Towards Fine-Grained Reasoning for Fake News Detection

Yiqiao Jin\textsuperscript{1}, Xiting Wang\textsuperscript{2}, Ruichao Yang\textsuperscript{3}, Yizhou Sun\textsuperscript{1}, Wei Wang\textsuperscript{1}, Hao Liao\textsuperscript{4}, Xing Xie\textsuperscript{2}

\textsuperscript{1} University of California, Los Angeles, \textsuperscript{2} Microsoft Research Asia, \textsuperscript{3} Hong Kong Baptist University, \textsuperscript{4} Shenzhen University

ahren2040@g.ucla.edu, {xitian, xing.xie}@microsoft.com, {yzsun, weiwang}@cs.ucla.edu, csrcyang@comp.hkbu.edu.hk, haoliao@szu.edu.cn

Abstract

The detection of fake news often requires sophisticated reasoning skills, such as logically combining information by considering word-level subtle clues. In this paper, we move towards fine-grained reasoning for fake news detection by better reflecting the logical processes of human thinking and enabling the modeling of subtle clues. In particular, we propose a fine-grained reasoning framework by following the human’s information-processing model, introduce a mutual-reinforcement-based method for incorporating human knowledge about which evidence is more important, and design a prior-aware bi-channel kernel graph network to model subtle differences between pieces of evidence. Extensive experiments show that our model outperforms the state-of-the-art methods and demonstrate the explainability of our approach.

Introduction

The emergence of social media has transformed the way users exchange information online. People are no longer mere reviewers of information, but content creators and message spreaders. Consequently, it has become much easier for fake news to spread on the Internet. Since fake news can obscure the truth, undermine people’s belief, and cause serious social impact (Brewer, Young, and Morreale 2013), detecting fake news has become increasingly important for a healthy and clean network environment (Shu et al. 2017).

Recently, neural models have been proposed to detect fake news in a data-driven manner (Pan et al. 2018; Dun et al. 2021). These works have shown the promise in leveraging big data for fake news detection. However, works that approach the task from the perspective of reasoning are still lacking. According to the literature on psychology, reasoning is the capability of consciously applying logic for truth seeking (Honderich 2005), and is typically considered as a distinguishing capacity of humans (Mérouard and Sperber 2017). We observe that such ability is essential to improve the explainability and accuracy for fake news detection:

Explainability. Most existing works on fake news detection either do not provide explanations or enable explainability for a small part of the model (e.g., the attention layer).

Explainability by 1) better reflecting the logical processes of human thinking and 2) enabling fine-grained modeling of subtle clues. In particular, we study three research questions:

\begin{itemize}
  \item \textbf{Evidence group 1: imply TRUE news}
  \item \textbf{Evidence group 2: imply TRUE news}
  \item \textbf{Evidence group 3: imply FAKE news}
  \item \textbf{Evidence group 4: imply FAKE news}
\end{itemize}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig.png}
\caption{A motivating example of fine-grained reasoning for fake news detection.}
\end{figure}

The major part of the model (e.g. the overall workflow) remains obscure to humans. This prevents humans to better understand and trust the model, or steer the model for performance refinement.

\textbf{Accuracy.} When humans reason about the authenticity of a news article, they have the ability to perform \textit{fine-grained} analysis for identifying subtle (e.g., word-level) clues, and can connect different types of clues (e.g., textual and social) to draw conclusions. An example is shown in Fig. 1. Although the four groups of evidence are semantically dissimilar, humans can logically connect them in terms of subtle clues such as the word “property”, which leads to a much more confident conclusion about the article. For example, reasoning in terms of “property” reveals that the accusation of finding bodies in Clintons’ property (evidence group 1) might be false, since the women were dead before the Clintons bought the property (evidence group 3). Reasoning with respect to “hate” suggests that users in groups 1 and 2 might post false messages because they hate the Clintons. The overlap between users in groups 1 and 2 further strengthens this suggestion. Existing methods lack such capability of fine-grained reasoning; they either do not model the interactions between different types of evidence or model them at a coarse-grained (e.g., sentence or post) level.

We aim to move towards using deep reasoning for fake news detection. The goal is to improve accuracy and explainability by 1) better reflecting the logical processes of human thinking and 2) enabling fine-grained modeling of subtle clues. In particular, we study three research questions:
**Methodology**

**Problem Definition**

**Input.** The inputs of our model are threefold. Each training sample consists of 1) a news article $S$ to be verified; 2) a collection of online posts $P$ for the news and the corresponding/retweeting relationships between the posts; and 3) the online users $U$ that publish the posts $P$.

**Output.** The output of our model is the predicted label of the news, which can be fake ($y = 1$) or real ($y = 0$).

**Fine-Grained Reasoning Framework**

We propose a *Fine-grained reasoning framework for Fake news detection* (FinerFact) by following the human’s information-processing model (Lang 2000). Our framework consists of two major modules as shown in Fig. 2.

The first module, *claim-evidence graph construction*, corresponds to the *storage* sub-process of the human’s information-processing model, in which people select the most important pieces of information and build their in-between associations to store them in the memory. As of fake news detection, it corresponds to the process in which people search for key information such as the major viewpoints, opinion leaders, and the most important posts, which enables them to get an idea about the key claims and their associated evidence (e.g., supported posts and users). This step is essential for filtering noise, organizing facts, and speeding up the fine-grained reasoning process at the later stage. It also enables us to incorporate human knowledge about which information is important.

The second module, *graph-based fine-grained reasoning*, corresponds to the *retrieval* sub-process of the human’s information-processing model, in which people reactivate specific pieces of information based on their associations for decision making. In fake news detection, this module enables fine-grained modeling of evidential relations by considering subtle clues, such as the word “property”, “hate”, and the overlapping user in Fig. 1.

By designing the framework based on the human’s information-processing model, the overall workflow of our method resembles the logical processes used by humans, and most of the intermediate results are understandable by ordinary users. This provides a good basis for users to trust the model and steer it by integrating human knowledge.

**Claim-Evidence Graph Construction**

Our graph construction method contains two parts: 1) *mutual-reinforcement-based evidence ranking*, which distinguishes important evidence from noise by incorporating human knowledge (filtering noise); and 2) *topic-aware claim-evidence association*, which follows journalists’ knowledge about quality journalism (Project for Excellence in Journalism 2005) to extract the key claims and associate them with the corresponding evidence (organizing facts).

**Mutual-Reinforcement-Based Evidence Ranking**

As a news article propagates on the Internet, it leaves many traces, e.g., posts about the news and users who support the news, which can all be considered as evidence for verifying the news. Such evidence typically has a large scale, and performing fine-grained reasoning by considering all evidence is quite difficult, if not impossible, due to limited GPU memory. To efficiently and accurately identify the most valuable evidence in an interpretable way, we propose ranking the evidence by integrating human knowledge. The
human knowledge can be divided into two types based on whether it considers the inherent attributes (Lampos et al. 2014) of the evidence or its topological (Pasquinielli 2009) information. We observe that these two types of knowledge can be integrated in one framework and computed efficiently by using the mutual reinforcement mechanism (Duan et al. 2012). Specifically, our mutual-reinforcement-based evidence ranking consists of the following three steps.

Step 1: Attribute saliency computation. We compute the attribute saliency $E$ based on the human knowledge summarized from the current literature. In particular, the attribute saliency $e_{u_i}$ for a user $u_i$ is computed by using the user impact index (Lampos et al. 2014):

$$e_{u_i} = \ln \left( \frac{(\phi_i^f + \theta_U)^2(\phi_i^l + \theta_U)}{\phi_i^l + \theta_U} \right)$$

(1)

where $\phi_i^f$, $\phi_i^l$, and $\theta$ are $u_i$’s follower count, friend count, and listed count, and $\theta$ is a smoothing constant added to ensure that the minimum saliency score is always positive. The attribute saliency $e_{p_i}$ of each post $p_i$ is computed based on its number of retweets $R$, considering that retweets rarely provide extra textual information and are usually regarded as votes to the original post (Chang et al. 2013). Specifically, $e_{p_i} = \ln (C_i + 1) + \theta_P$, where $\theta$ is a smoothing constant. The attribute saliency $e_{k_i}$ of each keyword $k_i$ is set to $\ln (\text{freq}(k_i) + 1) + \theta_K$, where $\text{freq}(k_i)$ is the term frequency of $k_i$ with respect to the news article and the posts, and $\theta$ is a smoothing constant (Rousseau and Vazirgiannis 2013).

Step 2: Mutual reinforcement evidence graph building. As shown in Fig. 2, we build the mutual reinforcement evidence graph $M$ so that it encodes both the relations within posts, users, or keywords, and the relations between them to enable the effective modeling of topological saliency. Mathematically, $M = \{A_{xy}, x, y \in \{P, U, K\}\}$ is a three-layer graph, where subscript indices $P, U, K$ denote posts, users, and keywords, respectively, and $A_{xy}$ is an affinity matrix that represents relations between items. We design the graph based on two considerations: 1) $M$ should effectively encode diverse types of social interactions and 2) edges in $M$ should reveal the mutual influence between items in terms of saliency. For example, constructing edges between users and their posts means that if a user is important, then his/her posts are important, and vice versa. Such human knowledge about which items should influence each other in terms of saliency can be effectively incorporated into the edge design. Based on these considerations, we construct $M$ by using the cosine similarity between the term frequency vectors of the posts ($A_{PP}$), the commenting relationships between users ($A_{UU}$), the co-occurrence relationships between keywords ($A_{KK}$), the mentioning relationships that link a keyword to all the posts and users that mention it ($A_{PK}$, $A_{UK}$), and the authoring relationships that link a user to all the posts that s/he has published ($A_{UP}$).

More details about the construction of $M$ and how human knowledge is used are given in the supplement.

Step 3: Iterative saliency propagation. We then compute the saliency $R$ based on the mutual reinforcement mechanism (Duan et al. 2012). In particular, we treat the attribute saliency $E$ as a prior and integrate it with the saliency propagation process on $M$:

$$R^{(i+1)} = dAR^{(i)} + (1 - d)E$$

(2)

$$\hat{A} = \begin{bmatrix} \beta_{PP}A_{PP} & \beta_{PK}A_{PK} & \beta_{PU}A_{PU} \\ \beta_{KP}A_{PK} & \beta_{KK}A_{KK} & \beta_{UK}A_{UK} \\ \beta_{PU}A_{UP} & \beta_{UK}A_{UK} & \beta_{UU}A_{UU} \end{bmatrix}, R^{(i)} = \begin{bmatrix} R_P^{(i)} \\ R_K^{(i)} \\ R_U^{(i)} \end{bmatrix}, E = \begin{bmatrix} E_P \\ E_K \\ E_U \end{bmatrix}$$

where $R^{(i)}$ is the joint ranking score vector in the $i$-th iteration, $\hat{A}$ is the normalized affinity matrix derived from $A$, and $\beta_{xy}$ is a balancing weight to adjust the interaction strength among posts, keywords, and users.

Interpretability and efficiency. The designed method is highly interpretable and steerable, since each type of saliency can be easily explained to and controlled by human users. According to Eq. (2), a piece of evidence (e.g., a supported user) is important if it has a large attribute saliency (e.g., has many followers) and is connected with other salient evidence (e.g., writes a salient post). Other types of human knowledge can also be easily integrated by slightly modifying the equation. Our method is also efficient. In practice, ranking 240,000 posts, users, and keywords takes 620 seconds. Without evidence ranking, performing fine-grained reasoning on the same data causes the out of memory issue on NVIDIA Tesla V100.

Topic-Aware Claim-Evidence Association Given the saliency scores $R$, a straightforward way for constructing the claim-evidence graph is to select the pieces of evidence with the largest saliency scores. However, this method may easily obscure the truth by focusing only on one aspect of the story. For example, the news article in Fig. 1 may be dominated by posts related to evidence group 1, which are posted deliberately by users who hate the Clintons. To disclose the truth, we need to observe all four evidence groups closely. This echoes the journalists’ knowledge about quality journalism (Project for Excellence in Journalism 2005), which states that a high-quality news article should cover multiple viewpoints to reveal two or more sides of the story. Motivated by this insight, we propose a topic-aware method, which consists of the following steps.

Step 1: Topic modeling. A typical solution to mine major viewpoints in a text corpus is topic modeling. In the scenario of fake news detection, the text corpus consists of each post and news sentence. We then utilize LDA (Blei, Ng, and Jordan 2003) to mine the topics, which summarizes the main viewpoints and serve as a bridge between the claims in the news sentences and the keywords in the evidence graph $M$:

- Each topic $t$ is represented by a distribution of keywords $P(k_i|t)$ in $M$. For each topic $t$, its top $N_K$ keywords $K^t$ are the ones that have the largest $P(k_i|t)$.
- Each news sentence $s_i$ is represented by a distribution of topics $P(t|s_i)$. Given a topic $t$, we can extract its top $N_S$ sentences $S^t$ that have the largest $P(t|s_j)$.

Step 2: Key claim and evidence extraction. To extract the key claims and their associated evidence based on the topics, we first select the top $N_T$ topics with the maximum aggregate saliency score $r_t = \sum_{k_i \in K^t} P(k_i|t)\gamma_{k_i}$, where $K^t$ is the set of top keywords extracted in step 1, and $\gamma_{k_i}$ is the saliency score computed by using mutual-reinforcement-based evidence ranking. The topics that are selected in this
way not only cover major viewpoints of the article, but are also related with the most salient evidence in $M$.

For each selected topic $t$, its corresponding claims are the top news sentences $S^t$ extracted in step 1. The key evidence consists of two parts. The first part is the key posts in $t$. Given a set of posts that are connected to the keyword set $K^t$, we normalize the saliency score of each post in the set to obtain a probability distribution, and then sample a set $P^t$ with $N_P$ posts according to the distribution. Similarly, we sample a set $U^t$ with $N_U$ users that are relevant with $t$, and treat $U^t$ as the second part of the evidence. Keywords are not considered as evidence because they are less useful for fine-grained reasoning when considered without the context. To model keywords more effectively, we treat the posts as ordered sequences of keywords in fine-grained reasoning.

**Step 3: Graph construction.** Finally, we build a claim-evidence graph $G$ as shown in Fig. 2. In $G$, each node $v$ is a tuple $(t, S^t, P^t, U^t)$ that corresponds to a selected topic $t$, where $S^t$ refers to the key claims and $P^t, U^t$ form an evidence group shown in Fig. 1. Given all nodes $V$, a straightforward approach to construct $G$ is to build edges between two nodes if the percentage of overlapping words is larger than a threshold (Zhong et al. 2020). However, this method may easily overlook important subtle clues, because 1) it is difficult to find an appropriate global threshold and 2) topics (or evidence groups) may be connected logically with different words (e.g., ‘damn’, ‘hate’, and ‘hell’ in Fig. 1). Based on this observation, we choose to build a fully connected graph and let the fine-grained reasoning module to decide whether subtle clues exist for a set of topics.

**Fine-Grained Graph-based Reasoning**

After constructing the claim-evidence graph $G$, we model subtle clues and effectively leverage them for prediction through fine-grained graph-based reasoning. Our method is based on the Kernel Graph Attention Network (KGAT) (Liu et al. 2020). We choose this method because it can effectively model subtle differences between statements and propagate the learned information on the graph. However, KGAT cannot be directly applied on our claim-evidence graph, because it handles only textual inputs and cannot integrate the learned saliency $R$, which incorporates human knowledge about which evidence is important. To solve these issues, we propose a Prior-Aware Bi-Channel KGAT that extends KGAT to 1) simultaneously model subtle clues from both textual (posts) and social (user) inputs with two connected channels; and 2) integrating existing knowledge about important evidence with attention priors. Mathematically, the final prediction $P(y|G, R)$ is obtained by combining individual node-level predictions:

$$P(y|G, R) = \sum_{v \in G} P(y|v, G) P(v|G, R) \tag{3}$$

This formulation provides explainability for individual prediction scores that each node (or claim-evidence group) gives and importance of the nodes for the final prediction.

**Node label prediction with bi-channel kernel matching**

which accurately computes $P(y|v, G)$ by integrating different types of subtle clues from the whole graph:

**Node importance learning with attention priors**, which effectively models $P(v|G, R)$ by integrating the evidence saliency $R$ as attention priors.

**Node Label Prediction with Bi-Channel Kernel Matching**

Given a node $v$, we predict the individual label it gives by aggregating subtle clues from the whole graph. To reason about subtle clues that are provided by both the textual and social (user) inputs, we design two interconnected channels. We will first introduce how each channel is designed when they are modeled independently, and then illustrate how the channels can be fused for prediction.

**Text-based reasoning channel.** We first derive an initial textual representation for node $v = (t, S^t, P^t, U^t)$ by concatenating the claims in $S^t$ and evidential posts in $P^t$ with token ‘[SEP]’. The concatenated string is then encoded by using BERT (Devlin et al. 2019):

$$[h_i^0, h_i^1, \ldots, h_i^{N_v}] = \text{BERT}(S^t \oplus P^t), \quad z_v = h_i^0 \quad \tag{4}$$

where $h_i^0$ denotes the BERT embedding for the $i$-th token, $z_v$, which corresponds to the embedding of the ‘[CLS]’ token, is considered as the initial textual representation for $v$.

The fine-grained match features between nodes $v$ and $q$ can then be extracted by constructing a token-level transition matrix $L_{q,v}$. In $L_{q,v}$, each entry $L_{q,v}^{i,j} = \cos(h_i^j, h_i^q)$ is the cosine similarity between their token representations. For each token $i$ in node $q$, we use $\Upsilon$ kernels to extract the kernel match features between the token and its neighbor $v$:

$$\Psi_r(L_{q,v}^i) = \log \sum_j \exp(-\frac{(\mu_r - \mu_v)^2}{2\sigma_r^2}) \quad \tag{5}$$

$$\hat{\Psi}(L_{q,v}^i) = \{\Psi_1(L_{q,v}^i), \ldots, \Psi_T(L_{q,v}^i)\} \quad \tag{6}$$

Each $\Psi_r$ is a Gaussian kernel that concentrates on the region defined by the mean similarity $\mu_r$ and the standard deviation $\sigma_r$. The kernel match feature $\hat{\Psi}$, which consists of $\Upsilon$ kernels, summarizes how similar the $i$-th token in $q$ is to all tokens in $v$ at different levels. Such soft-TF match features have been shown effective for fact verification (Liu et al. 2020). In our framework, they help identify subtle clues by comparing different claim-evidence groups (nodes). For example, in Fig. 1, the match features can help identify “hate” in group 4 (node $q$) by comparing it with all words in group 2 (node $v$). This neighbor-aware token selection is achieved by computing an attention score based on $\Psi$:

$$\alpha_{q,v}^i = \text{softmax}_i(W_1 \hat{\Psi}(L_{q,v}^i) + b_1) \quad \tag{7}$$

We can then compute the content to propagate from $q$ to $v$:

$$\hat{z}_{q,v} = \sum_i \alpha_{q,v}^i h_i^q \quad \tag{8}$$

Note that by setting $\sigma_r$ to $\infty$, Eq. (8) degenerates to mean pooling, which assigns an equal weight to all tokens.

Given $\hat{z}_{q,v}$ that contains the information to be propagated to $v$, we derive a final textual representation $\kappa_v$ for $v$ by attentively aggregating $\hat{z}_{q,v}$ from all $q \in G$:

$$\kappa_v = \sum_{q \in G} \gamma_{q,v} (\hat{z}_{q,v}) + z_v \quad \tag{9}$$

$$\gamma_{q,v} = \text{softmax}_q(\text{MLP}(\hat{z}_{q,v} + z_v)) \quad \tag{10}$$
The kernel-based textual representation $\kappa_v$ aggregates fine-grained, token-level subtle clues from the whole graph, and can be used to reason about the authenticity of the news from the perspective of node $v$. Next, we introduce how to derive kernel-based user representations with a user channel, and how these two channels can be fused for final prediction.

**User-based reasoning channel.** The initial user representation $x_v$ for node $v = (t, S^t, P^t, U^t)$ is derived by applying a graph neural network, APPNP (Klicpera, Bojchevski, and Günnemann 2018), on the mutual reinforcement evidence graph $M$. We choose APPNP because its efficient message-passing scheme enables it to scale to large graphs with hundreds of thousands of nodes. Specifically, for each user in $U^t$, we first obtain its feature embedding by using a look up layer, which encodes the main user attributes including the user’s follower count, friend count, listed count, favourite count, status count, the number of words in the self description, as well as the account status about whether the user is verified or geo-enabled. We then use the message-passing scheme of APPNP to aggregate the feature embeddings from the neighbor users in $M$. This results in an initial user representation $u^0_v$ for each user, and max pooling is used to derive the initial user presentation $x_v$ for node $v$:

$$[u^0_v, ..., u^N_v] = \text{APPNP}(M), \quad x_v = \text{maxpool}(u^0_v, ..., u^N_v) \quad (11)$$

We can then derive the kernel-based user representation $\kappa_v$ by using a kernel attention mechanism similar to that in the text-based reasoning channel:

$$\rho_{q,v} = \text{softmax}_q(W_2 \tilde{\Psi}(t, v, u^q_v) + b_2) \quad (12)$$

$$\xi_q,v = \sum_i \rho_{q,v} u^i_v \quad (13)$$

$$\lambda_{q,v} = \text{softmax}_q(\text{MLP}(\xi_{q,v} + x_v)) \quad (14)$$

$$\tilde{\kappa}_v = (\sum_q \lambda_{q,v} \xi_{q,v}) + x_v \quad (15)$$

where $\tilde{\kappa}_v$ is the user-level translation matrix in which each entry is a cosine similarity score between two initial user representations. This formulation of $\tilde{\kappa}_v$ allows us to reason about the final prediction by considering user-based subtle clues, e.g., the overlap between users in Fig. 1.

**Channel fusion.** We fuse the channels to better integrate information from the textual and social inputs. To this end, we first refine the node-level attention scores by aggregating the textual and user representations. Specifically, we replace $\gamma_{q,v}$ and $\lambda_{q,v}$ in Eqs. (10) and (14) with $\text{softmax}_q(\text{MLP}(\tilde{z}_{q,v} \odot z_v + \tilde{x}_{q,v} \odot x_v))$. This allows us to combine both textual and social clues when reasoning about one node based on another node. For example, in Fig. 1, we may consider both words related to “hate” and the overlapping user when reasoning about evidence group 1 from the perspective of evidence group 2.

We then fuse the kernel-based textual and user representations to predict the label that node $v$ gives:

$$P(y \mid v, G) = \text{sigmoid}_v(W_5(\kappa_v) + W_6(\tilde{\kappa}_v) + b_5) \quad (16)$$

**Joint Optimization** Let $|\theta|$ be the L2 norm of all model parameters. For each news article $S$, we compute the loss $L_S = -y^* \log(\hat{p}) + (1 - y^*) \log(1 - \hat{p}) + \lambda_{reg} ||\theta||^2$ (20) where $y^*$ is the ground-truth label, $\hat{p} = \mathbb{P}(y \mid G, R)$ is the probability learned based on Eq. (3), and $\lambda_{reg}$ is the regularization coefficient. The parameters are then optimized jointly by minimizing $\sum_{S \in \mathcal{N}} L_S$, where $\mathcal{N}$ consists of all the news articles in the training set.

**Experimental Setup**

**Dataset** To evaluate the performance of FinerFact, we conduct experiments on two benchmark datasets, PolitiFact and GossipCop (Shu et al. 2020), which contain 815 and 7,612 news articles, and the social context information about the news, their labels provided by journalists and domain experts. We follow (Dun et al. 2021) to preprocess the data and conduct experiments. More details about dataset and the pre-processing steps are given in the supplement.

**Baselines** We compare our FinerFact method with eight baselines, which can be divided into two groups:

The first group (G1) is context-based methods, which leverage the textual or visual content of the news for fake news detection. G1 contains four baselines: SVM (Yang et al. 2012), GRU-2 (Ma et al. 2016), RFC (Kwon et al. 2013), and DTC (Castillo, Mendoza, and Poblete 2011).

The second group (G2) consists of knowledge-aware methods that detect fake news by leveraging auxiliary knowledge such as knowledge graphs and social knowledge about the online posts. This group includes four methods: B-TransE (Pan et al. 2018), KCNN (Wang et al. 2018a), GCAN (Lu and Li 2020) and KAN (Dun et al. 2021).

**Evaluation Criteria** Our evaluation criteria include Precision (Pre), Recall (Rec), the F1 score, Accuracy (Acc), and Area Under the ROC curve (AUC). We conduct 5-fold cross validation and the average performance is reported.
Implementations Details. To choose the number of topics, we conducted a grid search within the range $[2, 10]$ and picked the number that results in the smallest perplexity. BERT is fine-tuned during training. A more comprehensive description about the implementation details, experimental setup, and evaluation for topic quality are in the supplement.

Overall Performance

Table 1 compares our method FinerFact with the baselines. As shown in the table, FinerFact consistently outperforms the baseline in both datasets. For example, FinerFact performs better than the most competitive baseline KAN by 6.3%, 5.1% on PolitiFact and 9.7%, 5.5% on GossipCop, respectively, in terms of the F1-scores and accuracy. This demonstrates the effectiveness of our fine-grained reasoning framework, which enables the model to make predictions by identifying and connecting different types of subtle clues. Meanwhile, comparing with GCAN, which models the interactions between the textual content and social information with co-attention, FinerFact increases F1 by 8.3%, 9.8% on the two datasets. This implies that our kernel-attention-based approach can better model the interactions between news articles and evidence. We also observe that methods that incorporate external knowledge (G2) generally perform better than content-based methods (G1). This illustrates the usefulness of external knowledge in fake news detection.

Ablation Study and Sensitivity Analysis

We conduct the ablation study by implementing five variants of our method: 1) \textbf{FF-P} removes the attention prior $\delta(v, l)$ when learning node importance; 2) \textbf{FF-B} eliminates bi-channel reasoning by removing the user-based reasoning channel; 3) \textbf{FF-K} replaces the kernel-based representation learning with a GNN-based aggregation scheme, i.e., replacing $z_{q,v}$ and $x_{q,v}$ with $z_q$ and $x_q$ in Eqs. (9)(15); 4) \textbf{FF-I} excludes node importance learning and assigns an equal weight to every node; 5) \textbf{FF-M} eliminates mutual-reinforcement-based evidence ranking, and selects the evidence for each topic $t$ by random sampling. The ablation study results in Fig. 3 show that removing each component leads to a decrease in model performance, which demonstrates the effectiveness of our major components. A result of FF-T, which replaces our pretrained text encoder with that of KAN, can be found in the supplement.

We then conduct a sensitivity analysis of FinerFact by changing the number of kernels $\tau$. Fig. 4 shows that FinerFact consistently outperforms the state-of-art baseline KAN with varying numbers of kernels, which demonstrates the robustness of our method. In addition, the performance is the best when using around 11 kernels. Using more kernels does not necessarily lead to better performance due to overfitting.

Case Study

In addition to improving accuracy, our method also enables humans to understand most parts in the reasoning workflow. In this case study, we illustrate how FinerFact reasons about the authenticity of a news story, which is about FBI lawyer Lisa Page disclosing that she was instructed to cover-up China’s hacks of the DNC server. FinerFact successfully identifies that the news is fake, with a detailed explanation about the salient evidence, subtle clues, and the prediction scores for each viewpoint.

Identifying salient evidence. As shown in Fig. 5(a), FinerFact identifies meaningful and relevant keywords that belong to diverse topics. For each topic $t$, we can further understand its key evidence by observing its salient posts $P_t$. The best results are highlighted in bold.

Table 1: Performance comparison of FinerFact w.r.t. baselines.

<table>
<thead>
<tr>
<th>Method</th>
<th>PolitiFact</th>
<th>GossipCop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre</td>
<td>Rec</td>
</tr>
<tr>
<td>SVM</td>
<td>0.7460</td>
<td>0.6826</td>
</tr>
<tr>
<td>RFC</td>
<td>0.7470</td>
<td>0.7361</td>
</tr>
<tr>
<td>DTC</td>
<td>0.7476</td>
<td>0.7454</td>
</tr>
<tr>
<td>GRU-2</td>
<td>0.7083</td>
<td>0.7048</td>
</tr>
<tr>
<td>B-TransE</td>
<td>0.7739</td>
<td>0.7658</td>
</tr>
<tr>
<td>KCNN</td>
<td>0.7852</td>
<td>0.7824</td>
</tr>
<tr>
<td>GCAN</td>
<td>0.7945</td>
<td>0.8117</td>
</tr>
<tr>
<td>KAN</td>
<td>0.8687</td>
<td>0.8499</td>
</tr>
<tr>
<td>FinerFact</td>
<td>0.9196</td>
<td>0.9037</td>
</tr>
</tbody>
</table>

Figure 3: Results of the ablation study.

Figure 4: Sensitivity analysis w.r.t. the number of kernels.
Figure 5: Reasoning with FinerFact: (a) the keyword layer of the mutual reinforcement graph $M$, with saliency $R$ indicated by the font size; (b) fine-grained reasoning over the claim-evidence graph $G$. Each color encodes a topic.

and average user saliency $\bar{r}_{ij}$. As shown in Fig. 5(b), users of topic 1 support the news because of their political stance (consider China as a threat) and are generally not credible opinion leaders (small $\bar{r}_{ij}$). In contrast, users of topic 4, who question that the news is fake with more objective reasons, e.g., it is unlikely that an outsider lawyer knows about the server hack (Fig. 5A), receive more attention (larger $\bar{r}_{ij}$).

**Reasoning with subtle clues.** The token-level and user-level attention scores $\alpha_{q,v}^i$ and $\rho_{q,v}^i$ reveal the subtle clues FinerFact detects. For example, in topic 2, the words with the largest $\alpha_{q,1,2}^i$ and $\alpha_{q,4,2}^i$ are “china” and “conspiracy”. These clues are meaningful and interesting: the statement about “china” being unlikely to collaborate with FBI in topic 2 decreases the credibility of the posts in topic 1, and relating “conspiracy” with topic 4 (about “hacking”) enables us to understand that the news may be fake because such a hacking conspiracy is likely to be made up by people who like to talk about it (Fig. 5B). Topics 1 and 4 are also related: user 2, who has the largest $\rho_{4,1,1}^i$, questions users in topic 1 by commenting on them, and points out that China’s problem in terms of intellectual property does not mean that Chinese will hack the server (Fig. 5C).

**Prediction for each viewpoint.** Based on the subtle clues, FinerFact makes prediction for each node. Our method understands that evidence from groups 1 and 3 imply that the news is true (Pr$\left( y = 0 | v, G \right) > 0.5$) and that the evidence from groups 2 and 4 imply that the news is fake (Pr$\left( y = 0 | v, G \right) < 0.5$). It assigns a low probability score that is close to 0.5 to group 1 by propagating the information from groups 2 and 4 to group 1 (large $\gamma_{0,v}$). It also assigns a small node importance Pr$\left( v | G, R \right)$ to group 1. This is reasonable, since group 1 has a low user saliency $\bar{r}_{ij}$, which can be modeled by using the attention prior. While the users in group 3 are considered salient according to the mutual reinforcement graph, we find that they are not talking about whether the article is true, but are instead drifting towards criticizing Hillary. Our model successfully identifies this and assigns a low node importance to topic 3.

**Steering the model.** FinerFact also provides opportunities for users to steer and refine the model. For example, we may integrate FinerFact with the method proposed by Liu et al. (2015) to enable interactive refinement of evidence ranking. Please refer to the supplement for more details.

**Related Works**

Methods for fake news detection can be divided into two main categories: content-based and knowledge-aware.

**Content-based** methods mainly utilize the textual or visual content from the news article and related posts for news verification (Yang et al. 2012; Afroz, Brennan, and Greenstadt 2012; Kwon et al. 2013; Przybylo 2020; Ma et al. 2016; Zellers et al. 2019; Qi et al. 2019; Gupta et al. 2013; Jin et al. 2016b; Kaliyar, Goswami, and Narang 2021). These methods enable the detection of fake news at an early stage (Wei et al. 2021; Pelrine, Danovitch, and Rabbany 2021). However, their performance is limited as they ignore auxiliary knowledge for news verification.

**Knowledge-aware** methods leverage auxiliary knowledge for news verification (Ruchansky, Seo, and Liu 2017; Wang et al. 2018b; Shu et al. 2019; Jin et al. 2016a; Ma, Gao, and Wong 2018; Cho et al. 2014; Wang et al. 2018a).

These methods typically utilize external knowledge about entity relationships (Xu et al. 2022; Dan et al. 2021; Pan et al. 2018; Silva et al. 2021; Hu et al. 2021) or social knowledge about online posts (Lu and Li 2020; Nguyen et al. 2020; Khoo et al. 2020; Bial et al. 2020) for fake news detection. While existing methods have demonstrated the usefulness of heterogeneous social relations and external information (Yuan et al. 2019), they either do not model the interactions between the news content and different types of knowledge data, or model them at a coarse-grained (e.g., sentence or post) level, which limits their performance. We tackle this issue by proposing a prior-aware bi-channel kernel graph network, which enables fine-grained reasoning and improves detection accuracy.

**Conclusion**

We propose FinerFact, a fine-grained reasoning framework for explainable fake news detection. We devise a mutual-reinforcement-based method for efficient evidence ranking and a prior-aware bi-channel kernel graph network for fine-grained reasoning on multiple groups of evidence. Experimental results show the effectiveness of our method.
Rousseau, F.; and Vazirgiannis, M. 2013. Graph-of-word and TW-IDF: new approach to ad hoc IR. In CIKM, 59–68.


Xu, W.; Wu, J.; Liu, Q.; Wu, S.; and Wang, L. 2022. Mining Fine-grained Semantics via Graph Neural Networks for Evidence-based Fake News Detection. In WWW.


Yuan, C.; Ma, Q.; Zhou, W.; Han, J.; and Hu, S. 2019. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In ICDM, 796–805. IEEE.
