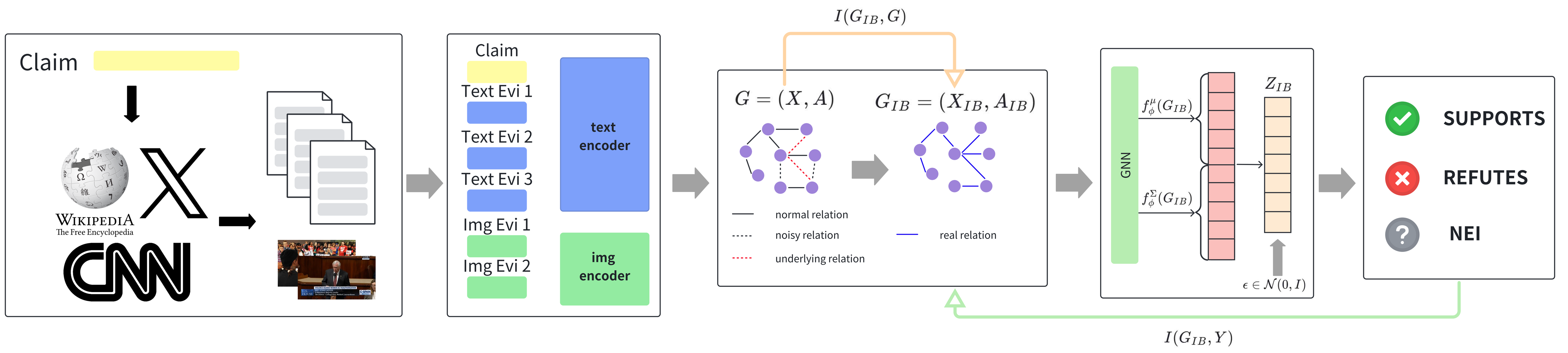
FC-VIB: Fact Checking with Variational Information Bottleneck

Ziheng Luo[1,2], Meng Shen[1,2], and Yong Liao[1,2]

CCCD Key Lab of Ministry of Culture and Tourism, Hefei, Anhui, China[1]

University of Science and Technology, Hefei, Anhui, China[2]

solxags@mail.ustc.edu.cn, asmw707@mail.ustc.edu.cn and yliao@ustc.edu.cn



(1)Evidence Retrieval (2)Initial Node Representations (3)Graph Structure Learning (4)Distribution Learning (5)Claim Verification

Abstract

The proliferation of false information, such as fake news, political deception, and online rumors, has emerged as a critical societal issue, necessitating automated fact-checking methods to mitigate their negative impact. Existing studies primarily follow a three-step pipeline: document retrieval, evidence sentence selection, and claim verification, with significant attention given to retrieval and multi-evidence reasoning. However, current approaches often suffer from noise propagation and oversmoothing issues in graph neural networks, which undermine their ability to capture diverse perspectives from evidence. To address these challenges, we propose a novel fact verification framework, FC-VIB, grounded in the variational information bottleneck (VIB) theory. FC-VIB introduces an adaptive evidence graph structure learning method that controls the flow of information into the reasoning model, balancing prediction and compression to filter task-irrelevant information. The framework includes four stages: (1) encoding claims and evidence using visual and text encoders to initialize node features; (2) learning the evidence graph structure adaptively to construct the IB-Graph (GIB); (3) utilizing a graph neural network to obtain the evidence graph representation distribution; and (4) sampling from the learned distribution via the reparameterization trick for claim verification. The model is trained with supervised classification and KL-divergence loss for the IB objective. Extensive experiments on a public text-based fact verification benchmark and two multimodal benchmarks demonstrate that FC-VIB effectively constructs high-quality evidence graphs, achieving superior performance compared to state-of-the-art baselines.

Method

Following previous research, FC-VIB first constructs an evidence graph using retrieved evidence $D = \{e_t^1, ..., e_t^n, e_i^1, ..., e_i^m\}$ for claim c. Building on the previously introduced graph information bottleneck theory, we employ a generator to produce the optimized $G_{Info-BN}$. We assume that $Z_{Info-BN}$ follows a normal distribution and train the model to learn the mean and variance of this distribution based on $G_{Info-BN}$. Using the reparameterization trick, we sample $Z_{Info-BN}$ from the learned normal distribution. Subsequently, a classifier is utilized to predict the claim label Y.

Initial Node Representations The node representations are initialized by concatenating the features of claim and evidence. Specifically, we leverage pre-trained text encoders (DeBERTa) to extract features from claims and textual evidence, while utilizing pre-trained image encoders (ViT) to capture features from image evidence.

For the claim and evidence, we get the token hidden states and obtain the features from the representation of the first token ("[CLS]"):

$$z_c = TextEncoder(c) \tag{1}$$
$$z_{e_t^j} = TextEncoder(e_t^j), e_t^j \in \{e_t^1, ..., e_t^n\} \tag{2}$$
$$z_{e_i^j} = ImageEncoder(e_i^j), e_i^j \in \{e_i^1, ..., e_i^m\} \tag{3}$$
$$\tag{4}$$

where z_c is the feature of claim. $z_{e_t^j}$ and $z_{e_i^j}$ are the features of text evidence and image evidence respectively.

Following previous works, we concatenate the features of the claim and the evidence as the initial node features.

$$x_j = z_c \oplus z_{e_t^j}, e_t^j \in \{e_t^1, ..., e_t^n\} \tag{5}$$
$$x_{j+n} = z_c \oplus z_{e_i^j}, e_i^j \in \{e_i^1, ..., e_i^m\} \tag{6}$$

where x_j represents the feature of the j-th node in the evidence graph G.

Info-Bottleneck Graph Generator We propose an Info-Bottleneck graph generator to construct the Info-Bottleneck graph $G_{Info-BN}$ for the input graph G. Based on the assumption that structural components may contain nuisance information, the procedure for generating the structure is as follows.

Each potential edge is represented as an independent Bernoulli random variable, with its probability determined by the learned attention weights π:

$$A_{Info-BN} = \bigcup_{u,v \in V} \{a_{u,v} \sim Ber(\pi_{u,v})\} \tag{7}$$

For every node pair, the edge sampling probability π is optimized alongside the graph representation learning process. $\pi_{u,v}$ reflects the task-specific relevance of the edge (u, v), where a lower value of $\pi_{u,v}$ suggests that the edge is more likely to be noisy and should receive a lower weight or potentially be eliminated.

For a pair of nodes (u, v), the edge sampling probability $\pi_{u,v}$ is calculated by:

$$Z(u) = \mathbf{f}(X_{Info-BN}(u)), \tag{8}$$
$$\pi_{u,v} = sigmoid(Z(u)Z(v)^T) \tag{9}$$

where $\mathbf{f}(\cdot)$ denotes a neural network and we use a two-layer MLP in this work. A challenge arises because $A_{Info\text{-}Bottleneck}$ is not differentiable with respect to π due to the discrete nature of the Bernoulli distribution. To address this, we adopt the concrete relaxation of the Bernoulli distribution to enable the optimization of π:

$$Ber(\pi_{u,v}) \approx sigmoid\left(\frac{1}{t}\left(\log\frac{\pi_{u,v}}{1-\pi_{u,v}} + \log\frac{\epsilon}{1-\epsilon}\right)\right) \tag{10}$$

where $\epsilon \sim Uniform(0, 1)$ and $t \in \mathbb{R}^+$ is the temperature for the concrete distribution. With the concrete relaxation applied, the binary variables $a_{u,v}$, originally sampled from a Bernoulli distribution, are reparameterized as a deterministic function of $\pi_{u,v}$ and a noise term ϵ.

After applying the concrete relaxation, the resulting graph becomes a weighted fully connected graph, leading to high computational costs. To mitigate this, we construct a symmetric sparse adjacency matrix by masking out entries with values below a non-negative threshold a_0.

Info-Bottleneck Graph Representation Distribution Learning To compute the compression term $I(Z_{Info\text{-}Bottleneck}; G)$, we assume both the prior $r(Z_{Info\text{-}Bottleneck})$ and the posterior $p(Z_{Info\text{-}Bottleneck}|G)$ follow parametric Gaussian distributions, enabling an analytical calculation of the Kullback-Leibler (KL) divergence:

$$r(Z_{Info-BN}) = \mathcal{N}(\mu_0, \Sigma_0) \tag{11}$$
$$p(Z_{Info-BN}|G) = \mathcal{N}\left(f_\phi^\mu(G_{Info-BN}, f_\phi^\Sigma(G_{Info-BN})\right) \tag{12}$$

where $\mu \in \mathbb{R}^K$ and $\Sigma \in \mathbb{R}^{K\times K}$ is the mean vector and the diagonal co-variance matrix of $Z_{Info-BN}$ encoded by $f_\phi(G_{Info-BN})$.

The dimensionality of $Z_{Info\text{-}Bottleneck}$, denoted as K, defines the size of the information bottleneck. We parameterize $f_\phi(G_{Info\text{-}Bottleneck})$ using a graph neural network (GNN) with parameters ϕ, where the GNN outputs a $2K$-dimensional vector comprising $f_\phi^\mu(G_{Info\text{-}Bottleneck})$ and $f_\phi^\Sigma(G_{Info\text{-}Bottleneck})$, representing the mean and variance, respectively:

$$\forall u \in V, Z_{Info-BN}(u) = \mathbf{GNN}(X, A_{Info-BN}) \tag{13}$$
$$\left(f_\phi^\mu(G_{Info-BN}), f_\phi^\Sigma(G_{Info-BN})\right) = \mathbf{Pooling}(\{Z_{Info-BN}(u), \forall u \in V\}) \tag{14}$$

where the first K-dimension outputs encode μ and the remaining K-dimension outputs encode Σ (we use a softplus transform for $f_\phi^\Sigma(G_{Info-BN})$ to ensure the non-negativity). We treat $r(Z_{Info-BN})$ as a fixed d-dimensional spherical Gaussian $r(Z_{Info-BN}) = \mathcal{N}(Z_{Info-BN}|0, I)$.

Info-Bottleneck Graph Representation Representation Sampler To compute $Z_{Info\text{-}Bottleneck}$, we apply the reparameterization trick, which allows for efficient gradient estimation:

$$Z_{Info-BN} = f_\phi^\mu(G_{Info-BN}) + f_\phi^\Sigma(G_{Info-BN}) \odot \epsilon \tag{15}$$

where $\epsilon \in N(0, I)$ is an independent Gaussian noise and $\odot$ denotes the element-wise product. By applying the reparameterization trick, the randomness is introduced through ϵ, ensuring that it does not interfere with backpropagation. For the first term $I(Z_{Info\text{-}Bottleneck}, Y)$, the distribution $q_\theta(Y|Z_{Info\text{-}Bottleneck})$ represents the label distribution of the learned graph $G_{Info\text{-}Bottleneck}$. We model this distribution using a multi-layer MLP classifier with parameters θ, where $Z_{Info\text{-}Bottleneck}$ serves as the input, and the network outputs the predicted label.

$$Y = \mathbf{MLP}(Z_{Info-BN}) \tag{16}$$

Training Objective. Using gradient descent and backpropagation techniques, we can efficiently compute the upper bounds on the training data samples. The total loss function is given by:

$$\mathcal{L} = \mathcal{L}_{CE}(Z_{Info-BN}, Y) + \beta\mathcal{D}_{KL}(p(Z_{Info-BN}|G)||r(Z_{Info-BN})) \tag{17}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss and $\mathcal{D}_{KL}(\cdot||\cdot)$ is the KL divergence.

Expriment

To validate the effectiveness of our method, we systematically compared it with multiple widely-adopted fake news detection techniques. We conduct experiments on three public benchmark datasets, i.e., FEVER, FACTIFY and MOCHEG. Compared to baseline models, FC-VIB achieves the best performance in most test scenarios. This demonstrates the effectiveness of FC-VIB in graph-based reasoning models. For comprehensive experimental findings, please refer to our paper.

Conclusion

In this work, we propose a fact verification method based on the Variational Information Bottleneck (VIB) theory. Our approach effectively controls the flow of evidence information into the graph reasoning model, striking a balance between prediction and compression. This balance helps limit the retention of task-irrelevant information within the graph reasoning model. The results demonstrate the effectiveness of this framework, as our final pipeline achieves significant improvements. In the future, we aim to further explore evidence interaction mechanisms and enhance the interpretability of fact verification models.