LOREN: Logic-Regularized Reasoning for Interpretable Fact Verification

Jiangjie Chen[1,2*], Qiaoben Bao[1], Changzhi Sun[2], Xinbo Zhang[2], Jiaze Chen[2], Hao Zhou[2], Yanghua Xiao[1,4†], Lei Li[3†‡]

[1]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[2]ByteDance AI Lab [3]University of California, Santa Barbara
[4]Fudan-Aishu Cognitive Intelligence Joint Research Center
{jjchen19, qbbao19, shawyh}@fudan.edu.cn, lilei@cs.ucsb.edu,
{sunchangzhi, zhangxinbo.freya, chenjiaze, zhouhao.nlp}@bytedance.com

Abstract

Given a natural language statement, how to verify its veracity against a large-scale textual knowledge source like Wikipedia? Most existing neural models make predictions without giving clues about which part of a false claim goes wrong. In this paper, we propose LOREN, an approach for interpretable fact verification. We decompose the verification of the whole claim at phrase-level, where the veracity of the phrases serves as explanations and can be aggregated into the final verdict according to logical rules. The key insight of LOREN is to represent claim phrase veracity as three-valued latent variables, which are regularized by aggregation logical rules. The final claim verification is based on all latent variables. Thus, LOREN enjoys the additional benefit of interpretability — it is easy to explain how it reaches certain results with claim phrase veracity. Experiments on a public fact verification benchmark show that LOREN is competitive against previous approaches while enjoying the merit of faithful and accurate interpretability. The resources of LOREN are available at: https://github.com/jiangjiechen/LOREN.

1 Introduction

The rapid growth of mobile platforms has facilitated creating and spreading of information. However, there are many dubious statements appearing on social media platforms. For example, during the 2020 U.S. presidential election, there are many false claims about Donald Trump winning the election, as shown in Figure 1. Verifying these statements is in critical need. How to verify the validity of a textual statement? We attempt to predict whether a statement is *supported*, *refuted* or *unverifiable* with an additional large textual knowledge source such as Wikipedia. Notice that it is computationally expensive to compute the input statement with every sentence in Wikipedia.

This work focuses on interpretable fact verification — it aims to provide decomposable justifications in addition to an overall veracity prediction. We are motivated by a simple intuition: the veracity of a claim depends on the truthfulness of its composing phrases, e.g., subject, verb, object phrases. A false claim can be attributed to one or more unsupported phrases, which we refer to as the *culprit*. The claim is valid if all phrases are supported by certain evidence sentences in Wikipedia. For example, one culprit in Figure 1 would be the phrase "won". Therefore, faithful predictions of phrase veracity would explain why a verification model draws such a verdict. In addition, through phrasal veracity prediction, identifying the culprit also alleviates the burden of correcting an untrustworthy claim, as we can easily alter "won" to "lost" to make it right.

Most current studies focus on designing specialized neural network architectures, with the hope of exploiting the semantics from sentences (Nie, Chen, and Bansal 2019; Zhou et al. 2019; Liu et al. 2020b; Zhong et al. 2020; Si et al. 2021; Jiang, Pradeep, and Lin 2021). However, these methods are limited in interpretability, as they usually only give an overall verdict. This puts forth trust issues for humans, as a deci-

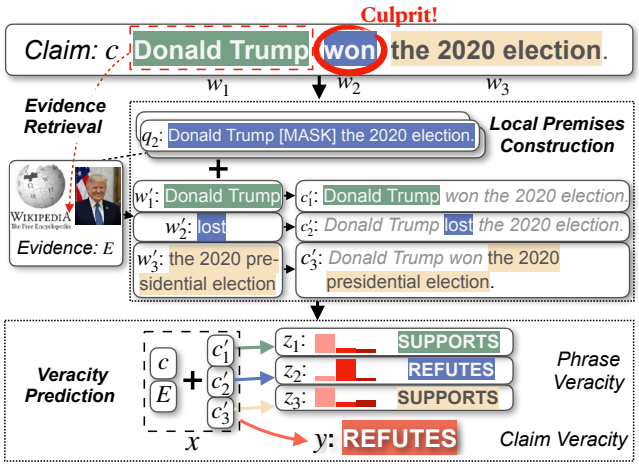


Figure 1: An example of how our proposed fact verification framework LOREN works. Texts highlighted denote the data flow for three phrases extracted from the claim. LOREN not only makes the final verification but also finds the culprit phrase (w_2) that causes the claim's falsity.

*Work is done during internship at ByteDance AI Lab.
†Corresponding authors.
‡Work is done while at ByteDance AI Lab.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

sion is usually made in a black-box fashion. Recent studies on explainable fact verification (Stammbach and Ash 2020; Samarinas, Hsu, and Lee 2021; Wu et al. 2021) mostly focus on giving intuitive display of the key basis for the model results, instead of building an interpretable models that output the reasons for obtaining the results while giving the results.

It is challenging to learn interpretable models that not only predict but also explain its rationale, since there is a lack of truthfulness labels on the phrase level. There are public datasets with veracity label for the overall claim, but it is unknown which part of the claim makes it untrustworthy. Manually annotating such fine-grained data is unrealistic and requires tremendous human labor. How to supervise a model to reach meaningful phrasal veracity predictions? Our insight comes from the intuition that simple *symbolic logical rules* can be utilized to create weak supervisions for intermediate phrasal predictions. Empirically, all phrases should be supported if a claim is true, and a claim is refuted if there exists at least one false phrase. If the outcome of a claim is *unverifiable*, then there must be no *refuted* phrase and at least one phrase that should be verified as *unverifiable*. With the logical rules in mind, we only have to identify the patterns these suspicious phrases give during training.

For this purpose, we propose LOREN, a LOgic-REgularized Neural latent model, to predict the veracity of a claim, as well as to give explanation. The overall idea of LOREN is to decompose the verification into a series of hypothesis verification at the phrase level, which are constrained by the introduced logical aggregation rules. Each rule concerns the compositional logic that describes how phrasal veracity predictions are logically aggregated into claim veracity. Together, the veracity prediction of every claim phrase serves as the atomic propositions of the compositional logic. Thus, a key perspective of LOREN is to represent these phrase veracity as latent variables regularized by the softened aggregation logic for meaningful predictions for claim phrases. To solve this latent model, LOREN adopts a modern approach using amortized variational inference (variational auto-encoding) (Kingma and Welling 2014). Furthermore, LOREN constructs the teacher model by aggregating logic over all latent variables, distilling logical knowledge to the student (claim verification) model. To arrive at these propositions, as another key perspective, we convert the problem of finding relevant phrases in evidence into a machine reading comprehension (MRC) task, where we generate probing questions for evidence to answer. To summarize, the contributions of this work include:

- We propose an interpretable method LOREN to predict the veracity of both a claim sentence and its phrases.
- We present a technique to weakly supervise phrasal veracity learning with a MRC module and latent variable modeling regularized by logical rules.
- We experiment LOREN on FEVER (Thorne et al. 2018), a large fact verification benchmark. Besides competitive verification results, LOREN also provides *faithful* (over 96% agreement) and *accurate* phrasal veracity predictions as explanations.

2 Related Work

There are several related problems about verifying the truthfulness of one or multiple sentences, including natural language inference (NLI) (Kang et al. 2018), claim verification (Thorne et al. 2018), misinformation detection (Zellers et al. 2019), etc. In this paper, we study the claim verification task (Thorne et al. 2018), which focuses on verifying claims against trustworthy knowledge sources. The majority of existing studies adopt a two-step pipeline to verify a textual claim, i.e., evidence retrieval and claim verification. Current verification systems can be categorized by the granularity of the interaction between claim and evidence, including those of sentence-level (Nie, Chen, and Bansal 2019; Zhou et al. 2019), semantic role-level (Zhong et al. 2020) and word-level (Liu et al. 2020b). They learn the representations of claim and evidence sentences of different granularity based on neural networks and gives a final verdict in an end-to-end fashion. In contrast, we conduct *phrase-level* verification and take a further step forward to more interpretable reasoning and verification.

There are some recent studies on interpretable fact verification, such as using GPT-3 (Brown et al. 2020) to summarize evidence and generate explanations (Stammbach and Ash 2020), pointing out salient pieces in evidence with attention weights (Samarinas, Hsu, and Lee 2021), and picking relevant sentences in retrieved evidence (Wu et al. 2021). Instead, we take a different route towards interpretable fact verification by producing where and how a claim is falsified. The final verdict is drawn based on explanations, making a step forward to being right for the right reasons.

Previous efforts towards unifying symbolic logic and neural networks include those of Sourek et al. (2015); Manhaeve et al. (2018); Lamb et al. (2020). A class of integrated symbolic logic and neural network methods is based on the variational EM framework (Qu and Tang 2019; Zhou et al. 2020). Another standard method is to soften logic with neural network components (Hu et al. 2016; Li et al. 2019; Wang and Pan 2020), which can be trained in an end-to-end manner. Our method draws inspiration from both lines of work. We represent the intermediary veracity predictions as latent variables in latent space, which are regularized with softened logic.

3 Proposed Approach

In this section, we present the proposed method LOREN for verifying a textual claim against a trustworthy knowledge source (e.g., Wikipedia), which consists of two sub-tasks: *1)* evidence retrieval and *2)* fact verification. In this paper, we primarily focus on fact verification and assume evidence text (e.g., several related sentences) is retrieved by a separate method. A possible verification result can be *supported* (`SUP`), *refuted* (`REF`) or *not-enough-information* (`NEI`).

Different from most previous methods that give an overall prediction, our goal is to predict the final *claim veracity* and faithful *phrase veracity* as explanations. First, we define the task of *claim verification* and *phrase verification*.

Claim Verification Given a claim sentence c and retrieved evidence text E, our goal is to model the probability distri-

bution $p(\boldsymbol{y}|c, E)$, where $\boldsymbol{y} \in \{\texttt{SUP}, \texttt{REF}, \texttt{NEI}\}$ is a three-valued variable indicating the veracity of the claim given evidence. In this paper, ***bold*** letters indicate variables.

Phrase Verification We decompose the verification of a claim at phrase-level, and predict the veracity $\boldsymbol{z}_i$ of a claim phrase $w_i \in \mathcal{W}_c$ by $p(\boldsymbol{z}_i|c, w_i, E)$, where $\boldsymbol{z}_i \in \{\texttt{SUP}, \texttt{REF}, \texttt{NEI}\}$. We extract the claim phrases $\mathcal{W}_c$ with a set of heuristic rules using a series of off-the-shelf tools provided by AllenNLP (Gardner et al. 2018). Claim phrases include named entities (NEs), verbs, adjective and noun phrases (APs and NPs).

Specifically, we leverage a part-of-speech (POS) tagger for identifying verbs and a constituency parser to identify noun phrases (NPs). For the fine-grained extraction of NPs, we further decompose them into more fine-grained ones using POS tagger and named entity recognizer (NER). We use several simple *heuristic* rules, including: 1) we parse all leaf NPs, and keep all verbs with a POS tagger; 2) we break down the NPs with an NER and isolate the adjectives from NPs for finer-grained phrases. For example, we have "*Donald Trump*" (NE), "*won*" (verb) and "*the 2020 election*" (NP) as claim phrases in Figure 1.

3.1 Logical Constraints

After introducing the phrase verification, we observe that some natural, logical consistencies between phrase verification and claim verification should be satisfied. Specifically, a claim is found *1)* `REF` if at least one claim phrase is refuted by evidence; *2)* `SUP` if all claim phrases are supported; *3)* `NEI` if neither of the above, that is, there is no contradictory but at least one phrase gets unknown outcome. Notice that the checking rule for the `REF` judgment has priority over `NEI`, because it is also possible for a phrase to be `NEI` in a *refuted* claim, but not vice versa. Formally, we give the following definition of the aggregation logic.

Definition 1 *Given a statement c, a set of claim phrases $\mathcal{W}_c$, and a set of evidence E, with $\top(c)$, $\bot(c)$ and $\ominus(c)$ denoted as true, false and unknown respectively. $V(c, \mathcal{W}_c, E)$ is defined as the value of c taking one of the three, i.e. $\{\top, \bot, \ominus\}$ w.r.t. $\mathcal{W}_c$ given evidence E, which corresponds to the predicted label $y \in \{\texttt{SUP}, \texttt{REF}, \texttt{NEI}\}$. Then we have:*

$$
\begin{aligned}
V(c, \mathcal{W}_c, E) &\models \top, \text{ iff } \forall w \in \mathcal{W}_c, V(c, w, E) \models \top \\
V(c, \mathcal{W}_c, E) &\models \bot, \text{ iff } \exists w \in \mathcal{W}_c, V(c, w, E) \models \bot \\
V(c, \mathcal{W}_c, E) &\models \ominus, \text{ iff } \neg(V(c, \mathcal{W}_c, E) \models \top) \wedge \\
&\qquad \forall w \in \mathcal{W}_c, V(c, w, E) \models \{\top, \ominus\}
\end{aligned}
$$

where $V(c, w, E)$ is defined as the value of c w.r.t. a single claim phrase w and the given evidence E.

With the logic in mind, we then introduce how LOREN learns to predict the veracity of both a claim and its phrases without direct supervision for the latter.

3.2 Overview of LOREN

The basic idea of LOREN is to decompose the verification of a claim at phrase-level, and treats the veracity of each phrase $w_i \in \mathcal{W}_c$ as a three-valued latent variable $\boldsymbol{z}_i$. We define $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{|\mathcal{W}_c|})$. The veracity of a claim $\boldsymbol{y}$ depends on the latent variables $\boldsymbol{z}$. Inspired by Hu et al. (2016), to impose the logical constraints mentioned above, we propose a distillation method that transfers the logical knowledge into the latent model. Next, we will detail the latent model and the logical knowledge distillation.

Latent Model We formulate the fact verification task in a probabilistic way. Given an input $x = (c, E)$ consisting of textual claim c and retrieved evidence text E, we define target distribution $p_\theta(\boldsymbol{y}|x)$ as below:

$$
p_\theta(\boldsymbol{y}|x) = \sum_{\boldsymbol{z}} p_\theta(\boldsymbol{y}|\boldsymbol{z}, x) p(\boldsymbol{z}|x) \tag{1}
$$

where $p(\boldsymbol{z}|x)$ is the prior distribution over latent variable $\boldsymbol{z}$ conditioned on the input x, and p_θ gives the probability of $\boldsymbol{y}$ conditioned on x and latent $\boldsymbol{z}$. Note that we assume that $\boldsymbol{z}_i$ is independent of each other, namely, $p(\boldsymbol{z}|x) = \prod_i p(\boldsymbol{z}_i|x, w_i)$. Given the gold label $y^\star$, the objective function is to minimize the negative likelihood as follow:

$$
\mathcal{L}(\theta) = -\log p_\theta(y^\star|x). \tag{2}
$$

Theoretically, we can adopt the EM algorithm for optimization. However, in our setting, it is difficult to compute the exact posterior $p_\theta(\boldsymbol{z}|\boldsymbol{y}, x)$ due to the large space of $\boldsymbol{z}$. With recent advances in the variational inference (Kingma and Welling 2014), we could amortize the variational posterior distribution with neural networks. It results in the well-known variational bound (negative Evidence Lower BOund, ELBO) to be minimized:

$$
\overbrace{-\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{y}, x)}[\log p_\theta(y^*|\boldsymbol{z}, x))] + D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}|\boldsymbol{y}, x) \parallel p(\boldsymbol{z}|x))}^{\text{negative ELBO}: \mathcal{L}_{\text{var}}(\theta, \phi)} \tag{3}
$$

where $q_\phi(\cdot)$ is the variational posterior distribution conditioned on $\boldsymbol{y}, x$, and D_{KL} is Kullback–Leibler divergence. In experiments, we use an off-the-shelf and pre-trained NLI model as prior distribution $p(\boldsymbol{z}|x)$, whose parameters are fixed.[1] The NLI model yields the distribution of `contradicted`, `neutral` and `entailment`, which we take correspond to `REF`, `NEI` and `SUP` to some extent.

Logical Knowledge Distillation To integrate the information encoded in the logical rules into latent variables, we propose a distillation method, which consists of a *teacher model* and a *student model*. The student model is the $p_\theta(\boldsymbol{y}|\boldsymbol{z}, x)$ we intend to optimize. The teacher model is constructed by projecting variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{y}, x)$ into a subspace, denoted as $q_\phi^{\mathrm{T}}(\boldsymbol{y}_z|\boldsymbol{y}, x)$. The subspace is constrained by the logical rules, since $\boldsymbol{y}_z$ is the logical aggregation of $\boldsymbol{z}$. Thus, simulating the outputs of q_ϕ^{T} serves to transfer logical knowledge into p_θ. Formally, the distillation loss is formulated as:

$$
\mathcal{L}_{\text{logic}}(\theta, \phi) = D_{\mathrm{KL}}\left(p_\theta(\boldsymbol{y}|\boldsymbol{z}, x) \parallel q_\phi^{\mathrm{T}}(\boldsymbol{y}_z|\boldsymbol{y}, x)\right). \tag{4}
$$

Overall, the final loss function is defined as the weighted sum of two objectives:

$$
\mathcal{L}_{\text{final}}(\theta, \phi) = (1 - \lambda)\mathcal{L}_{\text{var}}(\theta, \phi) + \lambda \mathcal{L}_{\text{logic}}(\theta, \phi) \tag{5}
$$

[1] We use a DeBERTa (He et al. 2021) fine-tuned on MNLI dataset (Bowman et al. 2015) as the NLI model.

where λ is a hyper-parameter calibrating the relative importance of the two objectives.

3.3 Teacher Model Construction

ELBO cannot guarantee latent variables to be the veracity of corresponding claim phrases without any direct intermediate supervisions. As a key perspective, they are aggregated following previously described logical rules, making them weak supervisions for phrase veracity.

To this end, we relax the logic with soft logic (Li et al. 2019) by product t-norms for differentiability in training and regularization of latent variables. According to §3.1, given probability of the claim phrase veracity $\boldsymbol{z}$, we logically aggregate them into $\boldsymbol{y}_z$ as follows (for simplicity, we drop the input x):

$$\begin{aligned} q_\phi^{\mathrm{T}}(\boldsymbol{y}_z = \texttt{SUP}) &= \prod_{i=1}^{|\boldsymbol{z}|} q_\phi(\boldsymbol{z}_i = \texttt{SUP}) \\ q_\phi^{\mathrm{T}}(\boldsymbol{y}_z = \texttt{REF}) &= 1 - \prod_{i=1}^{|\boldsymbol{z}|} (1 - q_\phi(\boldsymbol{z}_i = \texttt{REF})) \\ q_\phi^{\mathrm{T}}(\boldsymbol{y}_z = \texttt{NEI}) &= 1 - q_\phi^{\mathrm{T}}(\boldsymbol{y}_z = \texttt{SUP}) - q_\phi^{\mathrm{T}}(\boldsymbol{y}_z = \texttt{REF}) \end{aligned} \tag{6}$$

where $\sum_{\boldsymbol{y}_z} q_\phi^{\mathrm{T}}(\boldsymbol{y}_z) = 1$ and $\sum_{\boldsymbol{z}_i} q_\phi(\boldsymbol{z}_i) = 1$.

The prediction behavior of q_ϕ^{T} reveals the information of the rule-regularized subspace, indicating the uncertain and probabilistic nature of the prediction (Chen et al. 2020). By minimizing the distillation loss $\mathcal{L}_{\text{logic}}$ in Eq. 4, the phrasal veracity predictions are regularized by the aggregation logic even if we do not have specific supervisions for claim phrases.

3.4 Building Local Premises

Before parameterizing $p_\theta(\cdot)$ and $q_\phi(\cdot)$ in the latent model, we find the information required for verifying each claim phrase from evidence in an MRC style. We collect them into a set of *local premises* corresponding to each claim phrase, which is important for LOREN's interpretability w.r.t. phrasal veracity. One of the key perspective is to convert the finding of such information into a generative machine reading comprehension (MRC) task, which requires a question generation and answering pipeline.

Probing Question Generation Before MRC, we first build probing questions $\mathcal{Q}$ for every claim phrase respectively. Each question consists of two sub-questions: one *cloze* questions (Devlin et al. 2019) (e.g., "*`[MASK]` won the 2020 election.*") and *interrogative* questions (Wang et al. 2020) (e.g., "*Who won the 2020 election?*"). Both types of questions are complementary to each other. The cloze questions lose the semantic information during the removal of masked phrases (e.g., "*he was born in `[MASK]`*", where *`[MASK]`* can either be a place or a year.). And the generated interrogative ones suffer from the incapability of a text generator. In experiments, we use an off-the-shelf question generation model based on T5$_{base}$ (Raffel et al. 2020) to generate interrogative questions.

Local Premise Construction For every claim phrase $w_i \in \mathcal{W}_c$, we first generate probing question $q_i \in \mathcal{Q}$ with off-the-shelf question generators. The MRC model takes as an input $\mathcal{Q}$ and E and answers $\mathcal{W}_E$. Then, we replace $w_i \in \mathcal{W}_c$ with answers $w_i' \in \mathcal{W}_E$, yielding replaced claims c_i' such as "*Donald Trump lost the 2020 election*", where w_i' ="*lost*" and w_i ="*won*". Such replaced claims are denoted as local premises $\{c_i'\}_{i=1}^{|\mathcal{W}_c|}$ to reason about the veracity of every claim phrase.

Self-supervised Training of MRC The MRC model is fine-tuned in a self-supervised way to adapt to this task at hand. The MRC model takes as input a probing question and evidence sentences and outputs answer(s) for the question. During training, claim phrases $\mathcal{W}_c$ in a claim are used as ground truth answers, which is self-supervised. Note that we build the MRC dataset using *only* `SUP` samples, as the information in `REF` or `NEI` samples is indistinguishably untrustworthy and thus unable to be answered correctly. During inference, the MRC model produces an answer $w_i' \in \mathcal{W}_E$ for a claim phrase $w_i \in \mathcal{W}_c$, which is used to replace w_i for constructing a local premise.

A phrase in the claim may differ in surface form from the answers in the evidence, which is thus *not* suitable for an *extractive* MRC system. Therefore, we adopt a *generative* MRC model under the sequence-to-sequence (Seq2Seq) paradigm (Khashabi et al. 2020).

3.5 Veracity Prediction

Given pre-computed local premises, we then use neural networks to parameterize $p_\theta(\boldsymbol{y}|\boldsymbol{z}, x)$ and the variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{y}, x)$ for veracity prediction. They are optimized by the variational EM algorithm and decoded iteratively.

Given c, E and local premises $\mathcal{P}$ for claim phrases respectively, we calculate the contextualized representations with pre-trained language models (PLMs). We concatenate claim and each of the local premises with $\{x_{\text{local}}^{(i)} = (c, c_i')\}$ and encode them into hidden representations $\{\boldsymbol{h}_{\text{local}}^{(i)} \in \mathbb{R}^d\}$. Similarly, we encode the claim and concatenated evidence sentences as $x_{\text{global}} = (c, E)$ into the global vector $\boldsymbol{h}_{\text{global}} \in \mathbb{R}^d$, followed by a self-selecting module (Liu et al. 2020a) to find the important parts of a vector.

Not all phrases are the culprit phrase, so we design a *culprit attention* based on a heuristic observation that: a valid local premise should be semantically close to the evidence sentences. Thus, we design the similarity between $\boldsymbol{h}_{\text{local}}^{(i)}$ and $\boldsymbol{h}_{\text{global}}$ to determine the importance of the i-th claim phrase. We calculate the context vector $\boldsymbol{h}_{\text{local}}$ as follows:

$$\boldsymbol{h}_{\text{local}} = \tanh(\sum_{i=1}^{|\mathcal{W}_c|} \alpha_i \boldsymbol{h}_{\text{local}}^{(i)}); \alpha_i = \sigma(\mathbf{W}_\alpha[\boldsymbol{h}_{\text{global}}; \boldsymbol{h}_{\text{local}}^{(i)}]) \tag{7}$$

where $\mathbf{W}_\alpha \in \mathbb{R}^{1\times 2*d}$ is the parameter and σ is the softmax function.

After calculating these representations, we design $p_\theta(\cdot)$ and $q_\phi(\cdot)$ both to be two-layer MLPs, where the last layer is shared as label embeddings:

	Training	Development	Test
SUP	80,035	6,666	6,666
REF	29,775	6,666	6,666
NEI	35,659	6,666	6,666

Table 1: Statistics of FEVER 1.0 dataset.

- $q_\phi(\boldsymbol{z}_i|\boldsymbol{y}, x)$ takes as input the concatenation of the label embeddings of $\boldsymbol{y}$ (ground truth $y^\star$ in training), $\boldsymbol{h}^{(i)}_{\text{local}}$ and $\boldsymbol{h}_{\text{global}}$, and outputs the probability of $\boldsymbol{z}_i$. Note that $q_\phi(\boldsymbol{z}|\boldsymbol{y}, x) = \prod_i q_\phi(\boldsymbol{z}_i|\boldsymbol{y}, x)$.
- $p_\theta(\boldsymbol{y}|\boldsymbol{z}, x)$ takes as input the concatenation of $(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{\max})$ (max length by padding), $\boldsymbol{h}_{\text{global}}$ and $\boldsymbol{h}_{\text{local}}$, and outputs the distribution of $\boldsymbol{y}$.

During training, $q_\phi(\cdot)$ and $p_\theta(\cdot)$ are jointly optimized with Eq. 5. We use the Gumbel reparameterization (Jang, Gu, and Poole 2017) for discrete argmax operation from $\boldsymbol{z}$. Specifically, we keep the argmax node and perform the usual forward computation (Gumbel Max), but backpropagate a surrogate gradient (gradient of Gumbel Softmax).

Decoding During inference, we *randomly* initialize $\boldsymbol{z}$, and then iteratively decode $\boldsymbol{y}$ and $\boldsymbol{z}$ with $p_\theta(\boldsymbol{y}|\boldsymbol{z}, x)$ and $q_\phi(\boldsymbol{z}|\boldsymbol{y}, x)$ until convergence. In the end, we have both the final prediction y and the latent variables $\boldsymbol{z}$ serving as the phrasal veracity predictions for all claim phrases.

4 Experiments

4.1 Dataset and Evaluation Metrics

Dataset We evaluate our verification method on a large-scale fact verification benchmark, i.e., FEVER 1.0 shared task (Thorne et al. 2018), which is split into *training*, *development* and *blind test* set. FEVER utilizes Wikipedia (dated June 2017) as the trustworthy knowledge source from which the evidence sentences are extracted. The statistical report of FEVER dataset is presented in Table 1, with the split sizes of SUPPORTED (SUP), REFUTED (REF) and NOT ENOUGH INFO (NEI) classes. In this dataset, there are 3.3 phrases per claim/question on average.

Evaluation Metrics Following previous studies, we evaluate the systems using:

- ***L**abel **A**ccuracy* (LA): The accuracy of predicted label for claim regardless of retrieved evidence;
- ***FEVER** score* (FEV): The accuracy of both predicted label and retrieved evidence, which encourages the correct prediction based on correct retrieved evidence.

In other words, FEVER score rewards a system that makes predictions based on correct evidence. Note that no evidence is needed if a claim is labeled NEI.

In addition, we propose several metrics to evaluate the quality of explanations, i.e., phrasal veracity predictions z:

- *Logically aggregated **L**abel **A**ccuracy* (LA_z): We calculate the accuracy of logically aggregated y_z by Eq. 6, which evaluates the *overall* quality of explanations z;
- ***Culp**rit finding **A**bility* (CULPA): LA_z cannot evaluate *individual* phrase veracity z_i or decide whether a model finds the correct culprit phrase. Thus, we randomly select 100 refuted claims from development set, and manually label the culprit phrases (allowing multiple culprits).[2] CULPA calculates the **P**recision, **R**ecall and **F1** of the culprit finding based on *discrete* veracity from z.
- ***Agree**ment* (AGREE): The agreement between predictions of aggregated veracity y_z and the final veracity y, which evaluates the *faithfulness* of explanations;

We use two ways of aggregation logic for calculating LA_z and AGREE, i.e., discrete *hard* logic (as in §3.1) and probabilistic *soft* logic (as in §3.3).

4.2 Baseline Methods

We evaluate LOREN against several public state-of-the-art baselines:

- **UNC NLP** (Nie, Chen, and Bansal 2019) is the champion system in the FEVER competition, which uses ESIM (Chen et al. 2017) to encode pairs of claim and evidence sentence, enhanced with internal semantic relatedness scores and WordNet features.
- **GEAR** (Zhou et al. 2019), which is a pioneer model to utilize BERT (Devlin et al. 2019) to model the interaction between claim and evidence sentence pairs, followed by a graph network for the final prediction.
- **DREAM** (Zhong et al. 2020), which is built on top of an XLNet (Yang et al. 2019) and breaks the sentences into semantic graphs using semantic role labeler, followed by a graph convolutional network (Velickovic et al. 2018) and graph attention for propagation and aggregation.
- **KGAT** (Liu et al. 2020b), which collapses sentences into nodes, encodes them with RoBERTa (Liu et al. 2019), and adopts a Kernel Graph Attention Network for aggregation. Further research equips KGAT with CorefRoBERTa (Ye et al. 2020), a PLM designed to capture the relations between co-referring noun phrases.
- **LisT5** (Jiang, Pradeep, and Lin 2021) is currently the champion in FEVER 1.0 shared tasks. LisT5 employs a list-wise approach with data augmentation on top of a T5-3B (Raffel et al. 2020) with 3 billion parameters, which is almost 10 times larger than the large versions of BERT, RoBERTa and XLNet.

We note that the comparison between baselines is not always fair due to too many different settings such as evidence retrieval and backbone pre-trained language models.

4.3 Implementation Details

We describe the implementation details in the experiments for the following. LOREN consists of a pipeline of modules, among which the MRC model and the verification model are

[2]Note that the set of annotated culprits is a subset of the extracted claim phrases for the convenience of calculation. We find that there are on average 1.26 culprit phrases per claim for the sampled ones, indicating that the refuted claims in the FEVER dataset generally have a single culprit.

trained by exploiting the FEVER dataset. All of the backbone PLMs inherit HuggingFace's implementation (Wolf et al. 2020) as well as most of the parameters.

Training Details of MRC We train the model in a self-supervised way, i.e., using the SUP samples in training and development set. The constructed dataset consists of 80,035 training samples and 6,666 development samples, corresponding to the statistics of SUP samples in Table 1.

We fine-tune a BART_{base} model (Lewis et al. 2020) for the MRC model. Following the standard Seq2Seq training setup, we optimize the model with cross entropy loss. We apply AdamW as the optimizer during training. We train the model for 4 epochs with initial learning rate of 5e-5, and use the checkpoint with the best ROUGE-2 score on the development set.

Training Details of Veracity Prediction During data preprocessing, we set the maximum lengths of x_{global} and $x^{(i)}_{\text{local}}$ as 256 and 128 tokens respectively, and set the maximum number of phrases per claim as 8. For each claim phrase w_i, we keep the top 3 answers in the beam search as candidates from the MRC model, replace w_i with them respectively, and concatenate the sentences as the local premise for the claim phrase w_i. During training, we set the initial learning rate of LOREN with BERT and RoBERTa as 2e-5 and 1e-5, and batch size as 16 and 8 respectively. The models are trained on 4 NVIDIA Tesla V100 GPUs for ∼5 hours for best performance on development set. We keep checkpoints with the highest label accuracy on the development set for testing. During inference, decoding quickly converges after 2 or 3 iterations.

Evidence Retrieval Since the primary focus of this work is fact verification, we directly adopt the evidence retrieval methods from KGAT (Liu et al. 2020b) for comparison in the verification sub-task. We leave the reported performance of several evidence retrieval techniques and the results of LOREN with oracle evidence retrieval in Appendix.

5 Results and Discussion

In this section, we evaluate the performance of LOREN compared with baselines and analyze the interpretability of LOREN w.r.t. phrase veracity and local premise quality.[3]

5.1 Overall Performance

Table 2 reports the overall performance of LOREN compared with baselines on the development and test set of FEVER. In general, LOREN outperforms or is comparable to published baseline methods of similar sizes. LisT5 shows its superiority over other methods, which may be mainly attributed to its much larger and more powerful PLM (T5-3B). Still, LOREN outperforms LisT5 in FEV score in the development set. For DREAM, we notice it achieves better score in LA score in the test set than LOREN. Due to the difference in evidence retrieval strategies and backbone PLMs, LOREN is not fully comparable with DREAM. However, a higher FEV score of LOREN (for both BERT and RoBERTa) indicates it makes decisions more faithful to evidence than DREAM. In contrast, we make fairer comparisons with KGAT (same PLMs and evidence retrieval techniques), and find that LOREN with $\text{BERT}_{\text{large}}$ and $\text{RoBERTa}_{\text{large}}$ beats KGAT with $\text{RoBERTa}_{\text{large}}$ and $\text{CorefRoBERTa}_{\text{large}}$, respectively.

We then perform a detailed analysis of the proposed components in LOREN ($\text{RoBERTa}_{\text{large}}$) on the development set to assess their influences on the performance and explanation quality.

Model	Dev		Test	
	LA	**FEV**	**LA**	**FEV**
UNC NLP	69.72	66.49	68.21	64.21
GEAR ($\text{BERT}_{\text{base}}$)	74.84	70.69	71.60	67.10
DREAM ($\text{XLNet}_{\text{large}}$)	79.16	-	<u>76.85</u>	70.60
KGAT ($\text{BERT}_{\text{large}}$)	77.91	75.86	73.61	70.24
∟ ($\text{RoBERTa}_{\text{large}}$)	78.29	76.11	74.07	70.38
∟ ($\text{CorefRoBERTa}_{\text{l}}$)	-	-	75.96	72.30
LOREN ($\text{BERT}_{\text{large}}$)	78.44	76.21	74.43	70.71
∟ ($\text{RoBERTa}_{\text{large}}$)	<u>81.14</u>	**78.83**	76.42	<u>72.93</u>
LisT5 (T5_{3B})	**81.26**	<u>77.75</u>	**79.35**	**75.87**

Table 2: Overall performance of verification results on the dev and blind test set of FEVER task, where FEV (FEVER score) is the main evaluation metric. The best is bolded, and the second best is underlined.

λ in $\mathcal{L}_{\text{final}}$	**LA**	$\textbf{LA}_z$		**AGREE**	
		Hard	Soft	Hard	Soft
$\lambda = 0.0$	81.10	51.99	51.92	54.02	53.90
$\lambda = 0.3$	80.98	75.24	78.75	90.06	93.14
$\lambda = 0.5$	**81.14**	76.54	79.66	92.94	96.11
$\lambda = 0.7$	80.92	**77.77**	**80.28**	**93.81**	96.79
$\lambda = 0.9$	80.28	75.55	80.02	91.56	**98.43**

Table 3: Evaluation of phrase veracity quality with the adjustment of the balancing weight λ in the loss function (Eq. 5) in LOREN. The accuracy and faithfulness of phrase veracity boost as λ increases towards $\mathcal{L}_{\text{logic}}$.

5.2 Evaluation of Phrase Veracity

As one of the most important features in LOREN, phrasal veracity predictions explain the verification results. Therefore, such explanations must be accurate as well as faithful to the final prediction. Since the hyper-parameter λ controls the influence of logical constraints, we perform an ablation study of λ, where $\lambda = 0$ indicates no logical constraints on the latent variables.

As seen in Table 3, we report the results of LA, LA_z and AGREE which comprehensively evaluate the general quality of phrasal veracity predictions. We have three major observations from the table: *1)* Aggregation with soft logic is better than hard logic in terms of accuracy and faithful-

[3]We set $\lambda = 0.5$ by default in our experiments.

Choice of $p(z)$	LA	LA_z	AGREE	CULPA (P/R/F1)
NLI prior	**81.14**	79.66	96.11	**75.8**/75.9/**74.3**
Pseudo prior	80.93	80.44	**97.25**	70.5/77.1/71.4
Uniform prior	80.85	**80.74**	97.08	34.1/**78.8**/46.1

Table 4: Results of different choices of prior distribution $p(z)$ during training, where y_z in LA_z is calculated using *soft* logic.

ness. This indicates that predicted probability distributions of phrase veracity are important and gives more information than discrete labels. *2)* In general, the explanations are faithful, with over 96% of aggregated phrase veracity consistent with the claim veracity. The explanations are also accurate according to LA_z and LA scores. *3)* With the increase of λ and stronger regularization of $\mathcal{L}_{\text{logic}}$, the general accuracy and faithfulness of phrase veracity increase. Without $\mathcal{L}_{\text{logic}}$, LOREN cannot give any meaningful explanations.

In summary, the results demonstrate the effectiveness of phrase veracity and the importance of the aggregation logic.

Ablation on Prior Distribution As presented in §3.3, we use the results of a fixed, off-the-shelf NLI model (He et al. 2021) as the prior distribution $p(\boldsymbol{z})$. We first evaluate the quality of NLI predictions in this task by directly making them as phrasal veracity predictions. We make local premises as *premise* and the claim as *hypothesis*. The predictions are aggregated into y_z using the same soft logic, and we get the LA_z score at only 53.41%. However, with LOREN training, LA_z can reach the score at 79.66% or more.

We further perform an ablation study to investigate the influence of the choice of prior distribution. We propose two alternatives:

1. *logical pseudo distribution*. We create pseudo $p(\boldsymbol{z})$ and sample 1 or 2 phrases as the culprit phrase(s) based on culprit attention weight α in Eq. 7, and label them as `REF` and the rest as non-`REF`. Such $p(\boldsymbol{z})$ is in accordance with the logic but distinguishable of the culprit phrase(s);
2. *uniform distribution*, which is commonly used as $p(\boldsymbol{z})$.

z is *randomly* initialized during decoding in all scenarios.

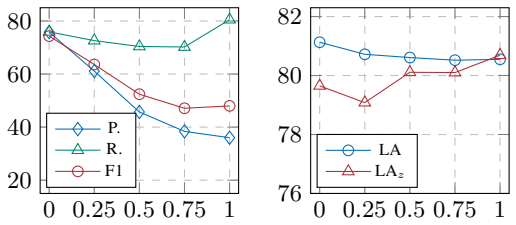
As reported in Table 4, after switching prior distributions, the model still performs well and learns logically consistent phrase veracity w.r.t. LA, LA_z and AGREE. For logical pseudo prior, LA_z and AGREE are better than NLI prior since there is gap between off-the-shelf NLI models and this task. But their scores on CULPA are close, proving similar culprit finding ability for both prior distributions. However, with uniform distribution, LOREN makes the *same* predictions for all claim phrases, which results in high CULPA recall (78.8%) but poor F1 scores due to its indistinguishability.

5.3 Evaluation of MRC Quality

The system should acquire enough distinguishable information to know the veracity of claim phrases. One of the key designs in LOREN is using an MRC model to retrieve evidential phrases for verifying claim phrases, i.e., constructing local premises. In this section, we evaluate the quality of the MRC model and its influence on culprit finding.

MRC Model	MRC Acc		
	`SUP`	`REF`	`NEI`
UnifiedQA (hit@1)	43.90	39.47	30.00
UnifiedQA (hit@4)	56.10	52.63	47.50
LOREN (hit@1)	**95.12**	78.95	83.75
LOREN (hit@4)	**95.12**	**89.47**	**87.50**

Table 5: Manual evaluation of the performance of MRC models. Hit@k denotes that we keep the top-k answers in the beam search as the candidates. The answer is accurate if any one of the k answers is correct.



(a) Results on CULPA. (b) Results on LA & LA_z.

Figure 2: Performance on culprit finding (CULPA) and verification (LA and LA_z) vs. the mask rate ρ of local premises, simulating the influence by deficiency of the MRC model.

Since there are no ground truth answers for `REF` and `NEI` claims, we *manually* evaluate the MRC model in LOREN, which is a $BART_{\text{base}}$ fine-tuned in a self-supervised way. The data samples are labeled as *correct* if they are the right answer(s) for verifying the claim phrase, otherwise *erroneous*.[4] We randomly selected 20 data samples per class for manual evaluation, with a total of 60 samples and 238 QA pairs (also 238 claim phrases). As a zero-shot baseline, we adopt UnifiedQA (Khashabi et al. 2020), which fine-tunes a $T5_{\text{base}}$ (Raffel et al. 2020) on existing QA tasks.

Results in Table 5 reveal the effectiveness of self-supervised training for adaptation and room for future refinement. Note that, different from traditional MRC tasks, the question can contain false information for non-`SUP` cases. Thus the accuracy drops as the question deteriorates. The results shed light on the *automatic correction* while performing verification (Thorne and Vlachos 2021) since the answers can serve as a correction proposal.

Influence of MRC Performance We further analyze the influence brought by the quality of the MRC model. To do so, we randomly mask the local premises at the rate of ρ (e.g., *Donald Trump won* `[MASK]`.), simulating the failure

[4]We extract the correct answers from evidence manually for evaluation. For `NEI` samples, there could be some claim phrases that do not have correct counterparts in the evidence. So we decide the MRC results for those phrases to be correct if the results are the same as claim phrases.

of the MRC model in an extreme situation. As seen in Figure 2, in general, the quality of local premises are critical for identifying the culprit phrases. In Figure 2(a), F1 score of CULPA quickly deteriorates as the quality of local premises gets worse. When mask rate reaches 100%, precision drops to 36.0% but recall hits 80.5%. This is because LOREN no longer identifies the culprit phrase and predicts all phrases to be the same, which is similar to the scenario of uniform prior distribution as discussed in §5.2. From Figure 2(b), we find claim verification ability (LA) of LOREN does not drop much, which is partly because the answers are already displayed in the evidence text. Also, the gap between LA_z and LA gradually narrows as mask rate ascends, because phrase verification degenerates into claim verification and makes the same predictions when local premises do not provide targeted information for claim phrases.

5.4 Case Study

We present three examples in Figure 3 to show the interpretability of LOREN. In the first, LOREN performs well in both claim and phrasal veracity predictions. It successfully finds the culprit phrase "*number three*", and a correction suggestion by MRC, i.e., "*number one*" in Premise 2.

In the second example, LOREN makes mistakes by predicting the veracity of the second phrase to be REF. The root causes for this mistake are complicated, including lack of commonsense knowledge and failure of the MRC and evidence retrieval modules. The MRC retrieves "*European*" (hit@1) for filling the masked "*Iranian*", whereas there is no definite answer to be drawn from the evidence. Strictly speaking, we can only know from the evidence text that *Ashley Cole* was born in England, but do not know whether he has dual citizenship or joined another country for certain. Therefore, we have *not enough information* (NEI) to draw the verdict, but LOREN predict it to be REF. However, the probability of NEI and REF for phrasal veracity prediction z_2 (0.466 vs. 0.520) and for claim veracity y_z (0.464 vs. 0.522) are rather close, which indicates that LOREN struggles to make that decision. These findings stress the usefulness and interpretability of the predicted phrase veracity z.

We investigate a multiple culprits scenario in the third example. The last three phrases in claim 3 could be seen as the culprits, and LOREN predicts "*nothing*" and "*Dorthy B. Hughes*" as REF. This corroborates that LOREN is by design capable of detecting multiple culprits in a claim.

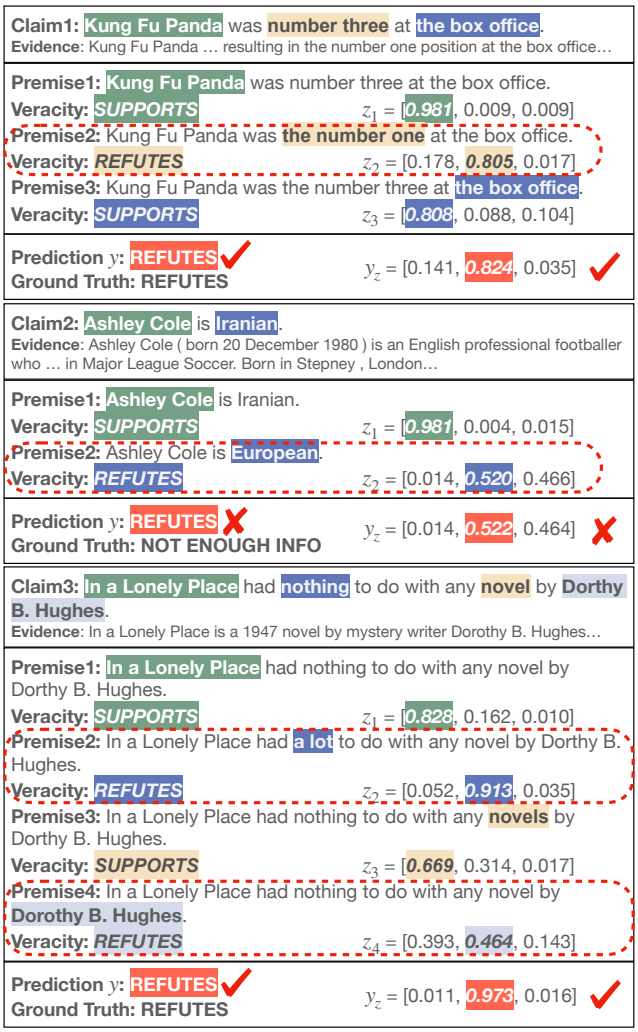


Figure 3: A case study of the interpretability of LOREN, where the probabilities in phrasal veracity prediction z_i are SUP, REF and NEI respectively.

6 Conclusion and Future Work

In this paper, we propose LOREN, an approach for interpretable fact verification by distilling the logical knowledge into the latent model. In the experiments, we find LOREN not only enjoys competitive performance with baselines but produces faithful and accurate phrase veracity predictions as explanations. Besides, the local premises constructed by the self-supervised MRC module are of high quality and deeply influence the finding of culprits, making LOREN's ability of automatic factual correction worthy of investigation in the future.

We add that, a general notion of culpability discovery in fact verification may depend on claim decomposition. A claim should be decomposed into fine-grained units where the culprit hides while making the units self-explanatory to humans. Besides phrases introduced in this paper, there could be other forms of decomposition units, e.g., dependency arc. We suggest future research focus on the limitations of LOREN, including decomposition, evidence retrieval, and out-of-domain issues. Accordingly, better solutions for these issues can improve LOREN's generality.

Acknowledgements

We thank Rong Ye, Jingjing Xu and other colleagues at ByteDance AI Lab as well as the anonymous reviewers for the discussions and suggestions for the manuscript. This work was supported by National Key Research and Development Project (No. 2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400) and Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

References

Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642. Lisbon, Portugal: Association for Computational Linguistics.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; et al. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced LSTM for Natural Language Inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1657–1668. Vancouver, Canada: Association for Computational Linguistics.

Chen, T.; Jiang, Z.; Poliak, A.; Sakaguchi, K.; and Van Durme, B. 2020. Uncertain Natural Language Inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8772–8779. Online: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. 2018. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 1–6. Melbourne, Australia: Association for Computational Linguistics.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *International Conference on Learning Representations*.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing Deep Neural Networks with Logic Rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2410–2420. Berlin, Germany: Association for Computational Linguistics.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jiang, K.; Pradeep, R.; and Lin, J. 2021. Exploring Listwise Evidence Reasoning with T5 for Fact Verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 402–410. Online: Association for Computational Linguistics.

Kang, D.; Khot, T.; Sabharwal, A.; and Hovy, E. 2018. Adversarial Training for Textual Entailment with Knowledge-Guided Examples. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. Melbourne, Australia.

Khashabi, D.; Min, S.; Khot, T.; Sabharwal, A.; Tafjord, O.; Clark, P.; and Hajishirzi, H. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1896–1907. Online: Association for Computational Linguistics.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Lamb, L. C.; d'Avila Garcez, A. S.; Gori, M.; Prates, M. O. R.; Avelar, P. H. C.; and Vardi, M. Y. 2020. Graph Neural Networks Meet Neural-Symbolic Computing: A Survey and Perspective. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4877–4884. ijcai.org.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Li, T.; Gupta, V.; Mehta, M.; and Srikumar, V. 2019. A Logic-Driven Framework for Consistency of Neural Models. In *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, 3924–3935. Hong Kong, China: Association for Computational Linguistics.

Liu, J.; Gardner, M.; Cohen, S. B.; and Lapata, M. 2020a. Multi-Step Inference for Reasoning Over Paragraphs. In *Proceedings of the 2020 Conference on EMNLP*, 3040–3050. Online: Association for Computational Linguistics.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, Z.; Xiong, C.; Sun, M.; and Liu, Z. 2020b. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7342–7351. Online: Association for Computational Linguistics.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. In *Advances in Neural Information Processing Systems*, 3749–3759.

Nie, Y.; Chen, H.; and Bansal, M. 2019. Combining Fact Extraction and Verification with Neural Semantic Matching Networks. In *AAAI*.

Qu, M.; and Tang, J. 2019. Probabilistic logic neural networks for reasoning. In *Advances in Neural Information Processing Systems*, 7712–7722.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Samarinas, C.; Hsu, W.; and Lee, M. L. 2021. Improving Evidence Retrieval for Automated Explainable Fact-Checking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, 84–91. Online: Association for Computational Linguistics.

Si, J.; Zhou, D.; Li, T.; Shi, X.; and He, Y. 2021. Topic-Aware Evidence Reasoning and Stance-Aware Aggregation for Fact Verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1612–1622. Online: Association for Computational Linguistics.

Sourek, G.; Aschenbrenner, V.; Zelezny, F.; and Kuzelka, O. 2015. Lifted relational neural networks. *arXiv preprint arXiv:1508.05128*.

Stammbach, D.; and Ash, E. 2020. e-fever: Explanations and summaries for automated fact checking. In *Proceedings of the 2020 Truth and Trust Online Conference (TTO 2020)*, 32. Hacks Hackers.

Thorne, J.; and Vlachos, A. 2021. Evidence-based Factual Error Correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 3298–3309. Online: Association for Computational Linguistics.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. New Orleans, Louisiana: Association for Computational Linguistics.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Wang, B.; Wang, X.; Tao, T.; Zhang, Q.; and Xu, J. 2020. Neural question generation with answer pivot. In *AAAI*, volume 34, 9138–9145.

Wang, W.; and Pan, S. J. 2020. Integrating Deep Learning with Logic Fusion for Information Extraction. In *AAAI*, 9225–9232.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Le Scao, T.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics.

Wu, L.; Rao, Y.; Lan, Y.; Sun, L.; and Qi, Z. 2021. Unified Dual-view Cognitive Model for Interpretable Claim Verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 59–68. Online: Association for Computational Linguistics.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, 5753–5763.

Ye, D.; Lin, Y.; Du, J.; Liu, Z.; Li, P.; Sun, M.; and Liu, Z. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on EMNLP*, 7170–7186. Online: Association for Computational Linguistics.

Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; and Choi, Y. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, 9051–9062.

Zhong, W.; Xu, J.; Tang, D.; Xu, Z.; Duan, N.; Zhou, M.; Wang, J.; and Yin, J. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6170–6180. Online: Association for Computational Linguistics.

Zhou, J.; Han, X.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 892–901. Florence, Italy: Association for Computational Linguistics.

Zhou, W.; Hu, J.; Zhang, H.; Liang, X.; Sun, M.; Xiong, C.; and Tang, J. 2020. Towards Interpretable Natural Language Understanding with Explanations as Latent Variables. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.