots, T_{|F|}\}$. We traverse each tree using breadth-first-search starting with the root node that has the highest degree to determine a parent node for each node, except for the root node. Since each path between a node and the root in a tree is unique, we call the set of nodes D_i the *descendants* of node i as the set of nodes whose paths to the root node pass through node i. For each node we

calculate a splitting score $ss_i = |D_i|(|T| - |D_i| - 1) :$ $i \in T$. We select a node $i^* = \arg\max\{ss_i\}$ to remove the arc between this node and its parent to split its respective tree T^* into two smaller trees $T\prime$ and T'' with relatively balanced sizes. The split trees replace the original tree T^* in the forest F. We repeat this procedure until $|T| \leq t_{max}$ for all $T \in F$ (typically $5 \leq t_{max} \leq 20$). We call the resulting trees in the forest *context components* or *CoCo*'s. Smaller values of k_{max} might include fragments of a particular context whereas larger values might start combining different but related context. Examples of CoCos can be seen in Figure 3, where $t_{max} = 20$. We denote the positive and negative forests as F^+ and F^-, respectively. There are typically several hundred to several thousand context components in the positive or negative forests combined, depending on the corpus size and the threshold levels.
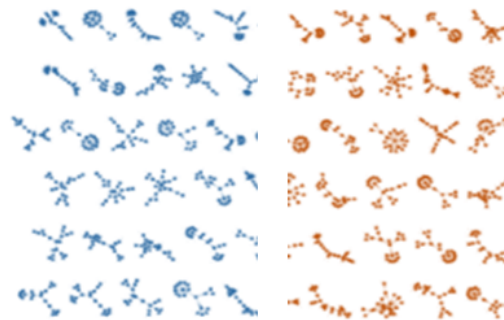

Figure 3. Network Decomposition into CoCo sets for Positive (Blue) and Negative (Orange) Document Classes ($t_{max} = 20$).

2.3. Distributed Representation Pipeline

Distributed representation of a word or sentence in an embedded vector space has become state-of-the-art in natural language processing. One of the most commonly used methods, which is available in multiple platforms, is *word2vec* [15], which creates vector representations of words from a corpus such that words with similar contextual relationship have similar orientations. For example, the vector difference between the words *king* and *queen* would be very similar to the vector difference between the words *man* and *woman*. These embedded spaces are trained using a shallow neural network with two layers. *Doc2Vec* [16] is an extension of *word2vec* that introduces a token for each document, which is trained along with the the words from the document, and creates an embedded representation for the document, similar to a word vector.

We apply doc2vec to the sentences that were split

during the preprocessing stage and create a vector representation for each sentence. The dimensionality of the embedded space is typically selected between 100 and 300 depending on the corpus size and preferred amount of training time. There are some other parameters such as number of epochs that can be adjusted accordingly in training the neural network. Sentences with similar meanings have similar vectors in the embedded space, which is an important property that lacks in high-dimensional sparse representations like one-hot vectors.

We use a representative subset of these sentence vectors to maintain our approach within a reasonable computational time window. As a baseline approach, a random sample of sentence vectors regardless of context components can be used, however this approach produces inconsistent results. Given a particular context component T, we identify the set of sentences in the corpus that contain one or more n-grams from the context component and denote the set of vectors for these sentences as V_T. Then we select the sentence that contains the maximum number of n-grams from the context component as the *representative sentence* for that context component and its sentence vector as the *representative vector*. This selection can also be done in alternative ways; for example, the most central vector in V_T can be identified as the representative vector for context component T. In our study, we used the former method as it produced more informative sentences for interactive visualization.

The set of representative vectors becomes the input for the t-distributed stochastic neighbor embedding (t-SNE) algorithm [17], which is the state-of-the-art nonlinear dimensionality reduction algorithm used for visualizing high-dimensional vector representations in two or three dimensions. The t-SNE algorithm models each high-dimensional vector with a low dimensional point in such a way that the probabilistic similarities and dissimilarities in the high-dimensional input vector space are preserved in the low-dimensional output space. We create a 2D point for each representative vector using t-SNE to represent its respective context component. We call the 2D space that contains all points the *context space*. We note that although the representative vectors come from positive and negative context components, t-SNE does not use the class label information in producing the 2D points. In addition, the positions of points relative to each other do not change with arbitrary rotations; therefore, the context space is rotation invariant and the horizontal and vertical axes have no particular significance.

2.4.   3D Grid Partitioning

After the 2D points are generated via t-SNE, we use their class label of their respective context components to generate a cumulative positive or negative impact score around their position in the context space. Let the 2D points be represented as sets X^+ and X^- depending on their class label and $X = X^+ \cup X^-$. To measure their impact on the context space, we generate a set of points Y in a grid layout that spans between the minimum and maximum coordinates of points in X. We set the distance d between the neighboring grid points such that the cardinality of Y is similar to the magnitude of order as that of X. The cumulative impact of points $x \in X$ on a grid point $y \in Y$ is calculated using a Gaussian kernel as

$$z(y) = \sum_{x \in X^+} e^{\frac{||y-x||^2}{\gamma}} - c_b \sum_{x \in X^-} e^{\frac{||y-x||^2}{\gamma}},$$

where kernel bandwidth γ serves as a smoothing parameter and $c_b = |X^+|/|X^-|$ is a balancing coefficient to account for the difference between the number of positive and negative context components. We call this impact score the *elevation* of a grid point. The elevation of neighboring grid points creates regions with hills and valleys in the context space similar to a geological map.

We use a novel and efficient graph-based method to determine the boundaries of elevated and depressed regions. We partition the grid points Y into two sets $Y^+ = \{y \in Y : z(y) \geq 0\}$ and $Y^- = \{y \in Y : z(y) < 0\}$. We consider a graph where each point in $y \in Y^+$ is represented by a node. For each point $y \in Y^+$, we identify the point in $y' \in Y$ within a radius of $r = kd$ of point y (typically $1 \leq k \leq 10$) with the maximum elevation and set the node of y' as the parent of the node for y to create an edge between y and y'. If a point has the maximum elevation within a radius r, then it has no parent. This graph defines a forest whose trees can simply be identified by a simple traversal algorithm. Then, for the points that are represented by the nodes of each tree in the forest, we create a simple polygon using an α-shape algorithm [18], which determines a closed positive region in the context space. Similarly, we repeat this procedure with the points in $y \in Y^-$ using the minimum elevation point within a radius of a point as its parent. The resulting regions are the negative regions in the context space.

All points in X that fall inside a region can be determined using a point-in-polygon algorithm [19] and their representative sentences can be displayed along with the boundary of the region interactively.

The representative sentences are highly interpretable in identifying the general context of a region. We consider sentences that belong to a region to create few automatic keywords with high term frequency-inverse document frequency (tf-idf) scores to be displayed at the highest (lowest) points of the positive (negative) regions. We call this interactive and interpretable map of the context space a *context map*.

2.5. Regional Polarity Analysis

The distribution of the positive and negative n-grams found in a context region R over a subset S of documents might lead to interesting insights. Subset S might be independent of the positive and negative classes in context region R, or it might show polarity towards a particular class.

Let all the n-grams in a corpus that belong to the positive and negative forests of context components be denoted as N_{F+} and N_{F-}, respectively. Given a subset S of documents, the probability of a random n-gram i found in S that belongs to a context component tree in region $R \subset (F^+ \cup F^-)$ to also belong to N_{F+} is calculated as

$$P(i \in N_{F+}|R,S) = \frac{f^+(R,S)}{f^+(R,S) + c_b f^-(R,S)},$$

where $f^+(R,S)$ and $f^-(R,S)$ are the frequencies of positive and negative context components in S that belong to R, respectively, and c_b is the same balancing coefficient used in 3-D grid partitioning. The probability for the n-gram in S from R to belong to N_{F-} is clearly

$$P(i \in N_{F-}|R,S) = 1 - P(i \in N_{F+}|R,S).$$

These probabilities can be considered as the prevalence of positive and negative context components in region R in relation to the documents in S, which allows an analysis of a set of documents in relation to the polarity of the context components within a specific region R.

3. Context Map Analysis of Fakes News

We apply our Context Map method to analyze fake news in social media. The BuzzFeed-Webis Fake News Corpus (2016) comprises the output of nine publishers in a week close to the 2016 US elections. Every post and linked article were collected from these publishers for a total of seven days, from September 19 to 23, and September 26 and 27, 2016. All posts are fact-checked by professional journalists at BuzzFeed. Potthast, et.al. (2018) describes the creation of the dataset. The dataset contains 1627 articles. We removed 81 duplicate articles from this data set, resulting in a total of 1546 documents.

Out of the nine publishers, six are hyper-partisan with three left-wing and three right-wing publishers, and the remaining three publishers are mainstream. We refer to this as the *orientation* of the news article: left, mainstream, or right. The articles have been labeled by journalists at BuzzFeed using the scale: no factual content, mostly false, mixture of true and false, and mostly true. We refer to this as the *veracity* of the news article. We consider the *positive* class to be mostly true articles and the *negative* class to be all other articles with any degree of false news label. Table 1 shows the distribution of articles in each veracity class for each news source orientation for the entire data set.

	Positive	Negative	SUM
Left	11.51%	4.79%	16.30%
Mainstream	49.09%	1.23%	50.32%
Right	16.95%	16.43%	33.38%
SUM	77.55%	22.45%	100%

Table 1. Veracity Percentage per Orientation

	CoCo's		Regions	
Positive	3295	(85.50%)	37	(51.39%)
Negative	559	(14.50%)	35	(48.61%)
SUM:	3854		72	

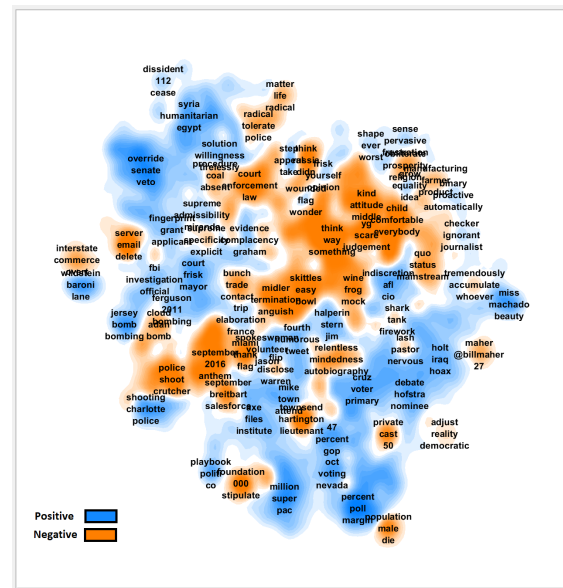**Table 2. Number of CoCos and Regions per Class**

Figure 4. Context Map: Positive (Blue) Regions represent True News and Negative (Orange) Regions represent Fake News.

Our Context Map analysis of this dataset creates a set of 3295 positive and 559 negative CoCo's ($c_b = 5.89$),
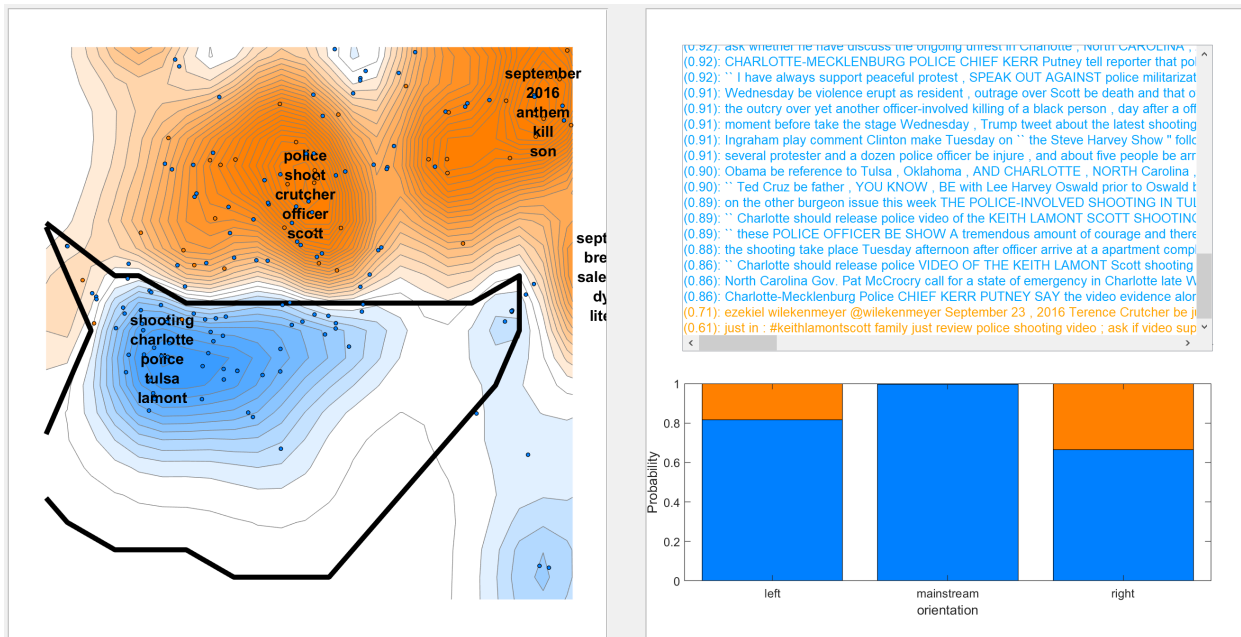
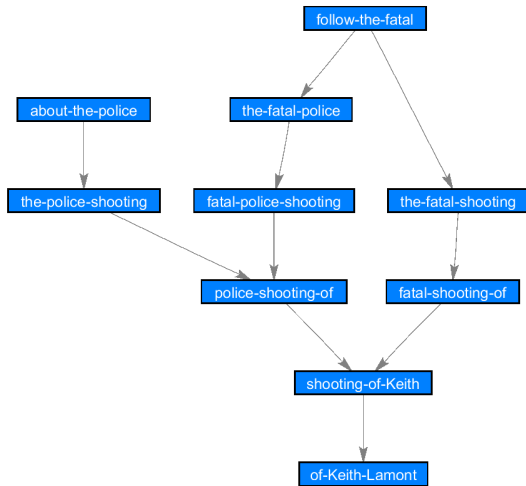**Figure 5.** Orientation Plot for Selected Positive Region: "Shooting/Charlotte/Police/Tulsa/Lamont".

Figure 6. Example CoCo from Selected Positive Region: "Shooting/Charlotte/Police/Tulsa/Lamont".

and a map of 37 positive and 35 negative regions (see Table 2). The context map output (shown in Figure 4) shows the positive (blue) regions for context clusters in which news articles are mostly true and negative (orange) regions for context clusters in which news articles are either mostly false, mix of false and true, or have no factual content. Darker colors show higher elevations (blue) and deeper depressions (orange). Each region represents a cluster of CoCo's based on their shared context frequency. Each region label is generated

from terms with the highest tf-idf scores in the clustered data. Using our interactive visualization output, a selected region can be further analyzed by examining the representative CoCo sentences for both positive and negative instances in the region.

For example, in Figure 5, we have selected the positive region labeled "shooting/charlotte/police/tulsa/lamont" and zoomed in on the selected region. The right side of the visualization output shows the representative sentences for each CoCo located in this context region. For example, some positive CoCo expressions for this region include: "charlotte should release police video", "have spark protest in charlotte", "tulsa officer who fatally shoot", "fatal police shooting of keith lamont". These CoCo's provide the context in which the region labels indicate mostly true news. That is, the word "police" by itself is not necessarily indicative of true news; however, "police" used in the context of these CoCo expressions is.

In Figure 6 we see the n-gram (node) and $n+1$-gram (arc) network for two positive CoCo's from the selected region. We define CoCo phrases or expressions as any path in the CoCo network. For example, expressions from the CoCo shown in the figure may include "fatal police shooting of keith lamont" or "about the police shooting" or "follow the fatal shooting of".

From the context map, we observe that adjacent regions may have some similar labels despite different region classification. For example,
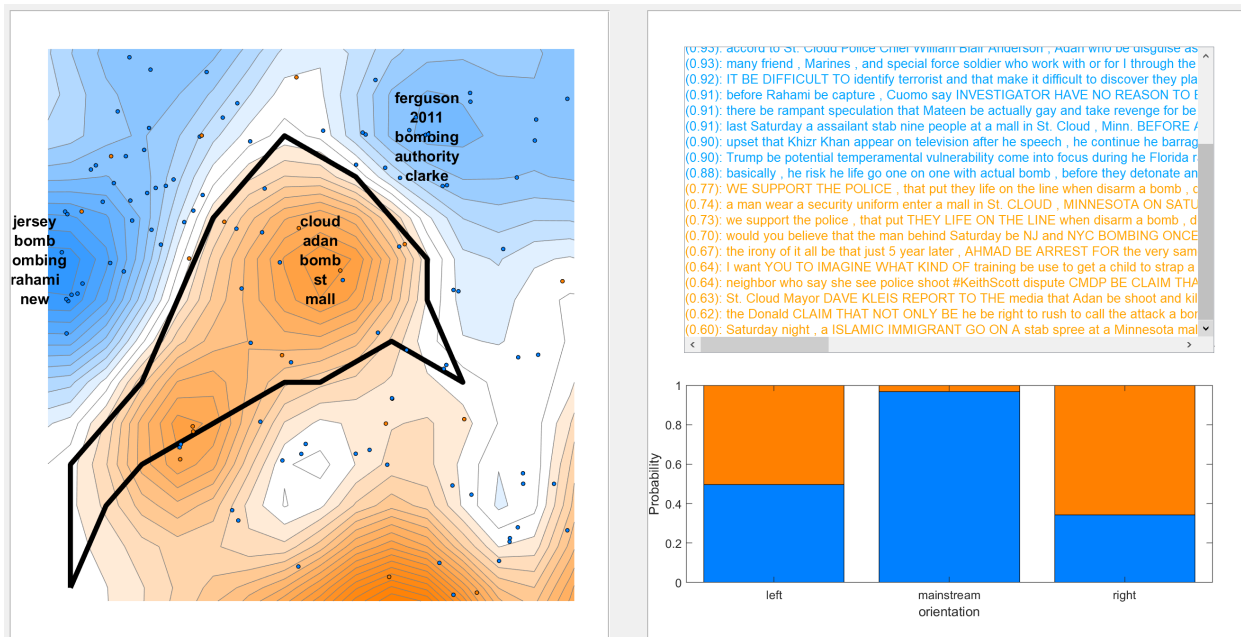
Figure 7. Orientation Plot for Selected Negative Region: "Cloud/Adan/Bomb/St/Mall" and Orientation Plot.

in Figure 5, adjacent to the positive region labeled "shooting/charlotte/police/tulsa/lamont" we see a negative region labeled "police/shoot/crutcher/officer/scott". Even though "police" and "shoot"/"shooting" appear in both labels, the region classes depend on the context in which these words are used. For example, in this adjacent negative region there are negative CoCo expressions such as: "they claim keith scott be", "indeed unarmed but", "crutcher not shoot with his hand up".

Comparing adjacent regions from different classes provides potential for further analysis in political science, sociology, rhetoric, or social psychology. For example, the negative (false news) CoCo's use hedging phrases such as "claim" or "unarmed but" to incite doubt regarding the shooting victim Keith Scott and defensive language in support of officer Crutcher.

In another example (shown in Figure 7) the selected region is negative with labels "cloud/adan/bomb/st/mall". The example sentences show several negative CoCo's as well as a few positive CoCo instances. For example, a rhetorical analysis may compare negative CoCo phrases such as "claim that not only be" and "be claim that he" in the false news articles versus "investigator have no reason" and "have find no evidence that" in the true news reports, where fake news uses hedging and doubt compared to the declarative statements used in true news articles. In addition, a politial science analysis may compare phrases such as "islamic immigrant" used in fake news

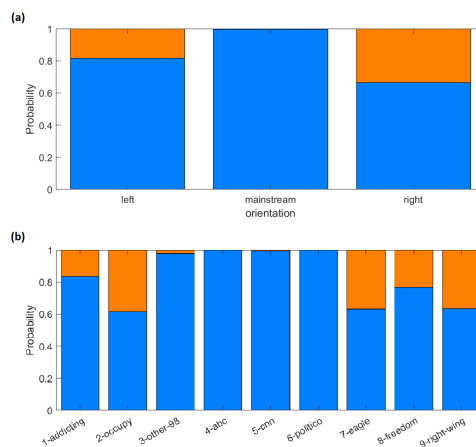versus "us citizen terrorism suspect" used in true news.

Figure 8. Selected Region Plots for (a) Orientation or (b) News Source.

In the bottom-right side of the visualization output we have a plot showing the relative amount of positive and negative CoCo's within this region's context for each news orientation (left, mainstream, right). Referring back to Figure 5 for the positive region labeled "shooting/charlotte/police/tulsa/lamont", we see that approximately 80% of left news articles use the positive CoCo's in this region to report true news as well as 100% of mainstream news articles found in this region. However, only 65% of the right-wing news articles use

the positive CoCo's from this region to report true news. In Figure 7, the selected negative region plots show that approximately 50% of left news articles use the negative CoCo's in this region when reporting fake news, 5% of mainstream news articles use these negative CoCo's, and 70% of right-wing news articles. We can also modify this plot to show the actual news sources within each orientation group. For example, in Figure 8, we see the variation of positive CoCo's used in left-wing news sources: "addicting info", "occupy democrats", and "the other 98"; mainstream news sources: "abc", "cnn", "politico"; right-wing news sources: "eagle rising", "freedom daily", and "right-wing news".

These examples illustrate the potential interpretation analysis of the Context Map visualization output. Compared to LDA topic analysis output (shown in Table 3), we believe that Context Maps provide a richer landscape in which topic classifications can be compared regionally or analyzed at a more detailed sentence level. The LDA output has a list of words taken out of context, with overlapping terms in different topic groups, making it challenging to describe characteristics of the fake news classes. Our network-based approach creates an output tool which lends itself to both macro- and micro-level descriptive analysis of the language used in news veracity and orientation.

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
trump	say	clinton	say	new
clinton	people	say	police	trump
say	country	trump	black	clinton
debate	go	state	officer	million
donald	make	campaign	people	percent
hillary	like	president	man	voter
campaign	good	former	law	poll
think	get	obama	take	2016
presidential	america	house	right	vote
go	know	bill	get	state

Table 3. Top 10 words in 5 LDA topics

4. Conclusions and Future Research

Our novel Context Map method uses network-based context clustering to provide interactive visualization output with the potential for rich analysis. Compared to word clouds or topic models, our visualization output presents multiple levels of detail for understanding context regions using labels generated by the CoCo analysis. We feel that both our approach and its information extraction capabilities offer contributions to the fields of information and computer science as well as the social sciences. Further research using Context Maps includes studying the context of social movement trends or consumer reviews. We also plan to extend our research beyond descriptive analytics to use our CoCo

networks in classification.

Our computational framework allows several modifications. In addition to t-SNE, we plan to study the effect of traditional dimensionality reduction methods like principal component analysis and multi-dimensional scaling as well as contemporary ones like uniform manifold approximation projection [20]. Although the Gaussian kernel with a single bandwidth parameter works well in the current application area, other similarity measures or kernel functions can easily be adopted to create the context regions. Our framework can also be extended to handle multiple classes by creating an n-Graph and a set of CoCos for each class. The context space can be partitioned into regions composed of multiple classes and a multi-class polarity analysis can be performed for each region. Moreover, different prevalence scores can be used on words in representative sentences to study the specificity of the context regions with respect to their relative location on the map.

References

[1] O. Netzer, R. Feldman, J. Goldenberg, and M. Fresko, "Mine your own business: Market-structure surveillance through text mining," *Marketing Science*, vol. 31, no. 3, p. 521543, 2012.

[2] S. L. France and S. Ghose, "An analysis and visualization methodology for identifying and testing market structure," *Marketing Science*, vol. 35, no. 1, p. 182197, 2016.

[3] X. Li, K. Chen, S. X. Sun, T. Fung, H. Wang, and D. D. Zeng, "A commonsense knowledge-enabled textual analysis approach for financial market surveillance," *INFORMS Journal on Computing*, vol. 28, no. 2, pp. 278–294, 2016.

[4] V. L. Miguis and H. Nvoa, "Exploring online travel reviews using data analytics: An exploratory study," *Service Science*, vol. 9, no. 4, p. 315323, 2017.

[5] S. Mankad, H. Han, J. Goh, and S. Gavirneni, "Understanding online hotel reviews through automated text analysis," *Service Science*, vol. 8, no. 2, pp. 124–138, 2016.

[6] J. Liu and O. Toubia, "A semantic approach for estimating consumer content preferences from online search queries," *Marketing Science*, vol. 37, no. 6, p. 930952, 2018.

[7] C. Lim and P. P. Maglio, "Data-driven understanding of smart service systems through text mining," *Service Science*, vol. 10, no. 2, pp. 154–180, 2018.

[8] A. Timoshenko and J. R. Hausera, "Identifying customer needs from user-generated content," *Marketing Science*, vol. 36, no. 1, pp. 1–20, 2019.

[9] M. Alharbi and R. S. Laramee, "SoS TextVis: A survey of surveys on text visualization," in *Computer Graphics and Visual Computing (CGVC)* (G. K. L. Tam and F. Vidal, eds.), The Eurographics Association, 2018.

[10] K. Kucher, C. Paradis, and A. Kerren, "The state of the art in sentiment visualization," *Computer Graphics Forum*, vol. 37, no. 1, pp. 71–96, 2018.

[11] J. Zhang, L. Cui, Y. Fu, and F. B. Gouza, "Fake news detection with deep diffusive network model," *CoRR*, vol. abs/1805.08751, 2018.

[12] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, (New York, NY, USA), pp. 637–645, ACM, 2018.

[13] N. Ruchansky, S. Seo, and Y. Liu, "CSI: A hybrid deep model for fake news," *CoRR*, vol. abs/1703.06959, 2017.

[14] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *CoRR*, vol. abs/1708.01967, 2017.

[15] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, (USA), pp. 3111–3119, Curran Associates Inc., 2013.

[16] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196, JMLR.org, 2014.

[17] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[18] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.

[19] K. Hormann and A. Agathos, "The point in polygon problem for arbitrary polygons," *Computational Geometry*, vol. 20, no. 3, pp. 131–144, 2001.

[20] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.