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

\[
V(c, W_c, E) = \top, \text{ iff } \forall w \in W_c, V(c, w, E) = \top \\
V(c, W_c, E) = \bot, \text{ iff } \exists w \in W_c, V(c, w, E) = \bot \\
V(c, W_c, E) = \ominus, \text{ iff } -V(c, W_c, E) = \top \\
\forall w \in W_c, V(c, w, E) = \{\top, \ominus\}
\]

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 $w_i \in W_c$ as a three-valued latent variable $z_i$. We define $z = (z_1, z_2, \ldots, z_{|W_c|})$. The veracity of a claim $y$ depends on the latent variables $z$. 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 $p_y(y|x)$ as below:

\[
p_y(y|x) = \sum_z p(y|z, x)p(z|x)
\]

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

\[
\mathcal{L}(\theta) = -\log p_y(y^*|x).
\]

Theoretically, we can adopt the EM algorithm for optimization. However, in our setting, it is difficult to compute the exact posterior $p_y(y|z, x)$ due to the large space of $z$. 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:

\[
-\mathbb{E}_{q_y(z|y, x)} \left[ \log p_y(y^*|z, x) \right] + D_{K L}(q_y(z|y, x) \parallel p(z|x))
\]

where $q_y(\cdot)$ is the variational posterior distribution conditioned on $y, x$, and $D_{K L}$ is Kullback–Leibler divergence. In experiments, we use an off-the-shelf and pre-trained NLI model as prior distribution $p(z|x)$, whose parameters are fixed. 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 $p_y(y|z, x)$ we intend to optimize. The teacher model is constructed by projecting variational distribution $q_y(z|y, x)$ into a subspace, denoted as $q^T_y(y|z, x)$. The subspace is constrained by the logical rules, since $y_z$ is the logical aggregation of $z$. Thus, simulating the outputs of $q^T_y$ serves to transfer logical knowledge into $p_y$. Formally, the distillation loss is formulated as:

\[
\mathcal{L}_{\text{logic}}(\theta, \phi) = D_{K L} \left(p_y(y|z, x) \parallel q^T_y(y|y, x, \mathbf{z})\right).
\]

\[\text{footnote}^1\text{We use a DeBERTa [He et al., 2021] fine-tuned on MNLI dataset (Bowman et al., 2015) as the NLI model.}\]
Overall, the final loss function is defined as the weighted sum of two objectives:

\[ L_{\text{final}}(\theta, \phi) = (1 - \lambda)L_{\text{val}}(\theta, \phi) + \lambda L_{\text{logic}}(\theta, \phi) \]  

(5)

where \( \lambda \) 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 \( z \), we logically aggregate them into \( y \), as follows (for simplicity, we drop the input \( x \)):

\[
\begin{align*}
q^T_\phi(y_z = \text{SUP}) &= \prod_{i=1}^{\lvert z \rvert} q_\phi(z_i = \text{SUP}) \\
q^T_\phi(y_z = \text{REF}) &= 1 - \prod_{i=1}^{\lvert z \rvert} (1 - q_\phi(z_i = \text{REF})) \\
q^T_\phi(y_z = \text{NEI}) &= 1 - q^T_\phi(y_z = \text{SUP}) - q^T_\phi(y_z = \text{REF})
\end{align*}
\]

(6)

where \( \sum_y q^T_\phi(y_z) = 1 \) and \( \sum_{z} q_\phi(z_i) = 1 \).

The prediction behavior of \( q^T_\phi \) 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 \( L_{\text{logic}} \) 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 perspectives 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 \( Q \) 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 T5base (Raffel et al. 2020) to generate interrogative questions.

#### Local Premise Construction

For every claim phrase \( w_i \in W_c \), we first generate probing question \( q_i \in Q \) with off-the-shelf question generators. The MRC model takes as an input \( Q \) and \( E \) and answers \( W_E \). Then, we replace \( w_i \in W_c \) with answers \( w_i' \in W_E \), yielding replaced claims \( c_i' \) such as “Donald Trump lost the 2020 election”, where \( w_i' = \text{"lost"} \) and \( w_i = \text{"won"} \). Such replaced claims are denoted as local premises \( \{c_i' | W_i\} \) 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 \( W_c \) 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 \( w_i' \in W_E \) for a claim phrase \( w_i \in W_c \), 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 \( p_\theta(\cdot) \) and \( q_\phi(\cdot) \) in the variational distribution \( q_\phi(z|y,x) \) 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 \( \{x_i^{(c)} = (c, c_i')\} \) and encode them into hidden representations \( \{h_i^{(c)} \in \mathbb{R}^d\} \). Similarly, we encode the claim and concatenated evidence sentences as \( x_i^{\text{global}} = (c, E) \) into the global vector \( h_i^{\text{global}} \in \mathbb{R}^d \), 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 \( h_i^{\text{local}} \) and \( h_i^{\text{global}} \) to determine the importance of the \( i \)-th claim phrase. We calculate the context vector \( h_i^{\text{local}} \) as follows:

\[
h_i^{\text{local}} = \tanh(\sum_{i=1}^{\lvert W_i \rvert} \alpha_i h_i^{(c)}); \alpha_i = \sigma(W_{\alpha} | h_i^{(c)}; h_i^{(\text{local})})
\]

(7)
During inference, werogate gradient (gradient of Gumbel Softmax), but backpropagate a surcally, we keep the argmax node and perform the usual for-Poole 2017) for discrete argmax operation from $z$. We use the Gumbel reparameterization (Jang, Gu, and

During training, $q_\phi(\cdot)$ and $p_\theta(\cdot)$ both to be two-layer MLPs, where the last layer is shared as label embeddings:

- $q_\phi(z_i|y,x)$ takes as input the concatenation of the label embeddings of $y$ (ground truth $y^*$ in training), $h_{\text{local}}^{(i)}$ and $h_{\text{global}}$, and outputs the probability of $z_i$. Note that $q_\phi(z_i|y,x) = \prod_i q_\phi(z_i|y,x)$.
- $p_\theta(y|z,x)$ takes as input the concatenation of $(z_1, z_2, \ldots, z_{\text{max}})$ (max length by padding), $h_{\text{global}}$ and $h_{\text{local}}$, and outputs the distribution of $y$.

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

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

### 4 Experiments

#### 4.1 Dataset and Evaluation Metrics

**Dataset** We evaluate our verification methods on a large-scale fact verification benchmark, i.e., FEVER 1.0 shared tasks (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.

| 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.

where $W_\alpha \in \mathbb{R}^{1 \times 2^d}$ is the parameter and $\sigma$ is the softmax function.

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

- **Logically aggregated Label Accuracy (LA$_z$):** We calculate the accuracy of logically aggregated $y_z$ by Eq.\textsuperscript{6} which evaluates the overall quality of explanations $z$.
- **Culprit finding Ability (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).\textsuperscript{2} CULPA calculates the Precision, Recall and F1 of the culprit finding based on discrete veracity from $z$.
- **Agreement (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.

\textsuperscript{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.
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 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 $x_{\text{global}}$ and $x_{\text{local}}$ 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. 2020) 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.

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 BERT_large and RoBERTa_large beats KGAT with RoBERTa_large and CorefRoBERTa_large, respectively.

We then perform a detailed analysis of the proposed components in LOREN (RoBERTa_large) on the development set to assess their influences on the performance and explanation quality.

5.2 Evaluation of Phrase Veracity

Table 3: 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.

| Model | Dev | Test |
|-------|-----|------|
|       | LA  | FEV  | LA  | FEV  |
| UNC NLP | 69.72 | 66.49 | 68.21 | 64.21 |
| GEAR (BERT_base) | 74.84 | 70.69 | 71.60 | 67.10 |
| DREAM (XLNet_large) | 79.16 | - | 76.85 | 70.60 |
| KGAT (BERT_large) | 77.91 | 75.86 | 73.61 | 70.24 |
| CorefRoBERTa | 79.29 | 76.11 | 74.07 | 70.38 |
| LOREN (BERT_large) | 78.44 | 76.21 | 74.43 | 70.71 |
| CorefRoBERTa_large | 81.14 | 78.83 | 76.42 | 72.93 |

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.

| $\lambda$ in $\mathcal{L}_{\text{final}}$ | LA  | $\mathcal{L}_{\text{final}}$ | AGREE |
|-------------------------------|------|-----------------------------|-------|
| $\lambda$ | LA | Hard | Soft | LA | Hard | Soft |
| $\lambda = 0.0$ | 81.0 | 79.8 | 51.99 | 51.92 | 54.02 | 53.90 |
| $\lambda = 0.2$ | 79.6 | 77.9 | 80.98 | 75.24 | 90.06 | 93.14 |
| $\lambda = 0.3$ | 81.0 | 75.4 | 80.69 | 79.66 | 92.94 | 96.11 |
| $\lambda = 0.5$ | 98.43 | 96.79 | 80.92 | 77.77 | 93.81 | 96.79 |
| $\lambda = 0.9$ | 98.43 | 96.79 | 80.28 | 80.28 | 91.56 | 98.43 |

We set $\lambda = 0.5$ by default in our experiments.
As seen in Table 3, we report the results of LA, LA\textsubscript{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 faithfulness. 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 more accurate according to LA\textsubscript{z} and LA scores. 3) With the increase of \( \lambda \) and stronger regularization of \( \mathcal{L}_{\text{logic}} \), the general accuracy and faithfulness of phrase veracity increase. Without \( \mathcal{L}_{\text{logic}} \), LORE\textsubscript{N} 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 Table 3, we use the results of a fixed, off-the-shelf NLI model [He et al. 2021] as the prior distribution \( p(z) \). 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\textsubscript{z} score at only 53.41%. However, with LORE\textsubscript{N} training, LA\textsubscript{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 \( p(z) \) and sample 1 or 2 phrases as the culprit phrase(s) based on culprit attention weight in Eq. 7 and label them as REF and the rest as non-REF. Such \( p(z) \) is in accordance with the logic but distinguishable of the phrase(s);

2. **uniform distribution,** which is commonly used as \( p(z) \).

\( z \) 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\textsubscript{z}, and AGREE. For logical pseudo prior, LA\textsubscript{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, LORE\textsubscript{N} makes the same predictions for all claim phrases, which results in high CULPA recall (78.8%) but poor F1 scores due to its indistinguishability.

### Table 4: Results of different choices of prior distribution \( p(z) \) during training, where \( y_z \) in LA\textsubscript{z} is calculated using soft logic.

| Choice of \( p(z) \) | LA       | LA\textsubscript{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 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.

| MRC Model       | SUP Acc | REF Acc | NEI Acc |
|-----------------|---------|---------|---------|
| UnifiedQA (hit@1) | 43.90   | 39.47   | 30.00   |
| UnifiedQA (hit@4) | 56.10   | 52.63   | 47.50   |
| LORE\textsubscript{N} (hit@1) | 95.12   | 78.95   | 83.75   |
| LORE\textsubscript{N} (hit@4) | 95.12   | 89.47   | 87.50   |

### 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 LORE\textsubscript{N} 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.

Since there are no ground truth answers for REF and NEI claims, we manually evaluate the MRC model in LORE\textsubscript{N}, which is a BART\textsubscript{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 erro- nous. 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\textsubscript{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 \( \rho \) (e.g., Donald Trump won \{MASK\}), simulating the failure 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 LORE\textsubscript{N} 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 LORE\textsubscript{N} does not drop.

---

4We 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.
Figure 2: Performance on culprit finding (CULPA) and verific-
ification (LA and LA$_z$) vs. the mask rate $r$ of local premises,
simulating the influence by deficiency of the MRC model.

much, which is partly because the answers are already dis-
played 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 tar-
ged information for claim phrases.

5.4 Case Study

We present three examples in Figure 3 to show the inter-
pretability of LORENE. In the first, LORENE performs well
in both claim and phrasal veracity predictions. It success-
fully finds the culprit phrase “number three”, and a corre-
cision suggestion by MRC, i.e., “number one” in Premise 2.

In the second example, LORENE makes mistakes by pre-
dicting 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 ev-

dence 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 Ash-
ley 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 LORENE 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 $y_L$ (0.464 vs.
0.522) are rather close, which indicates that LORENE
struggles to make that decision. These findings stress the useful-
ness and interpretability of the predicted phrase veracity $z$.

We investigate a multiple culprits scenario in the third ex-
ample. The last three phrases in claim 3 could be seen as
the culprits, and LORENE predicts “nothing” and “Dorothy B.
Hughes” as REF. This corroborates that LORENE is by de-
sign capable of detecting multiple culprits in a claim.

6 Conclusion and Future Work

In this paper, we propose LORENE, an approach for inter-
pretable fact verification by distilling the logical knowledge
into the latent model. In the experiments, we find LORENE
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 LORENE’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., depen-
dency arc. We suggest future research focus on the limi-
tations of LORENE, including decomposition, evidence re-

trieval, and out-of-domain issues. Accordingly, better solu-
tions for these issues can improve LORENE’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 List-wise 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
A Evidence Retrieval Results

For the sake of completeness, we describe here the commonly adopted method for evidence retrieval. Given a claim sentence, the evidence retrieval system first identifies entity phrases in the sentence and then searches for Wikipedia pages with the given entity names (e.g., Donald Trump). It further selects related sentences from the retrieved Wikipedia pages based on a neural sentence ranking model.

Since we focus on the second sub-task, we here describe the evidence retrieval techniques adopted in baselines. To illustrate the effectiveness of these methods, we present the reported results of these methods in Table 6 based on the top-5 retrieved evidence sentences per claim. According to Table 6, ER-KGAT, and ER-DREAM are consistently better than ER-ESIM, and they are comparable on the blind test set, yet ER-KGAT outperforms ER-DREAM on the development set. By default, we use evidence retrieved using ER-KGAT in LOREN for the following experiments.

| ER Method | Prec@5 | Rec@5 | F1@5 |
|-----------|--------|-------|------|
| Dev       |        |       |      |
| ER-ESIM   | 24.08  | 86.72 | 37.69|
| ER-DREAM  | 26.67  | 87.64 | 40.90|
| ER-KGAT   | 27.29  | 94.37 | 42.34|
| Test      |        |       |      |
| ER-ESIM   | 23.51  | 84.66 | 36.80|
| ER-DREAM  | 25.63  | 85.57 | 39.45|
| ER-KGAT   | 25.21  | 87.47 | 39.14|

Table 6: Reported performance of evidence retrieval (ER) strategies on Precision@5, Recall@5 and F1@5.

Performance with Oracle Evidence Retrieval Recall that LOREN focuses on the fact verification sub-task and uses evidence sentences retrieved from another system. Even with the relatively good MRC module as reported in Table 5, the LOREN pipeline still suffers from the initial evidence retrieval error. What if LOREN uses perfect evidence without information losses from evidence retrieval system? To answer this question, we fill claims in the development set with ground truth evidence sentences and excluding the 1/3 NEI cases since they do not have oracle evidence. Results in Table 7 show somehow the upper bound of LOREN, which also suggests a viable direction for future improvements.

| LOREN       | LA | FEV |
|-------------|----|-----|
| w/ ER-KGAT  | 86.12 | 82.66 |
| w/ ER-Oracle| 88.92 | 88.62 |

Table 7: Verification results of LOREN using retrieved evidence (ER-KGAT) and oracle evidence (ER-Oracle). We exclude NEI cases since they do not have corresponding evidence.