ABSTRACT
In recent years, to mitigate the problem of fake news, computational detection of fake news has been studied, producing some promising early results. While important, however, we argue that a critical missing piece of the study be the explainability of such detection, i.e., why a particular piece of news is detected as fake. In this paper, therefore, we study the explainable detection of fake news. We develop a sentence-comment co-attention sub-network to exploit both news contents and user comments to jointly capture explainable top-\(k\) check-worthy sentences and user comments for fake news detection. We conduct extensive experiments on real-world datasets and demonstrate that the proposed method not only significantly outperforms 7 state-of-the-art fake news detection methods by at least 5.33\% in F1-score, but also (concurrently) identifies top-\(k\) user comments that explain why a news piece is fake, better than baselines by 28.2\% in NDCG and 30.7\% in Precision.

CCS CONCEPTS
• Security and privacy → Social aspects of security and privacy.

KEYWORDS
Fake news; explainable machine learning; social network

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

1 Introduction
Social media platforms provide convenient conduit for users to create, access, and share diverse information. Due to the increased usage and convenience of social media, more people seek out and receive timely news information online. For example, the Pew Research Center announced that approximately 68\% of US adults get news from social media in 2012. However, at the same time, social media enables users to get exposed to a myriad of misinformation and disinformation, including fake news, i.e., news stories with intentionally false information [1, 40]. For example, a report estimated that over 1 million tweets were related to the fake news story “Pizzagate” by the end of 2016 presidential election.

Such widespread of fake news has detrimental societal effects. First, it significantly weakens the public trust in governments and journalism. For example, the reach of fake news during the 2016 U.S. presidential election campaign for top-20 fake news pieces was, ironically, larger than the top-20 most-discussed true stories3. Second, fake news may change the way people respond to legitimate news. A study has shown that people’s trust in mass media has dramatically degraded across different age groups and political parties4. Third, rampant “online” fake news can lead to “offline” societal events. For example, fake news claiming that Barack Obama was injured in an explosion wiped out $130 billion in stock value5. Therefore, it has become critically important to be able to curtail the spread of fake news on social media, promoting trust in the entire news ecosystem.

However, detecting fake news on social media presents unique challenges. First, as fake news is intentionally written to mislead readers, it is non-trivial to detect fake news simply based on its content. Second, social media data is large-scale, multi-modal, mostly user-generated, sometimes anonymous and noisy. Addressing these challenges, recent research advancements aggregate users’ social engagements on news pieces to help infer which articles are fake [13, 37], giving some promising early results. For example, Natali et al. [37] propose a hybrid deep learning framework to model news text, user response, and post source simultaneously for fake news detection. Guo et al. [13] utilize a hierarchical neural network to detect fake news, modeling user engagements with social attention that selects important user comments.

Despite the success of existing deep learning based fake news detection methods, however, the majority of these methods focus on detecting fake news effectively with latent features but cannot explain “why” a piece of news was detected as fake news. Being able to explain why news was determined as fake is much desirable because: (1) the derived explanation can provide new insights and knowledge originally hidden to practitioners; and (2) extracting explainable features from noisy auxiliary information can further

1https://tinyurl.com/ybcy2foa
2https://tinyurl.com/z38z5zh
3https://tinyurl.com/y8dckwhr
4https://tinyurl.com/y9kegobd
5https://tinyurl.com/ybs4tgpg
help improve fake news detection performance. However, to our best knowledge, there has been no prior attempt to computationally detect fake news with proper explanation on social media.

In particular, we propose to derive explanation from the perspectives of news contents and user comments (See Figure 1). First, news contents may contain information that is verifiably false. For example, journalists manually check the claims in news articles on fact-checking websites such as PolitiFact\(^6\), which is usually labor-intensive and time-consuming. Researchers also attempt to use external sources to fact-check the claims in news articles to decide and explain whether a news piece is fake or not [6], which may not be able to check newly emerging events (that has not been fact-checked). Second, user comments have rich information from the crowd on social media, including opinions, stances, and sentiment, that are useful to detect fake news. For example, researchers propose to use social features to select important comments to predict fake news pieces [13]. Moreover, news contents and user comments inherently are related each other and can provide important cues to explain why a given news article is fake or not. For example, in Figure 1, we can see users discuss different aspects of the news in comments such as "St. Nicholas was white? Really??LoL", which directly responds to the claims in the news content "The Holy Book always said Santa Claus was white." Therefore, in this paper, we study the problem of fake news detection by jointly exploring explainable information from news contents and user comments. To this end, we build an explainable fake news detection framework through a coherent process which consists of: (1) a component to encode news contents (to learn the news sentence representations through a hierarchical attention neural network to capture the semantic and syntactic cues), (2) a component to encode user comments (to learn the latent representations of user comments through a word-level attention sub-network), and (3) a sentence-comment co-attention component (to capture the correlation between news contents and comments and to select top-k explainable sentences and comments).

In essence, in this paper, we address the following challenges: (1) How to perform explainable fake news detection that can improve detection performance and explainability simultaneously; (2) How to extract explainable comments without the ground truth during training; and (3) How to model the correlation between news contents and user comments jointly for explainable fake news detection? Our solutions to these challenges result in a novel framework named dEFEND (Explainable Fake News Detection). Our main contributions are summarized as follows:

- We study a novel problem of explainable fake news detection on social media.
- We provide a principled way to exploit both news contents and user comments jointly to capture explainable user comments for fake news detection; and
- We conduct extensive experiments on real-world datasets to demonstrate the effectiveness of dEFEND for detecting fake news and explaining fake news results.

2 RELATED WORK

In this section, we briefly review the related works on fake news detection and explainable machine learning.

2.1 Fake News Detection

Fake news detection methods generally focus on using news contents and social contexts [40, 51, 52]. News content features are mainly extracted from textual and visual aspects. Textual features capture specific writing styles [34] and sensational emotions [12] that commonly occur in fake news contents. In addition, latent textual representations are modeled using tensor factorization [15], deep neural networks [20, 21, 44], which achieve good performance to detect fake news with news contents. Visual features are extracted from visual elements (e.g. images and videos) to capture the different characteristics for fake news [19].

For social context based approaches, the features mainly include user-based, post-based and network-based. User-based features are extracted from user profiles to measure their characteristics [3, 42]. Post-based features represent users’ social response in term of stances [43], topics [13], or credibility [18]. Network-based features are extracted by constructing specific networks, such as the diffusion networks [46], interaction networks [41], and propagation networks [30, 39]. Recently, research also focuses on challenging problems of fake news detection, such as fake news early detection by adversarial learning [45] and user response generating [35], semi-supervised detection [11] and unsupervised detection [15, 49], and explainable detection of fake news through meta attributes [48].

In this paper, we study the novel problem of explainable fake news detection which aims to improve fake news detection performance, and highlight explainable user comments and check-worthy news sentences simultaneously.

2.2 Explainable Machine Learning

Our work is also related to explainable machine learning, which can generally be grouped into two categories: intrinsic explainability and post-hoc explainability [8]. Intrinsic explainability is achieved by constructing self-explanatory models which incorporate explainability directly into their structures. The explainability is achieved by finding the features with large coefficients that play key roles in interpreting the predictions [7]. In contrast, the post-hoc explainability requires to create a second model to provide explanation for an existing model. Koh \etal [24] proposed to identify training points which are most related to a given prediction result through influence functions. Liu \etal propose to interpret network embedding representations via an induction of taxonomy structure [27].
Different from traditional machine learning algorithms, the learned representations of deep learning models (DNNs) are usually not interpretable by human [8]. Therefore, the explanation for deep neural networks (DNNs) mainly focuses on understanding the representations captured by neurons at intermediate layers of DNNs [9, 26, 28]. Liu et al. utilize the interpretation of machine learning models to perform adversarial detection [28]. Du et al. propose to instance-level interpretation of neural networks through guided feature inversion [47]. Karpathy et al. [22] analyzed the interpretability of RNN activation patterns using character level language modeling. Research [33] found that RNN can learn contextual representations by inspecting representations at different hidden layers.

In this paper, we propose to utilize a co-attention mechanism to jointly capture the intrinsic explainability of news sentences and user comments and improve fake news detection performance.

3 PROBLEM STATEMENT
Let A be a news article, consisting of N sentences \( s_1, \ldots, s_N \). Each sentence \( s_i \) contains \( M_i \) words. Let \( C = \{c_1, c_2, \ldots, c_T\} \) be a set of T comments related to the news A, where each comment \( c_j \) contains \( Q_j \) words. Similar to previous research [18, 40], we treat fake news detection problem as the binary classification problem, i.e., each news article can be true \( (y = 1) \) or fake \( (y = 0) \). At the same time, we aim to learn a rank list \( RS \) from all sentences \( \{s_i\}_{i=1}^N \), and a rank list \( RC \) from all comments \( \{c_j\}_{j=1}^T \), according to the degree of explainability, where \( RS_k \) (\( RC_k \)) denotes the \( k \)-th most explainable sentence (comment). The explainability of sentences in news contents represent the degree of how check-worthy they are, while the explainability of comments denote the degree of how much users believe if news is fake or real, closely related to the major claims in news. Formally, we can represent the problem as Explainable Fake News Detection:

**Problem: Explainable Fake News Detection.** Given a news article A and a set of related comments C, learn a fake news detection function \( f : f(A, C) \rightarrow (y, RS, RC) \), such that it maximizes prediction accuracy with explainable sentences and comments ranked highest in RS and RC respectively.

4 DEFEND: EXPLAINABLE FAKE NEWS DETECTION FRAMEWORK
In this section, we present the details of the proposed framework for explainability fake news detection, named as DEFEND (Explainable Fake News Detection). It consists of four major components (see Figure 2): (1) a news content encoder (including word-level and sentence-level) encoder, (2) a user comment encoder, (3) a sentence-comment co-attention component, and (4) a fake news prediction component.

Specifically first, the news content encoder component describes the modeling from the news linguistic features to latent feature space through a hierarchical word- and sentence-level encoding; next, the user comment encoder component illustrates the comment latent feature extraction through word-level attention networks; then, the sentence-comment co-attention component models the mutual influences between the news sentences and user comments for learning feature representations, and the explainability degree of sentences and comments are learned through the attention weights

4.1 News Contents Encoding
As fake news pieces are intentionally created to spread inaccurate information, they often have opinionated and sensational language styles, which have the potential to help detect fake news. In addition, a news document contains linguistic cues with different levels such as word-level and sentence-level, which provide different degrees of importance for the explainability of why the news is fake. For example, in a fake news claim "Pence: Michelle Obama is the most vulgar first lady we’ve ever had", the word "vulgar" contributes more signals to decide whether the news claim is fake rather than other words in the sentence.

Recently, researchers find that hierarchical attention neural networks [50] are very practical and useful to learn document representations [4] with highlighting important words or sentences for classification. It adopts a hierarchical neural network to model word-level and sentence-level representations through self-attention mechanisms. Inspired by [4], we proposed to learn the news content representations through a hierarchical structure. Specifically, we first learn the sentence vectors by using the word encoder with attention and then learn the sentence representations through sentence encoder component.

4.1.1 Word Encoder. We learn the sentence representation via a recurrent neural network (RNN) based word encoder. Though in theory, RNN is able to capture long-term dependencies, in practice, the old memory will fade away as the sequence becomes longer. To capture long-term dependencies of RNN, Gated recurrent units (GRU) [5] are used to ensure a more persistent memory. Similar to [50], we adopt GRU to encode the word sequence. To further
We obtain sentence annotation.

We obtain an annotation of word ward and backward hidden states, i.e., we can use the bidirectional GRU to encode the sentences as follows:

\[ h_i^+ = GRU(w_i), t \in \{1, \ldots, M_i\} \]

\[ h_i^- = GRU(w_i), t \in \{M_i, \ldots, 1\} \]

We obtain an annotation of word \( w_i \) by concatenating the forward hidden state \( h_i^+ \) and backward hidden state \( h_i^- \), i.e., \( h_i^t = [h_i^+, h_i^-] \), which contains the information of the whole sentence centered around \( w_i \). Note that not all words contribute equally to the representation of the sentence meaning. Therefore, we introduce an attention mechanism to learn the weights measuring word importance, and the sentence vector \( v^i \in \mathbb{R}^{2d \times 1} \) is computed as follows,

\[ v^i = \sum_{t=1}^{M_i} a_t^i h_t^+ \]

where \( a_t^i \) measures the importance of \( t^{th} \) word for the sentence \( s_i \), and \( a_t^i \) is calculated as follows,

\[ a_t^i = \frac{\exp(u_t^i u_v^i)}{\sum_{k=1}^{M_i} \exp(u_k^i u_v^i)} \]

where \( a_t^i \) measures the importance of \( t^{th} \) word for the sentence \( s_i \), \( u_t^i \) is a hidden representation of \( h_t^+ \) obtained by feeding the hidden state \( h_t^+ \) to a fully embedding layer, and \( u_v^i \) is the weight parameter that represents the world-level context vector.

### 4.1.2 Sentence Encoder

Similar to word encoder, we utilize RNNs with GRU units to encode each sentence in news. We capture the context information in the sentence-level to learn the sentence representations \( h^i \) from the learned sentence vector \( v^i \). Specifically, we can use the bidirectional GRU to encode the sentences as follows:

\[ h^i = GRU(v^i), i \in \{1, \ldots, N\} \]

\[ h^i = GRU(v^i), i \in \{N, \ldots, 1\} \]

We obtain sentence annotation \( s^i \in \mathbb{R}^{2d \times 1} \) by concatenating the forward and backward hidden states, i.e., \( s^i = [h^+, h^-] \), which captures the context from neighbor sentences around sentence \( s_i \).

### 4.2 User Comments Encoding

People express their emotions or opinions towards fake news through social media posts such as comments, such as skeptical opinions, sensational reactions, etc. These textual information has been shown to be related to the content of original news pieces. Thus, comments may contain useful semantic information that has the potential to help fake news detection. Next, we demonstrate how to encode the comments to learn the latent representations. The comments extracted from social media are usually short text, so we use RNNs to encode the word sequence in comments directly to learn the latent representations of comments. Similar to the word encoder, we adopt bidirectional GRU to model the word sequences in comments.

Specifically, given a comment \( c_j \) with words \( w_j^t, t \in \{1, \ldots, Q_j\} \), we first map each word \( w_j^t \) into the word vector \( w_j^t \in \mathbb{R}^d \) with an embedding matrix. Then, we can obtain the feedforward hidden states \( h_j^+ \) and backward hidden states \( h_j^- \) as follows,

\[ h_j^+ = GRU(w_j^t), t \in \{1, \ldots, Q_j\} \]

\[ h_j^- = GRU(w_j^t), t \in \{Q_j, \ldots, 1\} \]

We further obtain the annotation of word \( w_j^t \) by concatenating \( h_j^+ \) and \( h_j^- \), i.e., \( h_j^t = [h_j^+, h_j^-] \). We also introduce the attention mechanism to learn the weights to measure the importance of each word, and the comment vector \( c_j \in \mathbb{R}^{2d \times 1} \) is computed as follows:

\[ c_j = \sum_{t=1}^{Q_j} \beta_t^j h_j^t \]

where \( \beta_t^j \) measures the importance of \( t^{th} \) word for the comment \( c_j \), and \( \beta_t^j \) is calculated as follows,

\[ u_j^t = \tanh(W_c h_j^t + b_c) \]

\[ \beta_t^j = \frac{\exp(u_j^t u_v^j)}{\sum_{k=1}^{Q_j} \exp(u_j^t u_v^j)} \]

where \( u_j^t \) is a hidden representation of \( h_j^t \) obtained by feeding the hidden state \( h_j \) to a fully embedding layer, and \( u_v \) is the weight.

### 4.3 Sentence-Comment Co-attention

We observe that not all sentences in news contents are fake, and in fact, some sentences are true but only for supporting wrong claim sentences [10]. Thus, news sentences may not be equally important in determining and explaining whether a piece of news is fake. For example, the sentence “Michelle Obama is so vulgar she’s not only being vocal...” is strongly related to the fake claim “Pence: Michelle Obama is The Most Vulgar First Lady We’ve Ever Had”, while “The First Lady denounced the Republican presidential nominee” expresses some fact and is less helpful in detecting and explaining whether the news is fake.

Similarly, user comments may contain relevant information about the important aspects that explain why a piece of news is fake, while they may also be less informative and noisy. For example, a comment “Where did Pence say this? I saw him on CBS this morning and he didn’t say these things...” is more explainable and useful to detect the fake news, than other comments such as “Pence is absolutely right”.

Thus, we aim to select news sentences and user comments that can explain why a piece of news is fake. As they provide a good explanation, they should also be helpful in detecting fake news. This suggests us to design attention mechanisms to give high weights of representations of news sentences and comments that are beneficial to fake news detection. Specifically, we use sentence-comment co-attention because it can capture the semantic affinity of sentences and comments and further help learn the attention weights of sentences and comments simultaneously. We construct the feature matrix of news sentences \( S = [s^1, \ldots, s^N] \in \mathbb{R}^{2d \times N} \) and the feature map of user comments \( C = [c_1^i, \ldots, c_T^i] \in \mathbb{R}^{d \times T} \), the co-attention attends to the sentences and comments simultaneously. Similar
to [29], we first compute the affinity matrix $F \in \mathbb{R}^{T \times N}$ as follows,

$$F = \tanh(C^T W_f S)$$  

where $W_f \in \mathbb{R}^{2d \times 2d}$ is a weight matrix to be learned through the networks. Following the optimization strategy in [29], we can consider the affinity matrix as a feature and learn to predict sentence and comment attention maps as follows,

$$H^s = \tanh(W_s S + (W_c C)F)$$

$$H^c = \tanh(W_c C + (W_s S)F^T)$$  

(9)

where $W_s$, $W_c \in \mathbb{R}^{k \times 2d}$ are the weight parameters. The attention weights of sentences and comments are calculated as follows,

$$a^s = \text{softmax}(w^T_{hs} H^s)$$

$$a^c = \text{softmax}(w^T_{hc} H^c)$$  

(10)

where $a^s \in \mathbb{R}^{1 \times N}$ and $a^c \in \mathbb{R}^{1 \times T}$ are the attention probabilities of each sentence $s^i$ and comment $c^j$, respectively. $w_{hs}$, $w_{hc}$ are the weight parameters. The affinity matrix $F$ transforms user comment attention space to news sentence attention space, and vice versa for $F^T$. Based on the above attention weights, the comment and sentence attention vectors are calculated as the weighted sum of the comment features and sentence features, i.e.,

$$\hat{s} = \sum_{i=1}^{N} a^s_i s^i, \quad \hat{c} = \sum_{j=1}^{T} a^c_j c^j$$

(11)

where $\hat{s} \in \mathbb{R}^{1 \times 2d}$ and $\hat{c} \in \mathbb{R}^{1 \times 2d}$ are the learned features for news sentences and user comments through co-attention.

### 4.4 The Proposed Framework: dEFEND

We have introduced how we can encode news contents by modeling the hierarchical structure from word level and sentence level, how we encode comments by word-level attention networks, and the component to model co-attention to learn sentences and comments representations. We further integrate these components together and predict fake news with the following objective,

$$\hat{y} = \text{softmax}([\hat{s}, \hat{c}]W_f + b_f)$$  

(12)

where $\hat{y} = [\hat{y}_0, \hat{y}_1]$ is the predicted probability vector with $\hat{y}_0$ and $\hat{y}_1$ indicate the predicted probability of label being 0 (real news) and 1 (fake news) respectively, $y \in \{0, 1\}$ denotes the ground truth label of news. $[\hat{s}, \hat{c}]$ means the concatenation of learned features for news sentences and user comments. $b_f \in \mathbb{R}^{1 \times 2d}$ is the bias term. Thus, for each news piece, the goal is to minimize the cross-entropy loss function as follows,

$$L(\theta) = -y \log(\hat{y}_1) - (1-y) \log(1-\hat{y}_0)$$

(13)

where $\theta$ denotes the parameters of the network.

The parameters in the network are learned through RMSprop, which is an adaptive learning rate method which divides the learning rate by an exponentially decaying average of squared gradients. We choose RMSprop as the optimizer because it is a popular and effective method for determining the learning rate abnormally, which is widely used for training neural networks.

### 5 EXPERIMENTS

In this section, we present the experiments to evaluate the effectiveness of the proposed dEFEND framework. Specifically, we aim to answer the following evaluation questions:

- EQ1 Can dEFEND improve fake news classification performance by modeling news contents and user comments simultaneously?
- EQ2 How effective are news contents and user comments, respectively, in improving the detection performance of dEFEND?
- EQ3 Can dEFEND capture the news sentences and user comments that can explain why a piece of news is fake?

### 5.1 Datasets

We utilize one of the comprehensive fake news detection benchmark dataset called FakeNewsNet [38, 40]. The dataset is collected from two platforms with fact-checking: GossipCop and PolitiFact, both containing news content with labels and social context information. News content includes the meta attributes of the news (e.g., body text), and social context includes the related user social engagements of news items (e.g., user comments in Twitter). Note that we keep news pieces with at least 3 comments. The detailed statistics of the datasets are shown in Table 1.

### 5.2 Compared Fake News Detection Methods

The representative state-of-the-art fake news detection algorithms are listed as follows:

- **RST** [36]: RST stands for Rhetorical Structure Theory, which builds a tree structure to represent rhetorical relations among the words in the text. RST can extract news style features by mapping the frequencies of rhetorical relations to a vector space.

- **LIWC** [32]: LIWC stands for Linguistic Inquiry and Word Count, which is widely used to extract the lexicons falling into psycholinguistic categories. It learns a feature vector from psychology and deception perspective.

- **HAN** [50]: HAN utilizes a hierarchical attention neural network framework on news contents for fake news detection. It encodes news contents with word-level attentions on each sentence and sentence-level attentions on each document.

- **text-CNN** [23]: text-CNN utilizes convolutional neural networks to model news contents, which can capture different granularity of text features with multiple convolution filters.

- **TCNN-URG** [35]: TCNN-URG consists of two major components: a two-level convolutional neural network to learn representations from news content, and a conditional variational auto-encoder to capture features from user comments.

- **HPA-BLSTM** [13]: HPA-BLSTM is a neural network model that learns news representation through a hierarchical attention network on word-level, post-level, and sub-event level of user engagements on social media. In addition, post features are extracted to learn the attention weights during post-level.

---

1. The code is available at https://github.com/jiyfeng/DPLP
2. The readers can find more details about the software and feature description at http://liwc.wpengine.com/

---

<table>
<thead>
<tr>
<th>Platform</th>
<th>PolitiFact</th>
<th>GossipCop</th>
</tr>
</thead>
<tbody>
<tr>
<td># Users</td>
<td>68,523</td>
<td>156,467</td>
</tr>
<tr>
<td># Comments</td>
<td>89,999</td>
<td>231,269</td>
</tr>
<tr>
<td># Candidate news</td>
<td>415</td>
<td>5,816</td>
</tr>
<tr>
<td># True news</td>
<td>145</td>
<td>3,586</td>
</tr>
<tr>
<td># Fake news</td>
<td>270</td>
<td>2,230</td>
</tr>
</tbody>
</table>
Table 2: The performance comparison for fake news detection

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Metric</th>
<th>RST</th>
<th>LIWC</th>
<th>text-CNN</th>
<th>HAN</th>
<th>TCNN-URG</th>
<th>HPA-BLSTM</th>
<th>CSI</th>
<th>dEFEND</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolitiFact</td>
<td>Accuracy</td>
<td>0.607</td>
<td>0.769</td>
<td>0.653</td>
<td>0.837</td>
<td>0.712</td>
<td>0.846</td>
<td>0.827</td>
<td>0.904</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.625</td>
<td>0.843</td>
<td>0.678</td>
<td>0.824</td>
<td>0.711</td>
<td>0.894</td>
<td>0.847</td>
<td>0.902</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.523</td>
<td>0.794</td>
<td>0.863</td>
<td>0.896</td>
<td>0.941</td>
<td>0.868</td>
<td>0.897</td>
<td>0.956</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.569</td>
<td>0.818</td>
<td>0.760</td>
<td>0.860</td>
<td>0.810</td>
<td>0.881</td>
<td>0.871</td>
<td>0.928</td>
</tr>
<tr>
<td>GossipCop</td>
<td>Accuracy</td>
<td>0.531</td>
<td>0.736</td>
<td>0.739</td>
<td>0.742</td>
<td>0.736</td>
<td>0.753</td>
<td>0.772</td>
<td>0.808</td>
</tr>
<tr>
<td></td>
<td>Precision</td>
<td>0.534</td>
<td>0.756</td>
<td>0.707</td>
<td>0.655</td>
<td>0.715</td>
<td>0.684</td>
<td>0.732</td>
<td>0.729</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>0.492</td>
<td>0.461</td>
<td>0.477</td>
<td>0.689</td>
<td>0.521</td>
<td>0.662</td>
<td>0.638</td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td>F1</td>
<td>0.512</td>
<td>0.572</td>
<td>0.569</td>
<td>0.672</td>
<td>0.603</td>
<td>0.673</td>
<td>0.682</td>
<td>0.755</td>
</tr>
</tbody>
</table>

- CSI [37]: CSI is a hybrid deep learning model that utilizes information from text, response, and source. The news representation is modeled via an LSTM neural network with the Doc2Vec [25] embedding on the news contents and user comments as input, and for a fair comparison, the user features are ignored.

Note that for a fair comparison, we choose the above fake news methods that extract features from following aspects: (1) only news contents, such as RST, LIWC, text-CNN, HAN; (2) only user comments, such as HPA-BLSTM, and (3) both news contents and user comments, such as TCNN-URG and CSI. For feature extraction methods, we randomly choose 75% of news pieces for training and remaining 25% for testing, and the process is performed for 5 times and the average performance is reported in Table 2. From the table, we make the following observations:

- For news content based methods RST, LIWC and HAN, we see that HAN > LIWC > RST for both datasets. It indicates: 1) HAN can better capture the syntactic and semantic cues through hierarchical attention neural networks in news contents to differentiate fake and real news; 2) LIWC can better capture the linguistic features in news contents. The good results of LIWC demonstrate that fake news pieces are different from real news in terms of choosing the words that reveal psychometrics characteristics.

- In addition, methods using both news contents and user comments perform better than those methods purely based on news contents, and those methods only based on user comments, i.e., dEFEND > HAN or HPA-BLSTM and CSI > HAN or HPA-BLSTM. This indicates that features extracted from news content and corresponding user comments have complementary information, and thus boost the detection performance.

- Moreover, the performance of user comment based methods are slightly better than news content based methods. For example, we have HPA-BLSTM > HAN in terms of Accuracy and F1 on both PolitiFact and GossipCop data. It shows that features extracted from user comments have more discriminative power than those only on news content for predicting fake news.

- Generally, for methods based on both news content and user comments (i.e., dEFEND, CSI, and TCNN-URG), we can see that dEFEND consistently outperforms CSI and TCNN-URG and, i.e., dEFEND > CSI > TCNN-URG, in terms of all evaluation metrics on both datasets. For example, dEFEND achieves average relative improvement of 4.5%, 3.6% on PolitiFact and 4.7%, 10.7% on GossipCop, comparing with CSI in terms of Accuracy and F1 score. It supports the importance of modeling co-attention of news sentences and user comments for fake news detection.

5.4 Assessing Impacts of News Contents and User Comments

Figure 3: Impact analysis of news contents, comments, and sentence-comment co-attention for fake news detection.

In addition to news contents, we also capture information from user comments and integrate it with news contents with co-attention. In order to answer EQ2, we further investigate the effects of these components by defining three variants of dEFEND:

- dEFEND\C: dEFEND\C is a variant of dEFEND without considering information from user comments. It first encodes news contents with word-level attentions on each sentence, and then the resultant sentence features are averaged through an average pooling layer and feed into a softmax layer for classification.

- dEFEND\N: dEFEND\N is a variant of dEFEND without considering information from news contents. It first utilizes the comment encoder to learn comment features, and then the resultant comment features are averaged through an average pooling layer and feed into a softmax layer for classification.

- dEFEND\Co: dEFEND\Co is a variant of dEFEND, which eliminates the sentence-comment co-attention. Instead, it performs self-attention on sentences and comments separately and the
resultant features are concatenated to a dense layer and feed into a softmax layer for classification.

The parameters in all the variants are determined with cross-validation and the best performances are reported in Figure 3. We make the following observations:

- When we eliminate the co-attention for news contents and user comments, the performances are reduced. It suggests the importance of modeling the correlation and captures the mutual influence between news contents and user comments.
- When we eliminate the effect of news contents, the performance of dEFEND\N degrades in comparison with dEFEND. For example, the performance reduces 4.2% and 6.6% in terms of F1 and Accuracy metrics on PolitiFact, 18.2% and 6.8% on GossipCop. The results suggest that news contents in dEFEND are important.
- We have a similar observation for dEFEND\C when eliminating the effect of user comments. The results suggest the importance to consider the feature of user comments to guide fake news detection in dEFEND.

Through the component analysis of dEFEND, we conclude that (1) both components of news contents and user comments can contribute to the fake news detection performance improvement of dEFEND; (2) it is necessary to model both news contents and user comments because they contain complementary information.

5.5 Explainability Evaluation and Case Study

In this subsection, to answer EQ3, we evaluate the performance of explainability of dEFEND framework from the perspective of news sentences and user comments. It is worth mentioning that all of the baseline methods in 5.2 are designed for fake news detection, and none of them are initially proposed to discover explainable news sentences or user comments. To measure the performance of dEFEND for explainability, we choose HAN for comparison of news sentence explainability, and HPA-BLSTM as the baselines for user comments explainability since they can learn attention weights for news sentences and user comments, respectively. Note that HAN uses the attention mechanism to learn the document structure, while HPA-BLSTM utilizes the attention mechanism to learn the temporal structure of comments. Since there is no temporal structure in documents, so HAN cannot be used in comments; Similarly, there are no temporal relations in the document structure, so HPA-BLSTM cannot be directly applied to news contents.

5.5.1 News Sentence Explainability. In this subsection, we demonstrate the performance of the explainability rank list of news sentences, i.e., RS. Specifically, we want to see if the top-ranked explainable sentences determined by our method are more likely to be related to the major claims in fake news that are worth to check–i.e., check-worthy. Therefore, we utilize ClaimBuster [14] to obtain a ground truth rank list RS of all check-worthy sentences in a piece of news content. ClaimBuster proposes a scoring model that utilizes various linguistics features trained using tens of thousands of sentences from past general election debates that were labeled by human coders and gives a “check-worthiness” score between 0 and 1. The higher the score, the more likely the sentence contains check-worthy factual claims. The lower the score, the more non-factual, subjective and opinionated the sentence is. We compare top-k rank list of the explainable sentences in news contents by dEFEND (RS\1) and HAN (RS\2), with top-k rank list, RS, by ClaimBuster, using the evaluation metric, MAP@k (Mean Average Precision), where k is set as 5 and 10. We also introduce another parameter n which controls the window size that allows n neighboring sentences are considered when comparing the sentences in RS\1 and RS\2 with each of the top-k sentences in RS. From Figure 4, we make the following observations:

- In general, we can see that dEFEND > HAN > Random for the performance of finding check-worthy sentences in news contents on both datasets. It indicates that the sentence-comment co-attention component in dEFEND can help selecting more check-worthy sentences.
- With the increase of n, we relax the condition to match check-worthy sentences in the ground truth, and thus the MAP performance is increasing.
- When n = 1, the performance of dEFEND on MAP@5 and MAP@10 increases to exceed 0.8 for PolitiFact, which indicates that dEFEND can detect check-worthy sentences well within 1 neighboring sentence of the ground truth sentences in RS.

5.5.2 User Comments Explainability. We deploy several tasks using Amazon Mechanical Turk (AMT)\footnote{https://www.mturk.com/} to evaluate the explainability rank list of the comments RC for fake news. We perform the following settings to deploy AMT tasks for a total of 50 fake news pieces. For each news article, we first filter out very short articles with less than 50 words. In addition, for very long articles with more than 500 words in content, we presented only the first 500 words to reduce the amount of reading for workers. As the first 3-4 paragraphs of news articles often summarize the content, the first 500 words are usually sufficient to capture the gist of the articles. Then, we recruited AMT workers located in the US (who are more likely to be familiar with the topics of the articles) with the approval rate > 0.95. To evaluate the explainability of user comments, for each news article, we have two lists of top-k comments, \(C = (C_1, C_2, \cdots, C_k)\) for using...
Figure 5: The human-evaluation of explainable comment list of dEFEND and HPA-BLSTM with Task 1.

Figure 6: The discrepancy histograms of mean NDCG and mean Precision@5 of the results between two methods. dEFEND and $L^{(2)} = (L_1^{(2)}, L_2^{(2)}, \ldots, L_k^{(2)})$ for HPA-BLSTM. The top-$k$ comments are selected and ranked using the attention weights from the high to low. To evaluate the model ability to select topmost explainable comments, we empirically set $k = 5$. We deploy two AMT tasks to evaluate the explainable ranking performance.

For Task 1, we perform list-wise comparison. We ask workers to pick a collectively better list between $L^{(1)}$ and $L^{(2)}$. To remove the position bias, we randomly assign the position, top and bottom, of $L^{(1)}$ and $L^{(2)}$ when presented to workers. We let each worker pick the better list between $L^{(1)}$ and $L^{(2)}$ for each news piece. We ensure each news piece is evaluated by 3 workers, and finally obtained 150 results of workers’ choices. In a worker-level, we compute the number of workers that choose $L^{(1)}$ and $L^{(2)}$, and also compute the winning ratio (WR for short) for them. In a news-level, we perform majority voting for all 3 workers for each news and decide if workers choose $L^{(1)}$ or $L^{(2)}$. For each news, we also compute the worker-level choices by computing the ratio between $L^{(1)}$ and $L^{(2)}$. From Figure 5, we make the following observations:

- **dEFEND** can select better top-$k$ explainable comments than HPA-BLSTM both in worker-level and news-level. First, in worker-level, 98 out of 150 workers (with WR=0.65) choose $L^{(1)}$ over $L^{(2)}$. Second, in news-level, dEFEND has better performance in 32 out of 50 news pieces (with WR=0.64) than HPA-BLSTM.

- We can see that there are more news pieces such that 3 workers vote unanimously for $L^{(1)}$ (3 vs 0) than the opposite case (0 vs 3) for their explainability i.e., 14 vs 7. Similarly, there are more cases where 2 workers vote for dEFEND than HPA-BLSTM, i.e., 18 vs 11.

For Task 2, we perform item-wise evaluation. For each comment in $L^{(1)}$ and $L^{(2)}$, we ask workers to choose a score from 0, 1, 2, 3, 4, where 0 means “not explainable at all,” 1 means “not explainable,” 3 means “somewhat explainable,” 4 means “highly explainable,” and 2 means “somewhere in between.” To avoid the bias caused by different user criteria, we shuffle the order of comments in $L^{(1)}$ and $L^{(2)}$, and ask workers to assess how explainable each comment is with respect to the news. To estimate rank-aware explainability of comments (i.e., having a higher ranked explainable comment is more desirable than a lower ranked one), we use NDCG (Normalized Cumulative Gain) [16] and Precision@$k$ as the evaluation metrics. NDCG is widely used in information retrieval to measure document ranking performance in search engines. It can measure how good a ranking is by comparing the proposed ranking with the ideal ranking list measured by user feedback. Precision@$k$ is the proportion of recommended items in a top-$k$ set that are relevant. Similarly, We ensure each news piece is evaluated by 3 workers and obtain a total of 750 results of workers’ ratings for each method.

Figure 7: The explainable comments captured by dEFEND.
6 CONCLUSION AND FUTURE WORK
Fake news detection is attracting growing attention in recent years. However, it is also important to understand why a piece of news is detected as fake. We study the novel problem of explainable fake news detection which aims to: 1) improve detection performance significantly; and 2) discover explainable news sentences and user comments to understand why news pieces are identified as fake. We propose a deep hierarchical co-attention network to learn feature representations for fake news detection and explainable sentences/comments discovery. Experiments on real-world datasets demonstrate the effectiveness of the proposed framework. For future work, first, we can incorporate the fact-checking contents from journalist experts or fact-checking websites to further guide the learning process to obtain check-worthy news sentences. Second, we will explore how to use other user engagements as side information such as likes to help discover explainable comments. Third, we can consider the credibility of the users who posts explainable comments to further improve fake news detection performance.

7 ACKNOWLEDGMENTS
This material is in part supported by the NSF awards #1614576, #1742702, #1820609, and #1915801, ONR grant N00014-17-1-2605 and N000141812108, and ORAU-directed R&D program in 2018.

REFERENCES
8 APPENDIX ON REPRODUCIBILITY

In this section, we provide more details of the experimental setting and configuration to enable the reproducibility of our work.

8.1 Fake News Detection

We compared the proposed framework, dEFEND, with 7 baseline methods discussed in Section 5.2, including RST, LIWC, text-CNN, HAN, TCNN-URG, HPA-BLSTM, and CSI. All codes that we have implemented are available under the folder “Fake new detection” through the following link: https://tinyurl.com/ybl6gqrm. Other codes were obtained as follows:

- RST: we used the publicly available implementation for paper [17]: https://github.com/jyfeng/DPLP
- LIWC: we used the publicly available tool at: http://liwc.wpengine.com/
- text-CNN: we used the publicly available implementation at: https://github.com/dennybritz/cnn-text-classification-tf
- HAN: we used the publicly available implementation at: https://github.com/richliao/textClassifier
- TCNN-URG: we implemented this algorithm based on the description in the paper [35], and shared the code, named as tcnn.py, in the above link
- HPA-BLSTM: we used the implementation provided by the authors of [13]
- CSI: we used the implementation available at: https://github.com/sungyongs/CSI_Code
- dEFEND: we implemented our algorithm in Python–defend.py for main algorithm and go_defend.py for data processing–and shared the code, named as tcnn.py, in the above link.

For the dataset, we also used a publicly available dataset, Fake-NewsNet [38], available at: https://github.com/KaiDMML/FakeNewsNet. For parameter settings for dEFEND, we introduce the details of major parameter setting as shown in Table 3. The descriptions of the major parameters are as follows:

- MAX_SENTENCE_LENGTH: the threshold to control the maximum length of news sentences
- MAX_SENTENCE_COUNT: the threshold to control the maximum count of sentences
- MAX_COMMENT_LENGTH: the threshold to control the maximum length of user comments
- MAX_COMMENT_COUNT: the threshold to control the maximum count of user comments
- Vocabulary Size: the threshold to control the maximum size of vocabulary
- Embedding Dimension: the dimension of embedding layer
- Word Embedding: the word embedding package used for initializing the word vectors
- $d$: the size of hidden states for BLSTM
- $k$: the size of attention maps as in Eqn. 9

<table>
<thead>
<tr>
<th>Parameter</th>
<th>PolitiFact</th>
<th>GossipCop</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX_SENTENCE_LENGTH</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>MAX_SENTENCE_COUNT</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>MAX_COMMENT_COUNT</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>MAX_COMMENT_LENGTH</td>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>Word Embedding</td>
<td>Glove10</td>
<td>Glove11</td>
</tr>
<tr>
<td>Embedding Dimension</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$d$</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>$k$</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>Batch Size</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Maximum Epochs</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Vocabulary Size</td>
<td>20,000</td>
<td>20,000</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>0.01</td>
<td>0.001</td>
</tr>
<tr>
<td>RMSprop parameter ($\rho$)</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>RMSprop parameter ($\epsilon$)</td>
<td>1e-8</td>
<td>1e-8</td>
</tr>
<tr>
<td>RMSprop parameter (decay)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

8.2.2 User Comments. We introduce the details of the two tasks we deployed at Amazon Mechanical Turk.

- **Task 1.** We presented each fake article with two lists of top-$k$ comments identified by dEFEND and HPA-LSTM, and let workers choose the list that can “collectively” explain better why this is fake news. In order to remove the position bias, we shuffled the order of the two lists randomly, between top and bottom. Each Human Intelligence Task (HIT) contains five fake articles, and was assigned to three distinct workers. The HIT screen-shot for Task 1 is shown in Figure 8. To ensure the quality of crowdsourcing task, we set two requirements for AMT workers: (1) the approved percentage of assignments of a worker should be greater than 95%; and (2) the location of a worker should be in US.
- **Task 2.** We tested the explainability of user comments detected by dEFEND and HPA-LSTM, respectively. We presented each fake news article with a list of “mixed” comments identified by two methods, and let workers to assign a score of 0-4 to each comment, where 0 means “not explainable at all,” 1 means “not much explainable,” 3 means “explainable a bit,” 4 means “highly explainable,” and 2 means “somewhere in between.” Note that in order to remove the position bias, we shuffled the order of the comments randomly. Further, to ensure the quality of the task, we applied the same two requirements as in Task 1. Each HIT had one fake article, and was assigned to 3 distinct workers. The HIT screen-shot of Task 2 is shown in Figure 9.

8.2 Explainability on News and Comments

We elaborate further details on how we evaluated the explainability of sentences and comments in experiments.

8.2.1 News Sentences. We obtained the ground truth of check-worthy sentences from the online tool, ClaimBuster, with its default setting. ClaimBuster is available at: https://idir-server2.uta.edu/claimbuster/.
Choosing more explainable comments

Fake news is the news that is intentionally created to spread false information. More about fake news on Wikipedia page.

In this task, you need to read the news article and several user comments on Twitter related to the news and choose the comments that can explain why the news is fake or not.

We provide the first 500 words of the article to give you the sufficient sense of the major topic of the article, and short to read the entire long news report.

Note that we provide bonus reward to workers if they provide high-quality answers competing with other workers in this task.

Article ID: politico/1584
Article Title: International Arrest Warrant Issued for George Soros
Article Content: George Soros, the billionaire investor, has been linked to manipulating the financial markets in Asia, the UK, Greece, and Russia. This is done to create a financial crisis in the United States, which is known as the Soros Plan. The Soros Plan is designed to create a financial crisis in the United States by buying up stocks and bonds that are used to finance the country. This plan is designed to create a financial crisis in the United States by buying up stocks and bonds that are used to finance the country. The Soros Plan is designed to create a financial crisis in the United States by buying up stocks and bonds that are used to finance the country.

Rate the Explainability (0-4) of Comments for Fake News Articles 1

Fake news is the news that is intentionally created to spread false information. More about fake news on Wikipedia page.

In this task, you need to read the article body text and several user comments on Twitter related to the news and choose a score of 0-4 for each comment to show how much it can explain why the news is fake or not.

We provide the first 500 words of the article to give you the sufficient sense of the major topic of the article, and short to read the entire long news report.

This is a joint research project.

Note that we provide bonus reward to workers if they provide high-quality answers competing with other workers in this task.

Article ID: politico/1584
Article Title: International Arrest Warrant Issued for George Soros
Article Content: George Soros, the billionaire investor, has been linked to manipulating the financial markets in Asia, the UK, Greece, and Russia. This is done to create a financial crisis in the United States, which is known as the Soros Plan. The Soros Plan is designed to create a financial crisis in the United States by buying up stocks and bonds that are used to finance the country. This plan is designed to create a financial crisis in the United States by buying up stocks and bonds that are used to finance the country.

Figure 8: Task 1: Choosing collectively more explainable user comments for fake news articles.

Figure 9: Task 2: Rating the explainability (0-4) of user comments for fake news articles.