Searching for Diverse Perspectives in News Articles: Using an LSTM Network to Classify Sentiment

Christopher Harris
Department of Computer Science
University of Northern Colorado
Greeley, CO 80639

ABSTRACT

When searching for emerging news on named entities, many users wish to find articles containing a variety of perspectives. Advances in sentiment analysis, particularly by tools that use Recurrent Neural Networks (RNNs), have made impressive gains in their accuracy handling NLP tasks such as sentiment analysis. Here we describe and implement a special type of RNN called a Long Short Term Memory (LSTM) network to detect and classify sentiment in a collection of news articles. Using an interactive query interface created expressly for this purpose, we conduct an empirical study in which we ask users to classify sentiment on named entities in articles and then we compare these sentiment classifications with those obtained from our LSTM network. We compare this sentiment in articles that mention the named entity in a collection of news articles. Last, we discuss how this analysis can identify outliers and help detect fake news articles.

Author Keywords

Sentiment analysis; RNN; LSTM; named entities; artificial neural networks; news analysis; fake news.

ACM Classification Keywords

I.5.1 [Pattern Recognition]: Models → Neural nets; I.2.7 [Artificial Intelligence] → Natural Language Processing. H.3.3 [Information systems] → Information retrieval diversity

INTRODUCTION

Named entities, which we define as information units such as person, organization and location names, are extremely popular components of user queries. For example, Yin and Shah found that nearly 30% of searches on the Bing search engine were simply a named entity and 71% of searches contained a named entity as part of the query string [13]. Thus, the proper identification and handling of named entities is essential to provide an excellent search experience.

There has been a growing number of voices who claim bias in reporting from media sources, particularly (but not limited to) named entities in politics and entertainment. News articles covering the same named entity can be reported from a variety of perspectives, some sympathetic to the subject while others are far less so – a phenomenon widely noted during two 2016 events: the U.K. Brexit vote and U.S. elections. However, there are ways to evaluate and categorize this variation in reporting. Sentiment analysis, which has been widely applied to classifying movie and product reviews, could also be applied to the sentiment used in reporting news articles, particularly those that focus on a specific named entity. Although early approaches in sentiment analysis suffered from poor accuracy, recent advances – particularly applying deep learning techniques such as Recurrent Neural Networks (RNNs) – have increased its accuracy and can even distinguish the sentiment between different named entities when an article contains references to more than one entity.

It is important for search systems to work with named entities containing both informal text (i.e., blog posts) and formal text (i.e., news articles). To this end, it is also important to distinguish these different types of sources to the user. When information on a named entity appears from a verified news source, it carries a different weight (in terms of authenticity) from a blog posting from a non-expert; the user should be made aware of this provenance in the search results and be able to filter the search results based on the verifiability of the news.

With the rise in social media as a user's primary news source [9], misleading news articles called *fake news* have clouded many users' ability to determine if a news article has merit or if it is a deliberate attempt to misinform and spread a hoax. Recently, more attention from the NLP community has been placed on identifying fake news, which we define as propaganda disguised as real news that is created to mislead readers and damage a person's, an agency's, or an entity's reputation.

A study conducted following the 2016 election found 64% of adults indicated that fake news articles caused a great deal of confusion and 23% said they had shared fabricated articles themselves – sometimes by mistake and sometimes intentionally [3]. We believe that sentiment analysis, when done properly, can be used to separate news from genuine news sources from fake news. We explore this concept briefly in this paper.

BACKGROUND AND MOTIVATION

Performing queries and obtaining news articles are tasks that rank only behind sending email as the most common internet activities, with 91% and 76% of users reportedly engaging in these activities, respectively [10]. Overall, the internet has grown in importance as a source of information and news on

ESIDA '18, March 11, Tokyo, Japan.

named entities. As of August 2017, 43% of Americans report often obtaining their news online, quickly approaching the 50% who often obtain news by television. This 7% gap has narrowed considerably from the 19% gap between the two sources found only 18 months earlier [6].

The Role of Social Media

Social media platforms such as Facebook and Twitter have transformed how news is created and disseminated. News content on any named entity can be spread among users without substantial third-party filtering, fact-checking, or editorial judgment on this information. It is now possible for a non-expert user with no prior reputation on a news topic to reach as many readers as the verified sources such as the Washington Post, CNN, or the BBC [1].

With social media, unsurprisingly, users tend to communicate with others having a similar political ideology, affecting the ability for them to gain a balanced perspective. Of the Facebook articles involving national news, politics, or world affairs, only 24% of liberals and 35% of conservatives have exposure to other perspectives through shares on social media [2]. Therefore, most social media users who wish to gain a different perspective on a named entity require a convenient yet customizable interface to search these articles and view information on these different perspectives. Although websites like Allsides[1] use a bias rating system to illustrate the spectrum of reporting on a liberal-conservative bias, to our knowledge, no search interface has been created to classify news articles based on the sentiment used in the text.

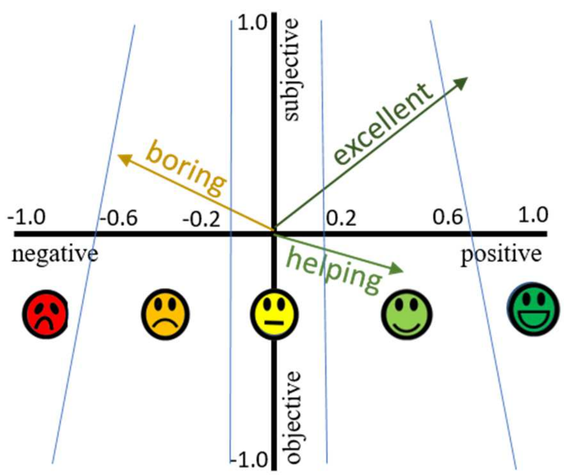

Figure 1: An example illustrating the vector representation of terms in the phrase "She was excellent at helping others but found the task boring" illustrating the polarity along the x-axis and subjectivity along the y-axis. Magnitude is represented as the length of the vector. Vertical blue lines represent the boundaries between sentiment classes, with a tighter range for terms labeled subjective as compared with those labeled as objective.

Sentiment Analysis

News articles shared on social media are often used to incite affective behavior in readers [7] and are ideal for sentiment classification. Sentiment analysis is an area of Natural Language Processing (NLP) that examines and classifies the affective states and subjective information about a topic or entity. The research question we wish to examine is how well machine classified sentiment analysis is correlated with the sentiment as determined by users (which we set as our ground truth). We do this by looking at the subjectivity/objectivity, the polarity, and the magnitude of sentiment in the text of the article at the sentence level while keeping track of contextual issues such as anaphora resolution. By creating a two-dimensional vector to represent the sentiment for each named entity in each sentence (see Figure 1), we can create an overall vector to match this to the overall sentiment of the article. In Figure 1, the blue lines represent the boundaries between the classifications of sentiment, from very negative to very positive. Note that some of the boundary lines between sentiment ratings (the blue lines) are not strictly vertical; if a word is more objective, the threshold for it to be at the extremes (either very positive or very negative) is lower than that when the term is denoted as subjective. We discuss how we classify these terms in the next section.

Long Short Term Memory (LSTM) Models

We use the LSTM model introduced by Hochreiter and Schmidhuber [8], and subsequently modified to include forget gates as implemented by Gers, Schmidhuber, Cummins in [4] and by Graves in [5]. LSTMs have been traditionally applied to machine translation efforts, but here we apply them to classifying sentiment.

With RNNs, a weight matrix is associated with the connections between the neurons of the recurrent hidden layer. The purpose of this weight matrix is to model the synapse between two neurons. During the gradient back-propagation phase of a traditional neural network, the gradient signal can be multiplied many times by this weight matrix, which means it have a disproportionately strong influence on the learning process.

When weights in this matrix are small (i.e., the leading eigenvalue of the weight matrix < 1.0), a situation called vanishing gradients can occur. In this situation, the gradient signal gets so small that learning either becomes very slow or may stop completely. This has a negative impact on learning the long-term dependencies in the data. However, when the weights in this matrix are large (i.e., the leading eigenvalue of the weight matrix > 1.0), the gradient signal can become so large that learning will diverge, which is often referred to as exploding gradients.

Minimizing the vanishing and exploding gradients is the primary motivation behind the LSTM model. This model

---

[1] https://www.allsides.com/unbiased-balanced-news

introduces a new structure called a memory cell (see Figure 2). A memory cell is comprised of four main elements: (a) an input gate, (b) a neuron with a self-recurrent connection, (c) a forget gate, and (d) an output gate. The self-recurrent connection maintains a weight very close to 1.0. Its purpose is to ensure that from one timestep to the next, barring any outside interference, the state of a memory cell will remain constant. The gates serve to modulate the interactions between the memory cell itself and its environment. The input gate can allow incoming signal to alter the state of the memory cell or block it. On the other hand, the output gate can allow the state of the memory cell to affect other neurons. Last, the forget gate modulates the memory cell's self-recurrent connection, allowing the cell to remember to ignore, or forget, its previous state.

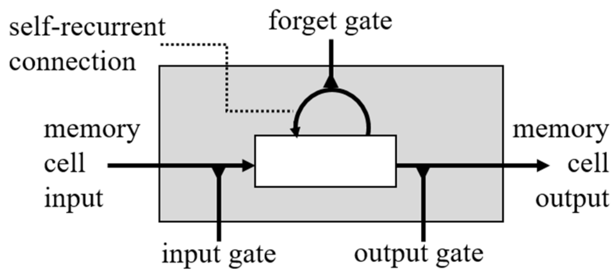

Figure 2: Illustration of an LSTM memory cell.

The following equations illustrate how a layer of memory cells is updated at timestep t. We define x_t and h_t as the input and output, respectively, to the memory cell layer at time t, $\mathbf{W_i}$, $\mathbf{W_f}$, $\mathbf{W_c}$, $\mathbf{W_o}$, hidden-state-to-hidden-state matrices $\mathbf{U_i}$, $\mathbf{U_f}$, $\mathbf{U_c}$, $\mathbf{U_o}$, are the weight matrices, and $\mathbf{b_i}$, $\mathbf{b_f}$, $\mathbf{b_c}$ and $\mathbf{b_o}$ are the bias vectors. First, we determine the values for the input gate, $\mathbf{i_t}$, and the candidate values for the states of the memory cells at time t, $\tilde{C}_t$:

(1) $\mathbf{i_t} = \sigma(\mathbf{W_i} x_t + \mathbf{U_i} h_{(t-1)} + \mathbf{b_i})$

(2) $\tilde{C}_t = \tanh(\mathbf{W_c} x_t + \mathbf{U_c} h_{(t-1)} + \mathbf{b_c})$

Next, we compute the value for $\mathbf{f_t}$, the activation of the memory cells' forget gates, at time t:

(3) $\mathbf{f_t} = \sigma(\mathbf{W_f}\, x_t + \mathbf{U_f}\, h_{(t-1)} + \mathbf{b_f})$

Given the value of the input gate activation it, the forget gate activation, $\mathbf{f_t}$, and the candidate state value, $\tilde{C}_t$, we can compute C_t, the memory cells' new state, at time t:

(4) $C_t = i_t * \tilde{C}_t + \mathbf{f_t} * C_{(t-1)}$

where * denotes a point-wise (Hadamard) multiplication operator. Once we obtain the new state of the memory cells, we can compute the value of their output gates, $\mathbf{o_t}$, and their outputs, $\mathbf{h_t}$:

(5) $\mathbf{o_t} = \sigma\,(\mathbf{W_o}\boldsymbol{x_t} + \mathbf{U_o}\boldsymbol{h}_{(\mathbf{t-1})} + \mathbf{b_o})$

(6) $h_t = \mathbf{o_t} * \tanh(C_t)$

Our model is a variation of the standard LSTM model; here the activation of a cell's output gate is independent of the memory cell's state $\tilde{C}_t$.

This variation allows us to compute equations (1), (2), (3), and (5) in parallel, improving computational efficiency. This is possible because none of these four equations rely on a result produced by any of the other three. We achieve this by concatenating the four matrices $\mathbf{W}_*$ into a single weight matrix $\mathbf{W}$, performing the same concatenation on the four weight matrices $\mathbf{U}_*$ to produce the matrix $\mathbf{U}$, and the four bias vectors $\mathbf{b}_*$ to produce the vector $\mathbf{b}$. Then, the pre-nonlinearity activations can be computed with:

(7) $z = \mathbf{W}\boldsymbol{x_t} + \mathbf{U}\boldsymbol{h}_{(\mathbf{t-1})} + \mathbf{b}$

The result is then sliced to obtain the pre-nonlinearity activations for $\mathbf{i}$, $\mathbf{f}$, $\tilde{C}_t$, and $\mathbf{o}$. These non-linearity activations are then applied independently to their respective cells.

Our model is composed of a single LSTM layer followed by an average pooling and a logistic regression layer as illustrated in Figure 3. From an input sequence $x_0, x_1, x_2, \ldots, x_n$, the memory cells in the LSTM layer will produce a representation sequence $h_0, h_1, h_2, \ldots, h_n$. This representation sequence is then averaged over all n timesteps resulting in representation, h. Last, this representation is fed to a logistic regression layer whose target is the class label associated with the input sequence, which is the five ordinal levels of sentiment, ranging from very positive to very negative. To map these vectorized terms (as seen in Figure 1) to an ordinal value for sentiment, we take the cosine of the term vector.

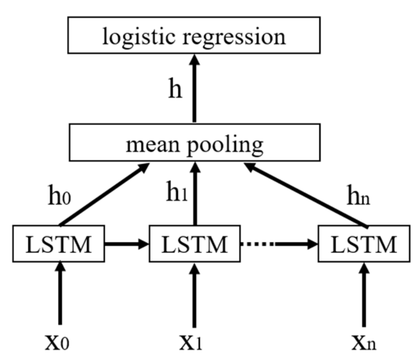

Figure 3: It is composed of a single LSTM layer followed by mean pooling over time and logistic regression.

INTERFACE COMPONENTS

Figure 4 illustrates the flow of a user query involving a named entity on our interactive query interface. In this section, we describe the major steps and related interfaces.

Data Collection

We use a collection of 433,175 news articles scraped from 211 formal and informal news sources. Of the 211 news sources, 109 of these are from verified sources. We determine verified sources as those from Media Bias/Fact Check that indicate a factual reporting score of "high". The articles in our collection are on a variety of topics, but all are

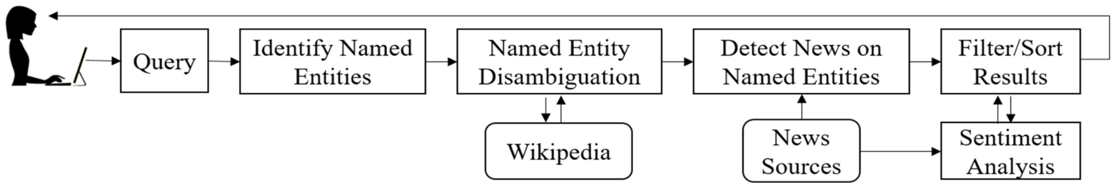

Figure 4: Flow diagram showing the major components of the search system.

written in English, have publication dates from 2012-2017, and are available on the internet (although some are available only through paywalls). Figure 5 illustrates the distribution of news articles, news sources, and verified sources for each year in our collection.

The processing of the data in the collection was designed to be done quickly. Using a single server, we were able to index, detect and classify sentiment for the entire collection of 433,175 articles in approximately 4 minutes, allowing us to handle emergent stream data (i.e. Twitter) with only a minor delay.

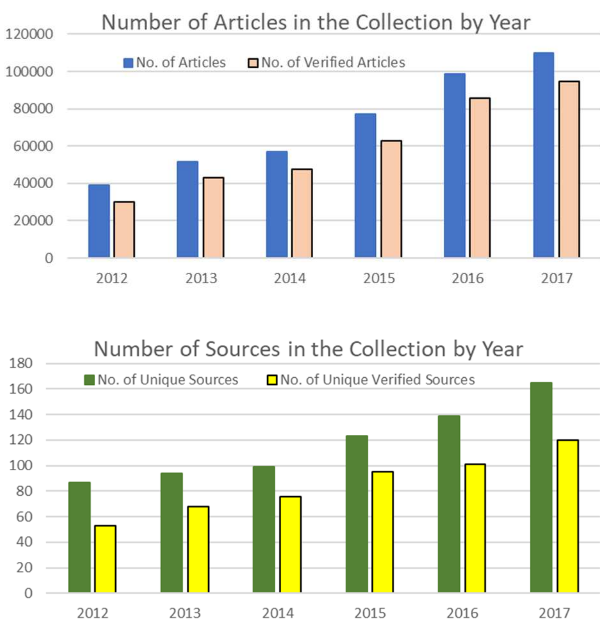

Figure 5: Number of articles (top) and number of unique sources (bottom) in our collection, by publication date of the article.

Training of the LSTM Network

The dataset used for training is the recently proposed Stanford Sentiment Treebank [11], which includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences. In our experiment, we focus in sentiment prediction of complete sentences with respect to the named entities contained within each sentence.

For our LSTM, we use a use the softsign activation function over tanh; it is faster than softmax and there is a smaller probability of saturation (i.e., having a gradient that approaches 0). We evaluated our training set over 20 epochs, (which was empirically determined). We use a learning rate of 10^{-5}, an L2 regularization weight of 0.009, and dropout value of 1.0.

Interactive Query Interface

Figure 6 shows the interactive query interface used in our study. The query interface is designed to give users as much information to refine their search based on the sentiment of the search results. The interface is divided into two columns. The left column contains an area to enter and refine queries, a checkbox for the user to only have results from verified sources returned, several checkboxes to determine the types of sentiments to include, from very negative to very positive. At the bottom of the left-hand column, the most popular search terms not used in the user query appear in the results, with color coding to indicate the sentiment of the term.

In the right-hand column, we have a display of the article counts by sentiment, and the top-ranked search results. Users are also given the ability to sort the search results based on relevance, date, sentiment, or verified source.

Next to each search result, users can see the sentiment our approach has indicated for that article, as well as an indication if the article is from a verified source.

We implemented searches on our collection using Indri, a scalable open-source search engine [12]. Indri works well with smaller queries, which are typically used in searches on named entities.

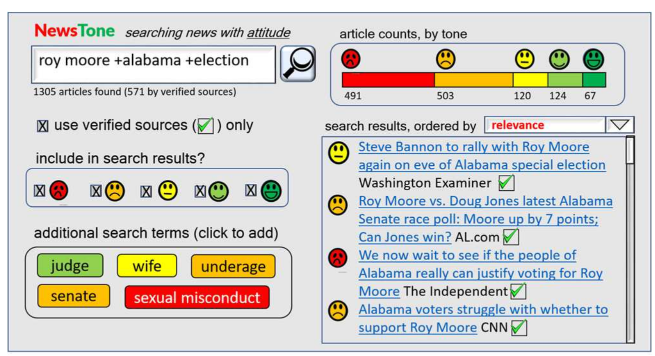

Figure 6: The Interactive Query Interface for searching our collection, showing an example query. The sentiment we derive from each article is represented as the sentiment of the article.

Detecting Ambiguous Named Entities

To ensure we are tracking the correct named entity, when appropriate, we need to disambiguate potentially confounding entities. We use an API from Wikipedia to check for a disambiguation page on that user-provided named entity. If one is found, we obtain the different categories, if any, that are provided by Wikipedia. Figure 7 shows an example of a search on “Michael Jackson” and the categories containing entities named “Michael Jackson”. This allows users to narrow their search to the correct entity, reducing the possibility of confounding results from mistakenly grouping disparate entities together.

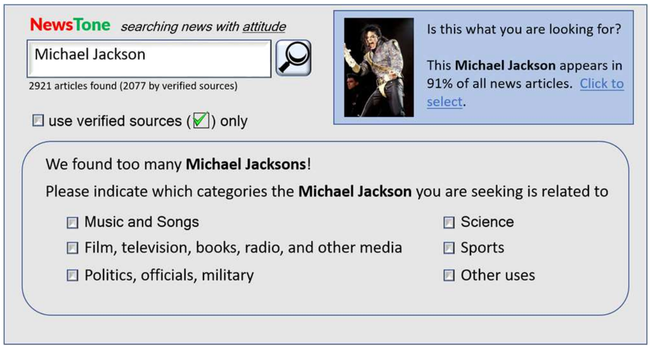

Figure 7: The disambiguation page for Michael Jackson. Categories are pulled from Wikipedia through their API, allowing the user to find the correct Michael Jackson. Note the shortcut in the upper right-hand side linking to the most popular named entity.

Detecting Verified Sources

As described earlier, we allow users to search on only verified news sources or all sources. This allows users to examine both informal and formal sources. We describe how we verify sources in the Data Collection section. Figure 8 shows search results without the verified sources only checkbox checked, allowing unverified sources.

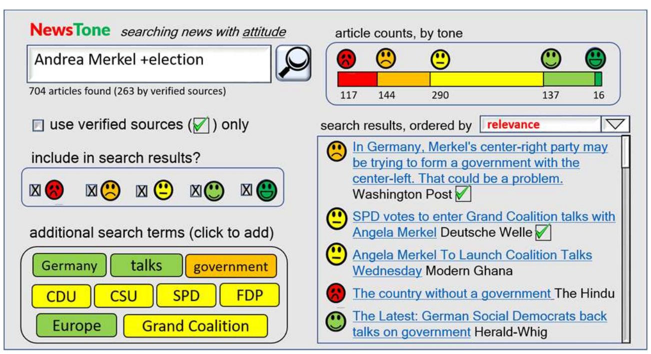

Figure 8: The Interactive Query Interface for searching our collection, showing search results containing unverified sources

Applying Sentiment Analysis

We use the LSTM method to detect and classify sentiment analysis for each major named entity in each article as well as the main keywords associated with that article. We provide five classes of sentiment, from very negative to very positive. We display this information to the user as the sentiment of the article.

EXPERIMENT DESIGN

Sentiment analysis is primarily associated with a named entity, so if multiple entities are described in the article text, each with a different sentiment, this can convolute the true sentiment around each entity if not properly handled. Also, the sentiment of the article is a relative concept – if all articles are negative about a named entity, even a slightly positive article can look very positive in comparison. Our research question is to evaluate if machine generated sentiment analysis is a strong predictor of article sentiment from a user’s perspective. We accomplish this by evaluating feedback on the sentiment rating the users provide.

Evaluating Sentiment

As with determining relevance in information retrieval, humans widely known to be better than machines at determining the correctness of article sentiment. We hired 293 crowdworkers from Amazon Mechanical Turk. These crowdworkers performed 600 separate tasks (HITs) to evaluate 1500 articles (approximately 0.35% of our collection) by searching on 150 named entities. Each article was evaluated by at least 3 different crowdworkers (crowdworkers could not evaluate an article more than once). The distribution of ratings made by crowdworkers is given in Figure 9. Most raters evaluated 5 articles and the mean number of articles rated was 15.

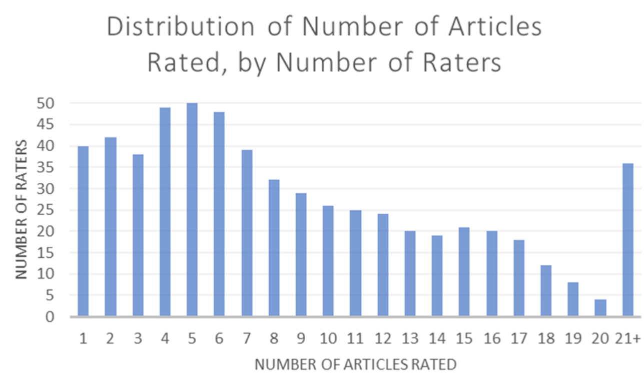

Figure 9: The number of articles rated (x-axis) by the number of raters evaluating that number of articles (y-axis).

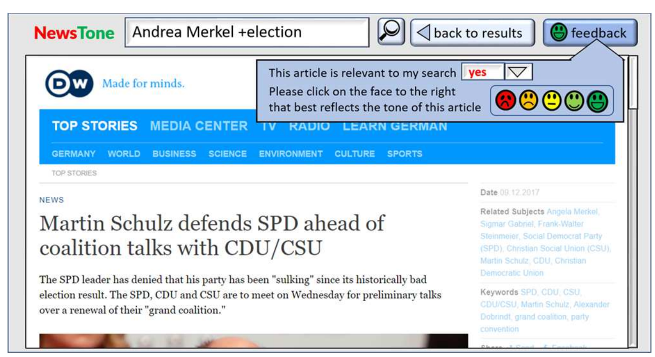

Figure 10: The interface used to evaluate the article’s relevance and classification of the article’s sentiment.

Instructions to Users

Each user (crowdworker) is asked to determine if the article retrieved by their query is relevant to the search criteria. This is used to help refine the search criteria parameters provided in Indri. More importantly, the user is asked to evaluate the sentiment assigned to the article on a five-point scale (see Figure 10). Users were also asked to take a survey on usability of the interface and the perceived accuracy of the LSTM classified sentiment.

Intra-Rater Reliability

To evaluate intra-rater reliability, we kept track of each crowdworker's ratings and the articles they rated without identifying them personally. When the articles were presented to the crowdworker to rate, they were not made aware of the overall rating previously made by our sentiment analysis model. We also kept track so that a single user could not evaluate any article more than once.

We understand that the raters' personal bias can influence their perspective on an article's sentiment. Although we did not attempt to recalibrate each crowdworker's ratings based on the pattern of their ratings, we did see if any crowdworker consistently selected the sentiment to be very positive or very negative, implying they were rushing through the task instead of evaluating each article thoroughly. Of the 600 tasks, only 3 tasks needed to be repeated due to this behavior.

Fake News

We also wish to determine if outliers in sentiment on a named entity were good predictors of fake news. For example, if a large percentage of articles for an entity are slightly positive or very positive, those articles with sentiments rated very negative (particularly from unverified sources) are candidates to be fake news articles. To examine the details further, we look at the most negative quotations or facts provided in these articles using a separate process, and look at the overlap between these sources and other articles in our collection. We briefly report and analyze these findings.

RESULTS AND ANALYSIS

Our primary research question was to examine how well the sentiment analysis provided by our LSTM model correlates with the sentiment rating made by users. Since each of the articles was evaluated at least 3 times, we took the average rating of the users (rounded to the nearest integer) to be the correct article sentiment.

We performed a Pearson correlation coefficient, r, on the 5 sentiment classes determined by our LSTM network with the 5 sentiment classes provided by the users. There was a positive correlation between the two variables [$r = 0.823$, $n = 1500$, $p < 0.001$]. Therefore, based on the sample of 1500 news articles evaluated, we believe the sentiment analysis provided by the LSTM model is a reasonably good predictor of an article's sentiment. Table 1 shows the correlation between the two sets of ratings.

To evaluate fake news articles, we examine named entities where the sentiment is skewed heavily in one direction (either very positive or very negative) and looked at those articles which were extreme outliers, or a difference in ratings of 3 or more on our 5-point scale. Of the 150 named entities examined in our study, we found 14 that had one or more articles meeting this condition. These 14 named entity searches yielded 29 articles, of which 28 were unverified news articles.

We ran a separate analysis of any quotations and facts raised in each of these 28 articles. We then tried to find these facts mentioned in the other articles. Of the 31 quotations in these articles about the named entity in question, we found 20 instances where the quotations did not exist in any other article in our collection and 11 instances where these quotations were mentioned, but convoluted in a way to contort its context. Of the 89 facts raised in these articles, 77 of these were not mentioned in any other article, and 12 that were mentioned but taken out of context with respect to the other articles in our collection. While we cannot confidently conclude that these articles represent fake news, we believe this approach can help identify articles that have a distinctly different sentiment from other articles and bring up quotations and facts not mentioned in other articles. We plan to explore this relationship in a future study.

Last, we asked the crowdworkers to provide optional feedback on the interface both in terms of usability and in terms of accuracy of sentiment classification on a five-point Likert scale. Of the 293 crowdworkers, we received feedback from 177 (60.4%). Survey takers scored the interface as 3.28 for usability (5 = best), with many providing comments that more work needs to be done to reduce its complexity. The survey takers scored the LSTM model's sentiment classification accuracy as 4.54, with many providing feedback indicating they concurred with the LSTM model's overall accuracy.

Correlation of Ratings between Users and the LSTM Model		**Rating Obtained by the LSTM Sentiment Analysis Model**				
		1	**2**	**3**	**4**	**5**
Average of User Ratings (min of 3 Ratings per Article)	**1**	**122**	70	6	3	0
	2	67	**242**	77	5	0
	3	4	84	**192**	79	5
	4	0	10	73	**250**	74
	5	0	2	11	45	**79**

Table 1: Correlation of Ratings between the average supplied by the users and those obtained by sentiment analysis model.

CONCLUSION AND FUTURE WORK

In this paper, we describe an interactive query interface that makes use of sentiment analysis. This allows users performing a named entity search to receive information on the sentiment of the article and therefore find a wide diversity of opinions on a named entity search quickly and easily.

We describe the LSTM model we used, and how this can be used to classify sentiment of the news article text into five classes, ranging from very negative to very positive. The advantage of this model is that even when multiple entities are mentioned in an article it can match the sentiment for the named entity in question. We have shown that this technique can process news articles quickly, allowing emergent news to be covered quickly.

We conducted a user study with 293 unique participants to answer a research question. They were instructed to classify the sentiment of 1500 articles and indicate how this sentiment correlates with the sentiment obtained from our model. Each article was evaluated by at least 3 users. With a Pearson correlation coefficient, r=0.823, we found the classification of article sentiment and the classification from the LSTM sentiment analysis tool are strongly correlated.

Combining the sentiment classification techniques with some additional analysis allows us to identify potentially fake news articles. We identified news articles where the ratings were outliers from a majority of the other relevant articles using the same named entity search. We find that 28 of the 29 articles identified using this approach were suspicious news articles and would need further investigation. We leave this for a future study.

There are some limitations to our work. First, out study only looks at queries on named entities, which are easier to retrieve and analyze semantically than general concepts. Second, the study worked with a collection of 433,175 articles, with 84.1% of these being pulled from verified sources. With exposure to more unverified sources our correlation may be lower, which we leave for future work.

Another limitation has to do with the sentence complexity. Our model evaluated sentiment at the sentence level. We found proximity to the named entity played a role; if more than one named entity was mentioned in a sentence, such as "In the 1938 movie *Carefree*, Fred Astaire performed well but Ralph Bellamy was forgettable.", we would expect our model to provide a positive sentiment for Fred Astaire, a neutral sentiment for "Carefree" and a negative sentiment for "Ralph Bellamy"; instead it provided a positive sentiment for "Fred Astaire" and "Carefree" and a neutral sentiment for "Ralph Bellamy". Evaluating at the phrase level instead of the sentence level will improve the accuracy of our results.

In other future work, we plan to examine the role of images in articles and how this can be analyzed for sentiment as well. We also plan to examine the choice of photos used to represent named entities in news articles. We plan to examine searches that don't contain named entities and evaluate if our methods are as accurate as they are with named entities.

REFERENCES

1. Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election (No. w23089). National Bureau of Economic Research.
2. Bakshy, E., Messing, S., & Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on Facebook. Science, 348(6239), 1130-1132.
3. Barthel, M., Michell, A, and Holcomb, J. (2016). Many Americans Believe Fake News Is Sowing Confusion. Pew Research Center. Available at: http://www.journalism.org/2016/12/15/many-americans-believe-fake-news-is-sowing-confusion/
4. Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM. Neural Computation, vol. 12, 2451–2471
5. Gottfried, J, and Shearer, E. (2017) Americans' online news use is closing in on TV news use. 2017. Pew Research Center. Available at: http://www.pewresearch.org/fact-tank/2017/09/07/americans-online-news-use-vs-tv-news-use/
6. Graves, A. (2012). Supervised sequence labelling with recurrent neural networks (Vol. 385). Heidelberg: Springer. Chicago
7. Hasell, A., & Weeks, B. E. (2016). Partisan provocation: The role of partisan news use and emotional responses in political information sharing in social media. Human Communication Research, 42(4), 641-661.
8. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
9. Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? In Proceedings of the 19th international conference on World wide web (pp. 591-600). ACM.
10. Purcell, K., Brenner, J., & Rainie, L. (2012). Search engine use. 2012. Pew Research Center. Available at: http://www.pewinternet.org/files/old-media/Files/Reports/2012/PIP_Search_Engine_Use_2012.pdf
11. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642).
12. Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005, May). Indri: A language model-based search engine for complex queries. In Proceedings of the International Conference on Intelligent Analysis (Vol. 2, No. 6, pp. 2-6).
13. Yin, X., & Shah, S. (2010). Building taxonomy of web search intents for name entity queries. In Proceedings of the 19th international conference on World wide web (pp. 1001-1010). ACM.