Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Neural User Response Generator: Fake News Detection with Collective User Intelligence

Feng Qian[*1], **Chengyue Gong**[*1], **Karishma Sharma**[*2], **Yan Liu**[2]
[1] Peking University
[2] University of Southern California
nickqian@pku.edu.cn, cygong@pku.edu.cn, krsharma@usc.edu, yanliu.cs@usc.edu

Abstract

Fake news on social media is a major challenge and studies have shown that fake news can propagate exponentially quickly in early stages. Therefore, we focus on early detection of fake news, and consider that only news article text is available at the time of detection, since additional information such as user responses and propagation patterns can be obtained only after the news spreads. However, we find historical user responses to previous articles are available and can be treated as soft semantic labels, that enrich the binary label of an article, by providing insights into why the article must be labeled as fake. We propose a novel Two-Level Convolutional Neural Network with User Response Generator (TCNN-URG) where TCNN captures semantic information from article text by representing it at the sentence and word level, and URG learns a generative model of user response to article text from historical user responses which it can use to generate responses to new articles in order to assist fake news detection. We conduct experiments on one available dataset and a larger dataset collected by ourselves. Experimental results show that TCNN-URG outperforms the baselines based on prior approaches that detect fake news from article text alone.

1 Introduction

In recent years, the proliferation of fake news on social media has become a major problem in our society. The problem has become so serious that in January 2017, a spokesman for the German government states that they "are dealing with a phenomenon of a dimension that [they] have never seen before".[1] In addition, research by [Friggeri *et al.*, 2014; Dow *et al.*, 2013] showed that fake news propagates exponentially at the early stage and can cause a significant loss in a short amount of time. For example, an instance was cited by Time Magazine in 2013 when a false announcement of Barack Obama's injury in a White House explosion "wiped off 130 Billion US Dollars in stock value in a matter of seconds".[2] Therefore automatically detecting fake news at the early stage has attracted significant research interest in both industries and academia.

Existing work on fake news detection perform poorly on early detection because most of them mainly utilize network-based information, such as user responses to news articles on social networks [Ma *et al.*, 2016; Ruchansky *et al.*, 2017; Castillo *et al.*, 2011], which is only available *after* a news article has been circulated and exposed to a large number of users. In this paper, we study the early fake news detection problem under the assumption that the text of the news article is the only information available at the time of detection. However, we notice that user responses towards previously propagated articles are available, and can be leveraged to enhance early detection performance on new articles for which user responses are not available.

Specifically, we treat user responses as soft semantic labels that essentially enrich the binary $\{0, 1\}$ labels by providing insights into why the article must be labeled as fake, based on the collective wisdom of users. To illustrate this, we sample a fake news article from the dataset with four corresponding user responses as shown in Figure 1, related to the banning of a drug by the FDA. The responses contain information that can explain why this article is fake, as stated by the users. We propose Two-Level Convolutional Neural Network with User Response Generator (TCNN-URG) which can provide deeper semantic analysis and understanding of the news article text and its veracity through the relationship between the article text content and the corresponding user responses it invokes.

The proposed **Two-Level Convolutional Neural Network** (TCNN) is designed to first condense word level information into sentences and the process the sentence level representations with a Convolutional Neural Network, to effectively capture semantic information from long article texts which can be used to classify the article as fake or not. The **User Response Generator** (URG) component is de-

[*]Equal contribution. The work is done while Feng Qian is a visiting student at University of Southern California.

[1]http://www.theguardian.com/world/2017/jan/09

[2]http://business.time.com/2013/04/24/how-does-one-fake-tweet-cause-a-stock-market-crash/

A fake news article with user responses	... FDA quietly bans powerful life-saving intravenous Vitamin C ...It would be naive to think that the FDA endeavours to protect the public's health as its primary focus ...
User 1	This is an **absolute disgrace**. It is a **well known cure**.
User 2	**Why** is the #FDA quietly banning **life-saving** #natural #medicine ?
User 3	It's funny. [[@ username]], I just **had it yesterday** in the hospital of [[# hashtag]]
User 4	Not really reliable, since a drug need to be **tested repeatedly** before being approved by FDA. They won't ban something so easily. It **costs too much money**...

Figure 1: An example to show that why user responses can be utilized as rich soft semantic labels to help the early fake news detection system.

signed to learn a generative model of user responses to article texts from the historical user responses to true and fake news articles and utilize the learned model to generate responses towards new articles to assist in detection. The TCNN-URG are combined to perform early fake news detection in that TCNN extracts article representation and URG generates user responses conditioned on the article representation and the article representation and generated user response are used for final classification. We conduct experiments on the Weibo dataset used in [Ma *et al.*, 2016; Ruchansky *et al.*, 2017] and a self-collected Twitter dataset of significantly larger size with longer article lengths to test the effectiveness of the proposed model.

The main contributions of this paper can be summarized as follows:

- User response generator (URG), a novel conditional generative model, learns how users respond to news articles. The users' wisdom is automatically inferred by the conditional generative model, which helps to enhance prediction accuracy when user responses are not available during early detection.
- The Two-Level Convolutional Neural Networks (TCNN) can effectively learn features from news articles in a two-level manner, i.e., condensing word-level information into sentence-level and applying the convolution over sentence-level representations, to capture semantic information effectively from longer article texts.
- TCNN-URG is an effective model for early fake news detection given the texts only in the detection phase. In particular, the special structure of TCNN-URG can effectively extract the soft semantic labels from user responses to guide the training of TCNN.

The rest of the paper is organized as follows: first, we survey related works on previous fake news detection approaches and provide necessary background on variational autoencoder (VAE); second, we describe the model architecture of TCNN-URG and the training procedure; and finally, we conduct qualitative and quantitative experiments to evaluate the effectiveness of TCNN-URG.

2 Related Works and Background

2.1 Fake News Detection

Earlier fake news detection works were mainly based on manually designed features extracted from news articles or information generated during the news propagation process [Castillo *et al.*, 2011; Ma *et al.*, 2015]. Though intuitive, manual feature engineering is labour intensive, not comprehensive, and hard to generalize. In contrast to manual feature engineering, deep learning models can extract features automatically from the text. [Wang, 2017] proposed a convolutional neural network to classify short political statements as fake or not using the text features of the statements and available metadata. [Volkova *et al.*, 2017] used recurrent neural networks for a similar classification problem and proposed to additionally feed linguistic cues into the network.

Other works focused on the modeling information diffusion patterns, assuming that true news and fake news has different ways of propagating. [Jin *et al.*, 2013; Ma *et al.*, 2016; Ruchansky *et al.*, 2017]. Further, [Tschiatschek *et al.*, 2017] leveraged crowd-sourcing solutions by processing users flagging and reporting of content. Unfortunately, these works cannot be applied for early detection because the main source of information they consider, such as enough user responses, is only available after a news article has already become popular.

2.2 Variational Autoencoder and Conditional Variational Autoencoder

Due to the close relationships between VAE, CVAE and the proposed URG module in TCNN-URG, detailed background information about VAE and CVAE is given in this subsection. Variational Autoencoder (VAE) [Kingma and Welling, 2014] is a generative model consisting of an encoder, a decoder and a loss function. This powerful probabilistic model is constructed by an inference neural network, namely the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, and a generative neural network, namely the decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, where ϕ, θ are the parameters of the respective networks. The encoder encodes information $\boldsymbol{x}$ to latent variables $\boldsymbol{z}$, and the decoder decodes latent variables $\boldsymbol{z}$ to try to reconstruct $\boldsymbol{x}$. Since VAE is trained to reconstruct original information from latent variables, VAE is considered a suitable generative model for tasks like handwritten numbers generation. The network parameters are jointly trained by maximizing the following objective where training samples are indexed by (i):

$$-D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}) \,||\, p_\theta(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z})\right] \tag{1}$$

The first term is the negative of the KL-divergence between the encoder's distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and prior $p_\theta(\boldsymbol{z})$ and acts as

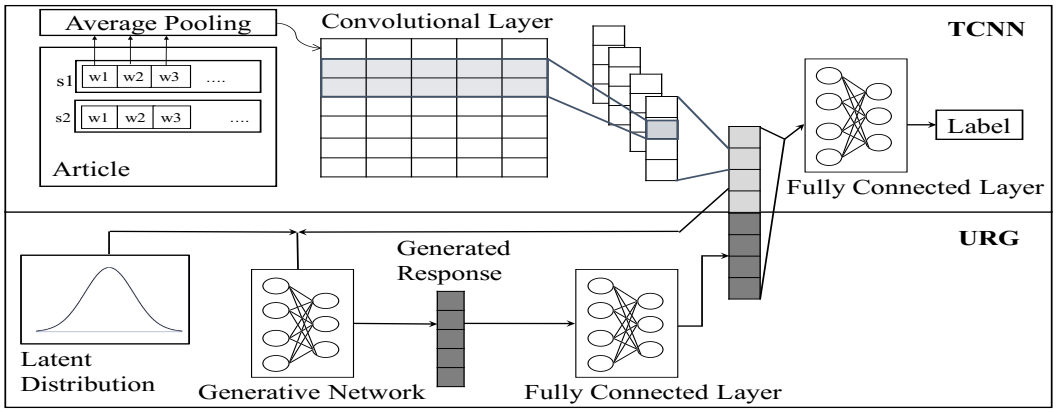

Figure 2: Architecture of TCNN-URG

a regularizer to encourage the distribution of latent variables to be closer to the prior distribution. The second term controls the reconstruction of input by maximizing the log-likelihood of regenerating the input $\boldsymbol{x}$, encouraging the decoder to learn to reconstruct the data from samples drawn from the latent distribution. Conditional Variational Autoencoder (CVAE) [Sohn *et al.*, 2015] which is an extension of VAE allows the data generation process to be conditioned on particular information y. Specifically, the encoding is now characterized by $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ and the decoding by $p_\theta(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z})$, by taking into account the conditioning on $\boldsymbol{y}$. URG in the proposed model is designed based on CVAE in the following section 3.

3 Model

In this section, we introduce the architecture of the proposed TCNN-URG early fake news detection model. TCNN-URG is composed of two parts: (1) TCNN represents each article in a two-level manner and is able to apply pure text classification on news articles and (2) URG is trained to learn how users respond to news articles, and can generate user responses to aid TCNN with user wisdom when user response is not available. The architecture of the proposed model is shown in Figure 2.

3.1 Notations

We consider the setting where we have a set of news articles D, and each article is denoted as d_i. Each article d_i is composed of a sequence of sentences $\boldsymbol{s}_1, \boldsymbol{s}_2 ... \boldsymbol{s}_{n_i}$, where n_i is the number of sentences in the article d_i. Each sentence, is a sequence of words, and $v(s)$ is the vector representation for each sentence. In the proposed model, the final feature vector extracted for each article d_i for classification is marked as $\boldsymbol{y}_i$. For each article, there will be several related user responses. A given response to article d_i is marked as $\boldsymbol{x}_{ij}$, where $j \in [1, J_i]$ and $j \in \mathcal{N}$. J_i stands for the number of the responses about the article d. For each article d_i, the target is marked as f_i. $f_i = 1$ means this article is true news, and $f_i = 0$ means this article is fake news.

3.2 Problem Definition

Considering that user responses $\boldsymbol{x}$ corresponding to each article d are not available during real-world fake news detection, the detection task can be defined as: given a news article d_i, the target is to predict the corresponding label f_i. However, it is important to understand that user response $\boldsymbol{x}$ corresponding to each article d is available in historical training data during the training process.

3.3 Two-Level Convolutional Neural Network

The TCNN extracts semantic information from the news article text using a two-level representation. As mentioned in section 1, the motivation to design the TCNN to first condense word-level information into sentence-level representations and in turn represent the article at the level of sentences, is to enable the learned representation of the article to capture rich semantic information and text features from not only short but also long articles effectively.

We first derive the sentence representation as the average of the word embeddings of words present in the sentence. Each sentence in a news article is represented by a one-hot vector $\boldsymbol{s} \in \{0, 1\}^{|V|}$ indicating which words from vocabulary V are present in the sentence. Then the sentence representation is defined by average pooling of word embedding vectors of words in the sentence. The operation is defined as:

$$\boldsymbol{v}(\boldsymbol{s}) = \frac{\boldsymbol{W}\boldsymbol{s}}{\sum_k s_k} \tag{2}$$

where $\boldsymbol{W}$ is the embedding matrix of all the words in the vocabulary, s_k marks the k^{th} dimension of the one-hot vector $\boldsymbol{s}$. Embedding of each word in $\boldsymbol{W}$ is pre-trained by skip-gram algorithm on all articles in the dataset.[3]

The article representation is derived from the sentence representations by concatenation of each sentence representation. The article d_i, containing n_i sentences, is represented as:

$$s_{1:n_i} = v(s_1) \oplus v(s_2) \oplus ... \oplus v(s_{n_i}) \tag{3}$$

where $\oplus$ is a concatenation operator. Note that each sentence is represented on a word level, the news article now is represented on a sentence level as shown in Equation 3.

[3]Word embeddings are trained by [Mikolov *et al.*, 2013] on all news articles in dataset.

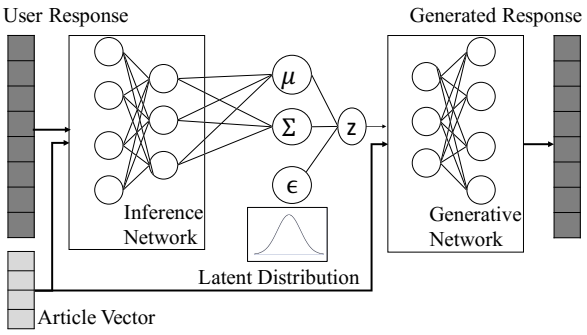

Figure 3: Generative process of URG conditional generator

A convolution operation then applies a filter $w \in \mathbb{R}^{hk}$ to a window of h sentences moving through the article to extract semantic information features from the article. A feature c_i is generated from a filter t and a window of $s_{i:i+h-1}$ by:

$$c_i = g(t \cdot s_{i:i+h-1} + b) \tag{4}$$

where $b \in \mathbb{R}$ is the bias and g is an activation function. After that, a max pooling operation is applied to the feature map so that the maximum value within each window is taken as the output of the corresponding filter. Filters which have different lengths or have a same length but with different parameters are applied in order to capture features of different lengths and meaning.

Finally, the features are used as the input to a fully connected layer and a softmax output. If this TCNN structure is to be trained individually, the aim is to predict the target f_i for each news article d_i. The objective is then a negative likelihood, as:

$$-\sum_{d_i \in D} \log\ p(f_i|d_i, \theta) \tag{5}$$

where θ represents all the neural network parameters and D is the training article set.

3.4 User Response Generator

A generative Conditional Variational Autoencoder (CVAE) is chosen to be the foundation of User Response Generator (URG). More specifically, the CVAE is trained to generate user responses using a specific article as the condition. Traditionally, deterministic recurrent neural network (RNN) encoder-decoder models are often used for natural language generation tasks. However we specifically choose CVAE in order to model stochasticity in user responses to a given article, which is more natural for this task. CVAE can learn a distribution over user responses, conditioned on the article, and can therefore be used to generate varying responses sampled from the learned distribution.

CVAE is applied for modeling the relations between the article $\boldsymbol{y}$, the user response $\boldsymbol{x}$ and the generative latent variable $\boldsymbol{z}$. The inference network and the generative network, namely the encoder and decoder is defined as: $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ and $p_\theta(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{y})$, where ϕ, θ are the parameters of the respective networks, since samples of both $\boldsymbol{x}$ and $\boldsymbol{z}$ and drawn under the influence of the article $\boldsymbol{y}$. Under the influence of article $\boldsymbol{y}$, encoder encodes user response $\boldsymbol{x}$ into latent variables $\boldsymbol{z}$, and then, decoder decodes latent variable $\boldsymbol{z}$ to reconstruct user response $\boldsymbol{x}$. We construct the input vector $\boldsymbol{y}$ as the article vector generated by TCNN by extracting semantic information from the sentence-level article representation. As for user response $\boldsymbol{x}$, we consrtuct a binary vector of vocabulary size to indicate which words appear in the user response and provide that is input to the encoder. The generative network learns to reconstruct the user response, resulting in a vector of vocabulary size with each component being the probability of the word's occurence in the user's response.

To learn the latent parameters of the generative model, we use the re-parameterization trick [Kingma and Welling, 2014] with $\boldsymbol{z} = g_\phi(\boldsymbol{\epsilon}, \boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, so that we can learn the parameters using back propagation by minimizing the following objective over each user response and article pair (there can be multiple user responses for a given article and we treat them as separate training samples indexed by i and j) as follows,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{z} \sim q_\phi(\boldsymbol{z}\ |\ \boldsymbol{x}^{(ij)}, \boldsymbol{y}^{(i)})} \left[-\log p_\theta(\boldsymbol{x}^{(ij)}|\boldsymbol{z}, \boldsymbol{y}^{(i)}) \right] + \\ D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(ij)}, \boldsymbol{y}^{(i)}) \,||\, p_\theta(\boldsymbol{z}))\end{aligned} \tag{6}$$

The first term is the reconstruction error designed as the negative log-likelihood of the data reconstructed from the latent variable $\boldsymbol{z}$ under the influence of article $\boldsymbol{y}$. The second term, the regularization, is used to minimize the divergence between the encoder distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})$ and the prior distribution $p_\theta(\boldsymbol{z})$. The architecture of URG is depicted in Figure 3.

3.5 Unified TCNN-URG System

TCNN is able to extract features from the article text and use that for predicting whether the article is fake or not. Whereas URG is able to generate user responses conditioned to a given news article. We use the text feature vector extracted by TCNN as article vector $\boldsymbol{y}$ to condition the response generated by URG when generating user responses to a given article. The user response generated by URG is put through a nonlinear neural network and then combined with the text features extracted by TCNN. Then, the final feature vector is fed into a feed forward softmax classifier for classification as shown in Figure 2 to predict whether the article is fake or true.

We formulate the problem of predicting output f for the article $\boldsymbol{y}$ as,

$$p(f|\boldsymbol{y}) = \int_{\boldsymbol{x}} p(f|\boldsymbol{y}, \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \tag{7}$$

where the integration is intractable, so it is approximated with the expectation for which we can derive a Monte Carlo estimate as follows,

$$p(f|\boldsymbol{y}) \approx p(f|\boldsymbol{y}, \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y})}\left[\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{z}\right]) \tag{8}$$

where $\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y})}$ is calculated as an average of several samples generated from the URG conditioned on article $\boldsymbol{y}$. For a good estimate of the expected value, we use the average of 100 samples of user responses generated from the URG.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
LIWC (feature engineering)	56.61	58.24	61.25	61.48	63.66	64.27	64.97	65.4	66.06
POS-gram	63.67	63.46	65.82	66.45	67.36	68.88	72.19	72.91	74.77
1-gram	**80.19**	80.89	81.32	81.35	82.47	83.01	83.59	84.03	84.76
CNN	74.03	81.34	81.89	82.82	84.01	84.56	84.86	85.11	86.23
TCNN	76.06	82.51	84.32	84.72	85.97	86.86	86.92	87.46	88.08
TCNN-URG	79.00	**84.52**	**85.51**	**86.26**	**88.05**	**88.41**	**88.43**	**88.56**	**89.84**

Table 1: Experimental results on Weibo dataset. Percentage on the top stands for the percentage of all data used as training data.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
LIWC (feature engineering)	51.8	53.47	55.2	56.25	57.74	58.96	60.34	60.9	62.13
POS-gram	57.55	64.44	66.36	68.21	69.06	69.32	69.86	69.87	70.34
1-gram	76.47	77.57	78.23	79.09	79.4	79.69	80.38	80.37	80.69
CNN	70.15	72.2	76.65	78.49	80.26	80.45	81.15	82.74	83.24
TCNN	77.46	77.59	78.21	80.18	81.61	83.73	84.29	85.96	86.02
TCNN-URG	**77.47**	**77.71**	**79.38**	**81.92**	**83.98**	**86.13**	**86.68**	**88.28**	**88.83**

Table 2: Experimental results on self collected fake news detection dataset. Percentage on the top stands for the percentage of all data used as training data.

This formulation for prediction combines the discriminative power of a deep learner that can automatically extract semantic information and other text features, with the flexibility provided by the generative model that can model the collective intelligence stored in user responses towards articles. In practice, the training process is divided into three steps:

1. TCNN is first trained separately. The training target of this step is to minimize the negative log-likelihood objective described in Equation 5. We use mini-batch gradient descent to optimize this training target.
2. URG is trained next. The training target is to minimize the loss function described in Equation. 6.
3. The TCNN and URG are combined together by first obtaining the article vector from TCNN and generating user responses based on the article vector according to formulation in Equation 8, followed by retraining the unified model to minimize the negative log-likelihood objective given by Equation 5. In this way, the TCNN is assisted by URG to enhance prediction ability.

4 Experiments

4.1 Dataset Description and Collection

To test the proposed early fake news detection system, we need both real news articles and corresponding user responses in the dataset. We chose to conduct experiments on a public Weibo (A Chinese social network) dataset [Ma *et al.*, 2016]. However, the body of news articles in this dataset is usually less than 100 words. In order to also test our model on longer news articles instead of only on short paragraphs, we need a dataset with both longer (at least 500 words) news articles and corresponding user responses. However, due to the unavailability of such a dataset, we collected a new dataset with both news articles and related user responses where the news articles have an average length of 950 words.

The dataset collection process is as follows. First, lists of websites are manually assessed and collected comprising of a set of trustworthy websites such *The Guardian*, *New York Times*, etc., and the other is a set of notorious fake news websites such as *NaturalNews*. We obtained article URLs from news articles under the different websites. Using the article URLs, we searched Twitter for user responses related to each article URL. Since the article URLs are referenced in Twitter posts (tweets) related to the article, we retain only those articles in our dataset for which we find at least one user response (tweet) on Twitter. We construct the dataset with these labeled articles along with related user responses incorporating the text and metadata information for both articles and responses. We plan to publish this dataset along with the collected list of websites.

4.2 Preprocessing and Pre-training Embedding

Since real articles and responses are collected from the Internet, the vocabulary is large and notations used are abundant. Some preprocessing process steps are taken before the data is fed as input into the proposed model:

- We separate punctuations and replace specific strings with tokens denoting their types. These notations are standardized as: mention notations such as @xx are converted into [[@]], hashtag #xx to [[#]], time notations such as “17:21" and “07:12:2013" are converted into [[time]], data notations such as “17.21" and “3/5" and “60%" are converted to [[data]], and money notations such as “$11" is converted to [[money]]. website urls to [[url]].
- We split articles into sentences and tokenize each sentence using Stanford CoreNLP tool. We pre-trained word embedding by skip-gram [Mikolov *et al.*, 2013] on all collected news articles.

4.3 Baselines

We compare our work to existing work that uses different techniques for detecting fake news from article text ranging from feature engineering to using linguistic techniques

as well as neural network models used in earlier work. The techniques we compare the proposed TCNN-URG to, are as follows:

1. **Feature engineering.** Based on the work of [Ott *et al.*, 2013], the first baseline we propose is based on using LIWC (Linguistic Inquiry and Word Count) features for text analysis. LIWC is a widely used lexicon in social science studies [Pennebaker *et al.*, 2015] [4].
2. **1-gram.** 1-gram features show good performance in deception detection works [Ott *et al.*, 2013; Feng *et al.*, 2012]. In order to keep comparison dimension fair, we use tf-idf to choose words between top 1000-2500 for the 1-gram features.
3. **POS-gram.** Linguistic methods based on Part-of-speech (POS) tags [5] shows good predictive power as found by [Feng *et al.*, 2012; Ott *et al.*, 2013] and are therefore chosen for comparison.
4. **CNN.** Convolutional neural networks have achieved state-of-the-art in text classification tasks and based on the work of [Wang, 2017] which demonstrates superior performance of CNN over recurrent neural architectures like the bidirectional LSTM (long short-term memory) for fake news detection, we choose CNN for comparison with. The text is represented at the word-level and fed to the CNN that extracts semantic representation of the article text for classification.

4.4 Experimental Setting

In the experiments, we set the word embedding dimension to be 128 and filter size to 2,4,5. For each filter size, 64 filters are initialized randomly and trained. When generating user responses from URG, we use the average of 100 samples to get accurate estimates of the expectation over the distribution of user responses generated. For training, we use a mini-batch size of 64 and articles of similar length are organized in the same batch. We build and train the model using TensorFlow and use ten-fold cross validation for evaluation of the model.

4.5 Results and Analysis

Experimental results on Weibo dataset are shown in Table 1, and results on self-collected Twitter dataset are shown in Table 2. We present the results in terms of the accuracy of detection, on varying percentage (10-90%) of data samples used as training data to evaluate the variation and stability in performance for the evaluated methods. Overall, TCNN outperforms the other methods compared against including CNN, and moreover, URG further improves the performance of TCNN and pushed the accuracy even higher, even when the training data is limited.

TCNN outperforms LIWC, POS-gram and 1-gram due to its ability to extract deep semantic information from the article text content. Moreover, TCNN outperforms CNN with the proposed two-level representation. Single layer CNN built over word-level article representations can only utilize combinations of several nearby words. However, by first condensing word-level information into each sentence, then deriving sentence-level representation for the news article, higher-level semantic information can be extracted more effectively, especially for longer article lengths as can be seen from experiment results. The improvement of TCNN over CNN is more pronounced in the Twitter dataset (Table 2) that contains articles of average length of about 1000 words as compared to the Weibo dataset (Table 1) with shorter articles of about 100 words. Furthermore, the difference in the detection accuracy is larger when the training data size is smaller for longer articles with the maximum being 7% higher than CNN for the Twitter dataset with 10% of the data used for training.

A fake news article sampled from the test set:			
... FDA quietly bans powerful life-saving intravenous Vitamin C ...It would be naive to think that the FDA endeavours to protect the public's health as its primary focus ...			
Top 20 words generated by URG in response:			
[[!]]	**[[?]]**	[[@]]	[[link]]
c	care	fda	food
false	health	intravenous	life
only	**problem**	protect	rich
tax	wait	watch	**why**

Figure 4: Top 20 response words generated by URG presented in alphabetical order.

URG further improves the accuracy of TCNN as can be seen in both result tables. URG learns the nature of user responses conditioned on the article text and is able to generate responses to new articles for early fake news detection. By capturing the intricate relationship between news articles and user responses, the URG empowers the system with user wisdom that is not directly available from the article text alone.

To further understand how URG works, we sample an example from the test set for which we already provided the true user responses in Figure 1. We specifically eliminated this article from the training data to provide insights into the capabilities of the URG as a generative model for generating user responses to unseen article texts. Even though the true responses are not seen by the URG, since the article is in the test set only, the URG is still able to generate reasonable responses to the article using the inferred latent parameters encoding the relationship between user responses and article contents. We provide the top 20 response words that are generated by URG for this example fake article as listed in alphabetical order in Figure 4. We can see that URG generates some negative responses and questioning responses such as [[?]], [[!]], fake and so on, as highlighted in the Figure 4, which are very important signals of fake news.

[4] Available at http://liwc.wpengine.com/

[5] We use the Stanford Parser [Klein and Manning, 2003] to obtain POS tags.

This demonstrates that the URG is able to capture intricate information expressed implicitly in user responses to other news articles, which it can utilize to guide the prediction for new articles. URG benefits from higher-level semantic reasoning extracted from historical user responses which indirectly act as soft semantic labels enriching the simple binary $\{0, 1\}$ labels of news articles that capture the subtle reasons behind why articles might be considered fake instead of just knowing whether they are fake or not.

5 Conclusion

Existing works cannot be applied to the problem of early fake news detection because most of them mainly rely on user response that is not available for early fake news detection. Whereas works that utilize only the article text for detection, ignore the rich information and latent user intelligence stored in user responses towards previously propagated articles. Our proposed TCNN-URG combines the power of discriminative fake news detection from article text feature extraction, with the power of generative modeling to leverage collective user intelligence on why articles must be true or fake and thereby simulate user responses for new articles to assist in early detection of fake news articles.

Acknowledgments

The work is supported in part by NSF Research Grant IIS-1619458 and IIS-1254206 as well as the National Natural Science Foundation of China NSFC Grant (NSFC Grant Nos.61772039, 91646202 and 61472006). The views and conclusions are those of the authors and should not be interpreted as representing the social policies of the funding agency, or the U.S. Government.

References

[Castillo *et al.*, 2011] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM, 2011.

[Dow *et al.*, 2013] P Alex Dow, Lada A Adamic, and Adrien Friggeri. The anatomy of large facebook cascades. *ICWSM*, 1(2):12, 2013.

[Feng *et al.*, 2012] Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 171–175. Association for Computational Linguistics, 2012.

[Friggeri *et al.*, 2014] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. Rumor cascades. In *ICWSM*, 2014.

[Jin *et al.*, 2013] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. Epidemiological modeling of news and rumors on twitter. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 8. ACM, 2013.

[Kingma and Welling, 2014] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[Klein and Manning, 2003] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

[Ma *et al.*, 2015] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754. ACM, 2015.

[Ma *et al.*, 2016] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119, 2013.

[Ott *et al.*, 2013] Myle Ott, Claire Cardie, and Jeffrey T Hancock. Negative deceptive opinion spam. In *HLT-NAACL*, pages 497–501, 2013.

[Pennebaker *et al.*, 2015] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[Ruchansky *et al.*, 2017] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM, 2017.

[Sohn *et al.*, 2015] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015.

[Tschiatschek *et al.*, 2017] Sebastian Tschiatschek, Adish Singla, Manuel Gomez Rodriguez, Arpit Merchant, and Andreas Krause. Detecting fake news in social networks via crowdsourcing. *arXiv preprint arXiv:1711.09025*, 2017.

[Volkova *et al.*, 2017] Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 647–653, 2017.

[Wang, 2017] William Yang Wang. " liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 422–426, 2017.