# Claim-Dissector: An Interpretable Fact-Checking System with Joint Re-ranking and Veracity Prediction

**Anonymous ACL submission**

## Abstract

We present Claim-Dissector: a novel latent variable model for fact-checking and analysis, which given a claim and a set of retrieved evidences jointly learns to identify: (i) the relevant evidences to the given claim, (ii) the veracity of the claim. We propose to disentangle the per-evidence relevance probability and its contribution to the final veracity probability in an interpretable way — the final veracity probability is proportional to a linear ensemble of per-evidence relevance probabilities. In this way, the individual contributions of evidences towards the final predicted probability can be identified. In per-evidence relevance probability, our model can further distinguish whether each relevant evidence is supporting (S) or refuting (R) the claim. This allows to quantify how much the S/R probability contributes to the final verdict or to detect disagreeing evidence.

Despite its interpretable nature, our system achieves results competetive with state-of-the-art on the FEVER dataset, as compared to typical two-stage system pipelines, while using significantly fewer parameters. Furthermore, our analysis shows that our model can learn fine-grained relevance cues while using coarse-grained supervision, and we demonstrate it in 2 ways. (i) We show that our model can achieve competitive sentence recall while using only paragraph-level relevance supervision. (ii) Traversing towards the finest granularity of relevance, we show that our model is capable of identifying relevance at the token level. To do this, we present a new benchmark TLR-FEVER focusing on token-level interpretability — humans annotate tokens in relevant evidences they considered essential when making their judgment. Then we measure how similar are these annotations to the tokens our model is focusing on. The code and dataset will be released.

## 1 Introduction

Today's automated fact-checking systems are moving from predicting the claim's veracity by capturing the superficial cues of credibility, such as the way the claim is written, the statistics captured in the claim author's profile, or the stances of its respondents on social networks (Zubiaga et al., 2016; Derczynski et al., 2017; Gorrell et al., 2019; Fajcik et al., 2019; Li et al., 2019) towards evidence-grounded systems which, given a claim, identify relevant sources and then use these to predict the claim's veracity (Thorne et al., 2018; Jiang et al., 2020; Park et al., 2022). In practice, providing precise evidence turns out to be at least as important as predicting the veracity itself. Disproving a claim without linking it to factual evidence often fails to be persuasive and can even cause a "backfire" effect — refreshing and strengthening the belief into an erroneous claim (Lewandowsky et al., 2012)[1].

For evidence-grounded fact-checking, most of the existing state-of-the-art systems (Jiang et al., 2021; Stammbach, 2021; Khattab et al., 2021) employ a 3-stage cascade approach; given a claim, they retrieve relevant documents, rerank relevant evidences (sentences, paragraphs or larger text blocks) within these documents, and predict the claim's veracity from the top-$k$ (usually $k$=5) relevant evidences.

This comes with several drawbacks; firstly, *the multiple steps of the system lead to error propagation*, i.e. the input to the last system might often be too noisy to contain any information. Some previous work focused on merging evidence reranking and veracity prediction into a single step (Ma et al., 2019; Schlichtkrull et al., 2021). Secondly, in open-domain setting, *number of relevant evidences can be significantly larger than $k$*[2], especially when there is a lot of repeated evidence. Thirdly, in open-domain setting, *sometimes there is both, supporting and refuting evidence*. The re-ranking systems often do not distinguish whether evidence is relevant because it supports or refutes the claim, and thus

[1] Further discussion in Appendix F.

[2] e.g.,~3.7% of FEVER's non-exhaustive annotations.

may select the evidence from one group based on the in-built biases.

To further strengthen the persuasive effect of the evidences and understand the model's reasoning process, some of these systems provide cues of interpretability (Popat et al., 2018; Liu et al., 2020). However, the interpretability in the mentioned work was often considered a useful trait, which was evaluated only qualitatively, as the labor-intensive human evaluation was out of the scope of their focus.

To this extent, we propose Claim-Dissector (CD), a latent variable model which:

1. jointly ranks top-relevant, top-supporting and top-refuting evidences, and predicts veracity of the claim in an interpretable way, where the probability of the claim's veracity is estimated using the linear combination of per-evidence probabilities (Subsection 2.2),
2. can provide fine-grained (sentence-level or token-level evidence), while using only coarse-grained supervision (on block-level or sentence-level respectively),
3. can be parametrized from a spectrum of language representation models (such as RoBERTa or DeBERTaV3 (Liu et al., 2019; He et al., 2021)).

Finally, we collect a 4-way annotated dataset `TLR-FEVER` of per-token relevance annotations. This serves as a proxy for evaluating interpretability: we measure how similar are the cues provided by the model to the ones from humans. We believe future work can benefit from our quantitative evaluation approach, while maintaining focus.

## 2 Model Description

We present a 2-stage system composed of the *retriever* and the *verifier*. The documents are ranked via retriever. Each document is split into blocks. The blocks from top ranking documents are passed to verifier and jointly judged. Our interpretable CD verifier is capable of re-ranking documents for any granularity of relevant evidence (e.g., document, block, sentence, token). Jointly, the same model predicts the claim's veracity. The overall schema of our approach is depicted in Figure 1.

### 2.1 Retriever

Given a claim $c \in \mathcal{C}$ from the set of all possible claims $\mathcal{C}$ and the corpus $\mathcal{D} = \{d_1, d_2, ..., d_n\}$ composed of documents $d_i$, the retriever produces a ranking using function $\text{rank} : \mathcal{C} \times \mathcal{D} \rightarrow \mathbb{R}$ that assigns a claim-dependent score to each document in the corpus. In this work, we focus on the verifier; therefore, we take the strong retriever from Jiang et al. (2021). This retriever interleaves documents ranked by BM25 (Robertson and Zaragoza, 2009) $(a_1, a_2, ...a_n)$ and Wikipedia API $(b_1, b_2, ...b_m)$ following Hanselowski et al. (2018) as $(a_1, b_1, a_2, b_2, ...)$, while skipping duplicate articles. Each document is then split into non-overlapping blocks of size $L_x$, respecting sentence boundaries[3]. Our verifier then computes its veracity prediction from top-$K_1$ such blocks. To keep up with similar approaches (Hanselowski et al., 2018; Stammbach and Neumann, 2019), we also experiment with expanding evidence with documents hyperlinked to the top retrieved articles. We rank these documents according to rank and sequential order in the document they were hyperlinked from. We then process these extra ranked documents the same way as retrieved documents, adding top-$K_2$ blocks to the verifier's input. As discussed more closely in Stammbach and Neumann (2019), some relevant documents are impossible to retrieve using just claim itself, as their relevance is conditioned on other relevant documents. However, we stress

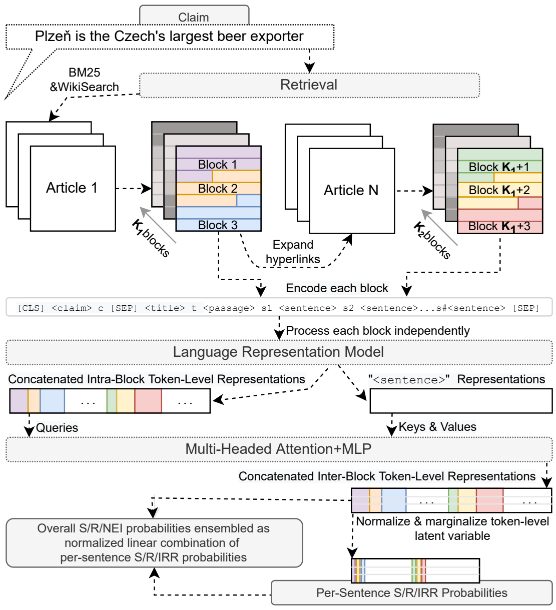


Figure 1: Diagram of Claim-Dissector's workflow. Abbreviations S, R, IRR, NEI stand for support, refute, irrelevant, not-enough-information. MLP function from the figure is defined by equation 1.

[3]Every block contains as many sentences as can fit into $L_x$ tokens, considering verifier's tokenization.

that such approaches also mimic the way FEVER dataset was collected, and thus the improvements of such approach on "naturally collected" datasets might be negligible if any.

### 2.2 Verifier

The verifier first processes each block independently using a language representation model (LRM) and then aggregates cross-block information via multi-head attention (Vaswani et al., 2017), computing matrix $\boldsymbol{M}$. This matrix is used to compute both, the probability of each evidence's relevance and the probability of the claim's veracity. Furthermore, the way the model is constructed allows learning a linear relationship between these probability spaces.

Formally given a claim $c$ and $K = K_1 + K_2$ blocks, $K$ input sequences $x_i$ for each block $i$ are constructed as

[CLS] <claim> $c$ [SEP] <title> $t$
<passage> $s_1$ <sentence> $s_2$
<sentence>...$s_\#$<sentence> [SEP],

where [CLS] and [SEP] are transformer special tokens used during the LRM pre-training (Devlin et al., 2019). Each block is paired with its article's title $t$ and split into sentences $s_1, s_2, ..., s_\#$. Symbols $c, t, s_1, s_2, ..., s_\#$ thus each denote a sequence of tokens. We further introduce new special tokens <claim>, <title>, <passage>, <sentence> to separate different input parts. Crucially, every sentence is appended with a <sentence> token. Their respective embeddings are trained from scratch. Each input $x_i$ is then encoded via LRM $\boldsymbol{E}_i = \mathrm{LRM}(x_i) \in \mathbb{R}^{L_B \times d}$, where $L_B$ is an input sequence length, and $d$ is LRM's hidden dimensionality. The representations of every block are then concatenated into $\boldsymbol{E} = [\boldsymbol{E}_1; \boldsymbol{E}_2; ...; \boldsymbol{E}_K] \in \mathbb{R}^{L \times d}$, where $L$ is the number of all tokens in the input sequences from all retrieved blocks. Then we index-select all representations from $\boldsymbol{E}$ corresponding to positions of sentence tokens in $s_1, s_2, ..., s_\#$ into score matrix $\boldsymbol{E_s} \in \mathbb{R}^{L_e \times d}$, where $L_e$ corresponds to the number of all tokens in all input sentences (without special tokens). Similarly, we index-select all representations at the same positions as the special <sentence> tokens at the input from $\boldsymbol{E}$ into matrix $\boldsymbol{S} \in \mathbb{R}^{L_S \times d}$, where $L_S \ll L_e$ is the total number of sentences in all inputs $x_i$. The matrix $\boldsymbol{M} \in \mathbb{R}^{L_e \times 3}$ is then given as

$$\boldsymbol{M} = \mathrm{SLP}(\mathrm{MHAtt}(\boldsymbol{E}_s, \boldsymbol{S}, \boldsymbol{S}))\boldsymbol{W}. \quad (1)$$

The $\mathrm{MHAtt} : \ldots \rightarrow \mathbb{R}^{L_e \times d}$ operator is a multi-head attention with queries $\boldsymbol{E}_s$, and keys and values $\boldsymbol{S}$. The SLP operator is a single layer perceptron with layer norm, dropout, and GeLU activation (details in Appendix D) and $\boldsymbol{W} \in \mathbb{R}^{d \times 3}$ is a linear transformation, projecting resulting vectors to the desired number of classes (3 in case of FEVER).

To compute the per-evidence probabilities we split the matrix $\boldsymbol{M}$ according to tokens belonging to each evidence. For instance, for sentence-level evidence granularity we do split $\boldsymbol{M} = [\boldsymbol{M}^{s_1,1}; \boldsymbol{M}^{s_2,1}; ...; \boldsymbol{M}^{s_\#,K}]$ along dimension $L_e$ into submatrix representations corresponding to sentence $s_1$ in block 1 up to last sentence $s_\#$ in block $K$. We then independently normalize each such matrix of $i$-th evidence of $j$-th block as[4]:

$$\mathrm{P}^{i,j}(w, y) = \frac{\exp \boldsymbol{M}^{i,j}_{w,y}}{\sum_{w'} \sum_{y'} \exp \boldsymbol{M}^{i,j}_{w',y'}}. \quad (2)$$

Note that $w \in \{1, 2, ..., |s_{i,j}|\}$ is a token index in the (i,j)-th evidence and $y \in \{\text{S, R, NEI}\}$ is the class label. Then we marginalize over latent variable $w$ to obtain the marginal log-probability per evidence.

$$\log \mathrm{P}^{i,j}(y) = \log \sum_{w'} \mathrm{P}^{i,j}(y, w') \quad (3)$$

Then objective $\mathcal{L}_R$ is computed for evidences annotated in label set $\mathbb{A} = \{(y_1^*, (i_1, j_1)), ...\}$ (usually $|\mathbb{A}| \ll L_S$) for a single claim[5].

$$\mathcal{L}_R = \frac{1}{|\mathbb{A}|} \sum_{y^*,(i,j) \in \mathbb{A}} \log \mathrm{P}^{i,j}(y^*) \quad (4)$$

In training, $\mathbb{A}$ contains the same amount of relevant and irrelevant labels. For relevant, the log-probability $\log \mathrm{P}^{i,j}(y = y^*)$ is maximized, based the overall claim's veracity label $y^* \in \{S, R\}$. For irrelevant evidences, $y^* = \text{IRR}$ is maximized. As FEVER contains only annotation of relevant sentences, we follow the heuristic of Jiang et al. (2021) and sample irrelevant sentences ranked between 50 and 200, in order to avoid maximizing the objective for false negatives. In test-time, we rank the evidence $(i, j)$ according to its combined probability of supporting or refuting relevance $score_{i,j} = \sum_{y \in \{S,R\}} \mathrm{P}^{i,j}(y)$.

[4] Note that the probability also depends on input sequences $\{x_i\}_{i \in \{1,2,...,K\}}$, but we omit this dependency for brevity.

[5] If example has NEI veracity in FEVER, $\mathcal{L}_R = 0$.

Next, we compute the probability of the claim's veracity $y \in \{\text{S, R, NEI}\}$. First notice that scores in $\boldsymbol{M}$ are logits (proof in Appendix O)

$$\boldsymbol{M}^{i,j}_{w,y} = \log(C^{i,j}\,\mathrm{P}^{i,j}(w,y)). \tag{5}$$

Therefore, we use a learnable extra non-negative degree of freedom $C^{i,j}$ to compute a linear ensemble[6] producing the final probability

$$\mathrm{P}(y) = \frac{\sum_{i,j,w} C^{i,j}\,\mathrm{P}^{i,j}(w,y)}{\sum_{y'}\sum_{i,j,w} C^{i,j}\,\mathrm{P}^{i,j}(w,y')}. \tag{6}$$

Lastly, we bias the model to focus only on some tokens in each evidence by enforcing an $\mathcal{L}_2$ penalty over the scores in $\boldsymbol{M}$ by

$$\mathcal{L}_2 = \frac{1}{L_e}||\boldsymbol{M}||_F^2, \tag{7}$$

where $||\cdot||_F$ denotes Frobenius norm. We show empirically that this objective leads to significantly better weakly-supervised token-level interpretability (Section 5). Therefore the final per-sample loss with hyperparameters $\lambda_R$, $\lambda_2$ is

$$\mathcal{L} = -\log \mathrm{P}(y) - \lambda_R\mathcal{L}_R + \lambda_2\mathcal{L}_2. \tag{8}$$

### 2.3 Baseline

Apart from previous work, we propose a baseline bridging the proposed system and the recent work of Schlichtkrull et al. (2021). In order to apply this recent work for FEVER, we introduce a few necessary modifications[7]. We normalize all scores in $\boldsymbol{M}$ to compute joint probability across all blocks

$$\mathrm{P}(w,y) = \frac{\exp \boldsymbol{M}_{w,y}}{\sum_{w'}\sum_{y'}\exp \boldsymbol{M}_{w',y'}}. \tag{9}$$

First, we marginalize out per-token probabilities in each evidence $s_{i,j}$.

$$\mathrm{P}(s_{i,j},y) = \sum_{w' \in s_{i,j}} \mathrm{P}(w',y) \tag{10}$$

Using this sentence probability formulation, the objective is computed for every relevant evidence.

$$\mathcal{L}_{b0} = \frac{1}{|\mathbb{A}_p|}\sum_{s_{i,j},y \in \mathbb{A}_p} \log \mathrm{P}(s_{i,j},y) \tag{11}$$

Second, unlike Schlichtkrull et al. (2021), we interpolate objective $\mathcal{L}_{b0}$ with objective

$$\mathcal{L}_{b1} = \log \mathrm{P}(y) = \log \sum_{s_{i,j}} \mathrm{P}(s_{i,j},y) \tag{12}$$

by computing their mean. Like CD, we use $\mathcal{L}_{b1}$ objective to take advantage of examples from NEI class for which we have no annotation in $\mathbb{A}_p$ (and thus $\mathcal{L}_{b0}$ is virtually set to 0). Unlike CD, the annotations $\mathbb{A}_p$ in $\mathcal{L}_{b0}$ contain only relevant labels where $y^* \in \{S,R\}$[8].

In order to not penalize non-annotated false negatives, we compute global distribution in $\mathcal{L}_{b0}$ during training only from representations of tokens from labeled positive and negative sentences in $\boldsymbol{M}$. In test time, we rank evidences according to $score_{i,j} = \sum_{y \in \{S,R\}} \mathrm{P}(s_{i,j},y)$, and predict claim's veracity according to $\mathrm{P}(y) = \sum_{s_{i,j}} \mathrm{P}(s_{i,j},y)$. We also considered different model parametrizations discussed in Appendix E.

### 2.4 Transferring Supervision to Higher Language Granularity

The proposed model can benefit from annotation on the coarser granularity of the language than tested. For example, evidence annotation can be done at the document, block, paragraph, or token level. In Section 5, we show despite the fact that the model is trained on coarse granularity level, the model still shows moderate performance of relevance prediction when evaluated on finer granularity. We demonstrate this with two experiments.

First, *the model is trained with sentence-level supervision and it is evaluated on a token-level annotation.* For this we leave model as it is — reminding that prior over per-token probabilities enforced by the objective $\mathcal{L}_2$ is crucial (Table 4).

Secondly, *we assume only block-level annotation is available in training and we evaluate on sentence-level annotation*. Here, we slightly alter the model, making it rely more on its sentence-level representations. In Section 5, we show this simple alteration significantly improves the performance at sentence-level. To compute block-level probability, the block is the evidence, therefore the evidence index can be dropped. The probability of the $j$-th block $b_j$ is obtained by marginalizing out the per-token/per-sentence probabilities.

[6] Assuming $y$=IRR=NEI.

[7] The necessity of these is explained in Appendix P.

[8] Maximizing NEI class for irrelevant sentences leads to inferior accuracy. This makes sense, since it creates "tug-of-war" dynamics between $\mathcal{L}_{b0}$ and $\mathcal{L}_{b1}$. The former objective tries to allocate mass of joint space in NEI class, since most documents are irrelevant, whereas the latter objective tries to allocate the mass in the dimension of labeled veracity class.

$$\mathrm{P}(b_j, y) = \sum_{s_{i,j} \in b_j} \mathrm{P}(s_{i,j}, y) = \sum_{s_{i,j} \in b_j} \sum_{w' \in s_{i,j}} \mathrm{P}(w', y) \tag{13}$$

In practice, we found it helpful to replace the block-level probability $\mathrm{P}(b_j, y)$ with its lower-bound $\mathrm{P}(s_{i,j}, y)$ computed for 1 sentence sampled from the relevant sentence likelihood.

$$\mathrm{P}(b_j, y) \approx \mathrm{P}(s_{i,j}, y); s_{i,j} \sim \mathrm{p}(\boldsymbol{s}_{i,j}, y \in \{S, R\}) \tag{14}$$

Intuitively, making a single sentence estimate (SSE) forces the model to invest the mass into a few sentence-level probabilities. This is similar to HardEM[9]. In $\mathcal{L}_R$ we then maximize the probabilities of positive blocks computed as in equation 14, and negative sentences[10] computed (and normalized) on sentence level as in equation 4.

#### 2.4.1 Baseline for Token-level Rationales

Similarly to Shah et al. (2020); Schuster et al. (2021), we train a masker — a model which learns to replace the least amount of token embeddings at the Claim-Dissector's input with a single learned embedding in order to maximize the NEI class probability. We compare the unsupervised rationales given by the masker with the unsupervisedly learned rationales provided by the Claim-Dissector on-the-fly. Our masker follows the same architecture as Claim-Dissector. We provide an in-depth description of our masker model and its implementation in Appendix G.

## 3 Related Work

**Datasets.** Previous work in supervised open-domain fact-checking often focused on large datasets with evidence available in Wikipedia such as FEVER (Thorne et al., 2018), FEVER-KILT (Petroni et al., 2021), FAVIQ (Park et al., 2022), HoVer (Jiang et al., 2020) or TabFact (Chen et al., 2020). We follow this line of work and selected FEVER because of its sentence-level annotation, 3 levels of veracity (into S/R/NEI classes), and controlled way of construction — verification should not require world knowledge, everything should be grounded on trusted, objective, and factual evidence from Wikipedia.

**Open-Domain Fact-Checking (ODFC)** Unlike this work, most of the previous work includes 3-stage systems that retrieve evidence, rerank each document independently, and then make a veracity decision from top-$k$ documents (Thorne et al., 2018; Nie et al., 2019; Zhong et al., 2020).

Jiang et al. (2021) particularly distinguished the line of work which aggregates the final decision from independently computed per-sentence veracity probabilities (Zhou et al., 2019; Soleimani et al., 2020; Pradeep et al., 2021b, *inter alia*) and the line of work where the top-relevant sentences are judged together to compute the final veracity probability (Stammbach and Neumann, 2019; Pradeep et al., 2021a, *inter alia*). Jiang et al. (2021) compares similar system against these two assumptions, showing that joint judgment of relevant evidence is crucial when computing final veracity. We stress that our system falls into joint judgment category. Although relevance is computed per sentence, these probabilities along with linear combination coefficients are computed jointly, with the model conditioned on hundreds of input sentences.

To deal with multi-hop evidence (evidence which is impossible to mark as relevant without other evidence) Subramanian and Lee (2020); Stammbach (2021) iteratively rerank evidence sentences to find minimal evidence set, which is passed to verifier. Our system jointly judges sentences within a block, while multi-head attention layer could propagate cross-block information. Our overall peformance results suggest that our system is about on-par with these iterative approaches, while requiring only single forward computation. However, further analysis shows our model underperforms on multi-hop evidence (more in Section 5).

**Interpretability** Popat et al. (2018); Liu et al. (2020) both introduced systems with an interpretable attention design and demonstrated their ability to highlight important words through a case study. In our work, we take a step further and propose a principled way to evaluate our system quantitatively. We note that Schuster et al. (2021) proposed a very similar quantitative evaluation of token-level rationales, for data from the VitaminC dataset. The dataset, constructed from factual revisions on Wikipedia, assumed that the revised part of facts is the most salient part of the evidence. In contrast, we instruct annotators to manually anno-

[9] In preliminary experiments, we also tried HardEM, but the results over multiple seeds were unstable.

[10] Indices of irrelevant sentences are mined automatically (see Section 2.1), therefore this supervision comes "for free".

tate terms important to their judgment (Section 4.1). The VitaminC dataset is not accompanied by the evidence corpus, thus we deemed it as unsuitable for open-domain knowledge processing.

Krishna et al. (2022) proposed a system that parses evidence sentences into natural logic based inferences (Angeli and Manning, 2014). These provide deterministic proof of the claim's veracity. Authors verify the interpretability of the generated proofs by asking humans to predict veracity verdict from them. However, the model is evaluated only on FEVER dataset and its derivatives, which contain potential bias to their approach — the claims in this dataset were created from fact through "mutations" according to natural logic itself.

**Joint Reranking and Veracity Prediction** Schlichtkrull et al. (2021) proposed a system similar to our work for fact-checking over tables. The system computes a single joint probability space for all considered evidence tables. The dataset however contains only claims with true/false outcomes, typically supported by a single table. While our work started ahead of its publication, it can be seen as an extension of this system.

# 4 Experimental Setup

Unless said otherwise, we employ DeBERTaV3 (He et al., 2021) as LRM. In all experiments, we firstly pre-train model on MNLI (Williams et al., 2018). We use maximum block-length $L_x = 500$. Our recipe for implementation and model training is closely described in Appendix J.

## 4.1 Datasets

**FEVER.** We validate our approach on FEVER (Thorne et al., 2018) and our newly collected dataset of token-level rationales. FEVER is composed from claims constructed from Wikipedia. Each annotator was presented with an evidence sentence, and first sentence of articles from hyperlinked terms. In FEVER, examples in development set contain multi-way annotation of relevant sentences, i.e., each annotator selected set of sentences (evidence group) they considered relevant. To analyze performance of our components on samples that need multi-hop reasoning, we further created subsets of training/development set. FEVER$_{MH}$ contains only examples where all annotators agreed on that more than 1 sentence is required for verification, whereas FEVER$_{MH_{ART}}$ contains only examples, where all annotators agreed that *sentences from different articles* are required for verification. As majority of examples of FEVER$_{MH}$ are from FEVER$_{MH_{ART}}$, we only evaluate on FEVER$_{MH}$. We include the subset statistics in Table 1.

| | FEVER | FEVER$_{MH}$ | FEVER$_{MH_{ART}}$ |
|---|---|---|---|
| **Train** | 145,449 | 12,958 (8.91%) | 11,701 (8.04%) |
| **Dev** | 19,998 | 1204/19998 (6.02%) | 1059/19998 (5.30%) |
| **Test** | 19,998 | - | - |

Table 1: FEVER dataset and its subsets.

**TLR-FEVER** To validate token-level rationales, we collect our own dataset on random subset of validation set (only considering examples with gold sentence annotation). We collect 4-way annotated set of token-level rationales. The annotators were the colleagues with NLP background from our lab. We instruct every annotator via written guidelines, and then we had 1-on-1 meeting after annotating a few samples, verifying that contents of the guidelines were understood correctly. We let annotators annotate 100 samples, resolve reported errors manually, obtaining 94 samples with fine-grained token-level annotation. In guidelines, we simply instruct annotators to *highlight minimal part of text they find important for supporting/refuting the claim. There should be such part in every golden sentence (unless annotation error happened).* The complete guidelines are available in Appendix I.

To establish performance of average annotator, we compute the performance of each annotator compared to other annotators on the dataset, and then compute the average annotator performance. We refer to this as *human baseline lower-bound*, as each annotator was compared to only 3 annotations, while the system is compared to 4 annotations. We measure performance via $F_1$ metric.

**Other Datasets.** We further validate our approach on FAVIQ-A: a more realistic dataset for fact-checking (Appendix B), where it achieves state-of-the-art results and HoVer (Appendix M), where we demonstrate its limitations on multi-hop evidence.

## 4.2 Evaluation

**Accuracy (A).** The proportion of correctly classified samples, disregarding the predicted evidence.

**Recall@5 (R@5).** The proportion of samples for which any annotated evidence group is matched within top-5 ranked sentences.

**FEVER-Score (FS).** The proportion of samples for which (i) any annotated evidence group is

| | System | FS | A | R@5 | HA | #$\theta$ |
|---|---|---|---|---|---|---|
| Development Set | TwoWingOS (Yin and Roth, 2018) | 54.3 | 75.9 | 53.8 | × | ? |
| | HAN (Ma et al., 2019) | 57.1 | 72.0 | 53.6 | ✓ | ? |
| | UNC (Nie et al., 2019) | 66.5 | 69.7 | 86.8 | ✓ | 408M |
| | HESM (Subramanian and Lee, 2020) | 73.4 | 75.8 | 90.5 | ✓ | 39M |
| | KGAT[OR] (Liu et al., 2020) | 76.1 | 78.3 | 94.4 | × | 465M |
| | DREAM (Zhong et al., 2020) | - | 79.2 | 90.5 | ✓? | 487M |
| | T5 (Jiang et al., 2021) | 77.8 | **81.3** | 90.5 | × | 5.7B |
| | LF+$D_{XL}$ (Stammbach, 2021) | - | - | 90.8 | × | 1.2B |
| | $LF_{2-iter}$+$D_{XL}$ (Stammbach, 2021) | - | - | **93.6** | ✓ | 1.2B |
| | ProofVer-MV (Krishna et al., 2022) | 78.2 | 80.2 | - | ✓ | 515M |
| | ProofVer-SB (Krishna et al., 2022) | **79.1** | 80.7 | **93.6** | ✓ | 765M |
| | $\text{Baseline}_{joint}$ | 75.2 | 79.8 | 90.0 | × | 187M |
| | $\text{Claim-Dissector}_{RoBERTa}$ | 74.6 | 78.6 | 90.4 | × | 127M |
| | $\text{Claim-Dissector}_{RoBERTaL}$ | 75.1 | 79.1 | 90.6 | × | 360M |
| | $\text{Claim-Dissector}_{RoBERTaL}$ \w HE | 76.1 | 79.4 | 91.7 | ✓ | 360M |
| | Claim-Dissector | 76.2 | 79.5 | 91.5 | × | 187M |
| | Claim-Dissector \w HE | 76.9 | 79.8 | 93.0 | ✓ | 187M |
| | $\text{Claim-Dissector}_{L}$ | 76.9 | 80.4 | 91.8 | × | 439M |
| | $\text{Claim-Dissector}_{L}$ \w HE | <u>78.0</u> | <u>80.8</u> | <u>93.3</u> | ✓ | 439M |
| | $\text{Claim-Dissector}_{L}$ \w HE [OR] | 78.9 | 81.2 | 94.7 | ✓ | 439M |
| Test Set | KGAT (Liu et al., 2020) | 70.4 | 74.1 | - | × | 465M |
| | DREAM (Zhong et al., 2020) | 70.6 | 76.9 | - | ✓? | 487M |
| | HESM (Subramanian and Lee, 2020) | 71.5 | 74.6 | - | ✓ | 58M |
| | ProofVer-MV (Krishna et al., 2022) | 74.4 | 79.3 | - | ✓ | 515M |
| | T5 (Jiang et al., 2021) | 75.9 | 79.4 | - | × | 5.7B |
| | $LF_{2-iter}$+$D_{XL}$ (Stammbach, 2021) | 76.8 | 79.2 | - | ✓ | 1.2B |
| | ProofVer-SB (Krishna et al., 2022) | **76.8** | **79.5** | - | ✓ | 765M |
| | $\text{Claim-Dissector}_{RoBERTaL}$ | 73.1 | 76.4 | - | × | 360M |
| | $\text{Claim-Dissector}_{RoBERTaL}$ \w HE | 74.3 | 77.8 | - | ✓ | 360M |
| | $\text{Claim-Dissector}_{L}$ | 74.7 | 78.5 | - | × | 439M |
| | $\text{Claim-Dissector}_{L}$ \w HE | <u>76.5</u> | <u>79.3</u> | - | ✓ | 439M |

Table 2: Performance on dev and test splits of FEVER. #$\theta$ denotes number of parameters in the model. Model names suffixed with [OR](as Oracle) inject missing golden evidence into its input. Models using any kind of hyperlink-augmentation (HA) are marked. Our models with hyperlink expansion are suffixed with (\w HE). **Overall best** and <u>our best</u> result are in bold and underlined respectively (disregarding oracle results).

| | FEVER | | | $\text{FEVER}_{MH}$ | | |
|---|---|---|---|---|---|---|
| System | FS | A | R@5 | FS | A | R@5 |
| $\text{CD}_{LARGE}$ \w HE [OR] | 78.9 | 81.2 | 94.8 | 50.3 | 81.9 | 58.9 |
| $\text{CD}_{LARGE}$ \w HE | 78.0 | 80.8 | 93.4 | 44.7 | 81.2 | 53.1 |
| CD \w HE | 76.9 | 79.8 | 93.2 | 41.3 | 80.8 | 49.9 |
| CD \w HE \wo MH | 76.5 | 79.5 | 93.0 | 41.7 | 80.8 | 50.2 |
| Baseline | 75.2 | 79.8 | 90.0 | 28.9 | 80.9 | 34.7 |
| CD | 76.2 | 79.6 | 91.7 | 30.0 | 79.2 | 36.4 |
| CD \wo $\mathcal{L}_2$ | 76.0 | 79.7 | 91.6 | 30.4 | 79.5 | 36.2 |
| CD \wo VC | - | - | 91.9 | - | - | 37.8 |
| CD \wo RC | - | 79.9 | - | - | 81.5 | - |

Table 3: Ablation Study. Minor differences to Table 2 are caused by different early-stopping (Appendix J).

matched within top-5 ranked sentences, and (ii) the correct class is predicted.

**$F_1$ Score** measures unigram overlap between predicted tokens and reference tokens, disregarding articles. Having multiple references, the maximum $F_1$ between prediction and any reference is considered per-sample. Our implementation closely follows Rajpurkar et al. (2016).

In practice, both CD and masker model infer continuous scores capturing relevance for every token[11]. When evaluating $F_1$, we select only tokens with scores greater than threshold $\tau$. We tune the optimal threshold $\tau$ w.r.t. $F_1$ on TLR-FEVER.

## 5 Results

We report results of base-sized models based on 3-checkpoint average. We train only a single large model. We further evaluate retrieval in Appendix H as it is a non-essential part of our contribution.

[11]We consider mask-class logits as scores for masker.

**Performance.** We compare the performance of our system with previous work in Table 2. Results marked with ? were unknown/uncertain, and unconfirmed by authors. We note that, apart from HAN (Ma et al., 2019), all previous systems were considering two separate systems for reranking and veracity prediction. Next, we note that only ProofVer system uses additional data. It leverages rewritten-claim data for fact-correction paired with original FEVER claims (Thorne and Vlachos, 2021).

We observe that (i) even our base-sized RoBERTa-based CD model outperforms base-sized HESM on dev data, and its large version outperforms large-sized KGAT, DREAM and HESM on test data, (ii) our base sized DebertaV3-based CD model matches large-based DREAM and even KGAT with oracle inputs on dev set, (iii) model version with hyperlink expansion (suffixed \w HE) further improves the overall performance, (iv) using larger model improves mostly its accuracy, (v) $\text{Claim-Dissector}_{L}$ \w HE achieves better FEVER score than T5-based approach (with two 3B models) and better accuracy than LongFormer+DebertaXL, but it is not pareto optimal to these previous SOTA, (vi) our model is outmatched by recent ProofVer-SB, though it is more efficient as ProofVer-SB requires two rounds of reranking and autoregressive decoding. We still consider this a strong feat, as our system was focusing on modeling reranking and veracity prediction jointly in an interpretable way. Finally, we inject blocks with missing golden evidence into inputs of $\text{Claim-Dissector}_{L}$ \w HE at random positions and measure oracle performance. We observe that items missed by retrieval are still beneficial to the performance.

**Ablations.** We ablate components of Claim-Dissector (CD) in Table 3. Firstly, we resort to single-task training. We drop veracity classification (VC) objective $\log \mathrm{P}(y)$ or relevance classification (RC) objective $\mathcal{L}_R$ from the loss. We observe an

| System | F1 |
|---|---|
| Select All Tokens | 52 |
| Select Claim Overlaps | 63 |
| Masker | 71 |
| Claim-Dissector \wo $\mathcal{L}_2$ | 60 |
| Claim-Dissector | 77 |
| Human Performance LB | 85 |

Table 4: Token-level relevance on TLR-FEVER.

| Model | FS | A | R@5 |
|---|---|---|---|
| Full Supervision | 76.2 | 79.5 | 91.5 |
| Block Supervision | 65.5 | 76.9 | 77.8 |
| Block Supervision + SSE | 69.7 | 78.1 | 83.0 |

Table 5: Sentence-level performance on FEVER dev set under different kinds of supervision.

overall trend — single-task model performs slightly better to multi-task model. The advantages of multi-task model, however, lie in its efficiency and ability to provide explanations between per-evidence relevances and final conclusion. Next, we observe that dropping the $\mathcal{L}_2$ objective doesn't affect the performance significantly. Further, we study the effect of hyperlink expansion (HE) and the effect of multi-head (MH) attention layer. As expected, hyperlink expansion dramatically increases performance on FEVER$_{MH}$. The multi-head attention also brings marginal improvements to results on FEVER. However, contrary to our expectations, there is no effect of the MH layer on FEVER$_{MH}$, the improvements happened in non-multihop part. Additional experiments with CD on HoVer dataset (Appendix M) confirm this, CD does not work well on examples with cross-article multihop evidence. Improving cross-article reasoning was not our aim, and we leave the investigation to our future research.

**Transferring sentence-level supervision to token-level performance.** We evaluated the performance of token-level rationales[12] on our dataset in Table 4. We considered two baselines. The first was to select all tokens in golden evidences (Select All Tokens). The second was to select only tokens that overlap with claim tokens (Select Claim Overlaps). We found that our model with weakly-supervised objective produces token-level rationales significantly better[13] than the masker — a separate model trained explicitly to identify tokens important to model's prediction. Furthermore, the results also demonstrate the importance of $\mathcal{L}_2$ objective. However, human performance is still considerably beyond the performance of our approach.

**Transferring block-level supervision to sentence-level performance.** The performance of our model on the sentence-level evidences is evaluated in Table 5. We notice that even our vanilla Claim-Dissector trained with block supervision reaches competitive recall@5 on sentence-level. However, adding SSE from equation 14 leads to further improvements both in recall, but also in accuracy. We expected the recall will be improved, because model now focuses on assigning high probability mass only to some sentences within block, since high-entropy of the per-sentence distribution would be penalized in loss. However, we have not foreseen the damaging effect on accuracy, which block-level supervision causes. Interestingly, the accuracy without any evidence supervision reported in last row of Table 3 was increased.

[12] We visualized predicted token-level rationales on 100 random dev set examples in supplementary material.

[13] See Appendix K for our F1 significance testing protocol.

## 6 Conclusion

In this work, we proposed Claim-Dissector, an interpretable probabilistic model for fact-checking and fact-analysis. Our model jointly predicts the supporting/refuting evidence and the claim's veracity. It achieves state-of-the-art results, while providing three layers of interpretability. Firstly, it identifies salient tokens important for the final prediction. Secondly, it allows disentangling ranking of relevant evidences into ranking of supporting evidence and ranking of refuting evidence. This allows detecting bipolar evidence without being exposed to such bipolar evidence sets during training (Appendix C). Thirdly, it combines the per-token relevance probabilities via linear combination into final veracity assessment. Therefore it can be identified to what extent the relevance of each token/sentence/block/document contributes to final assessment. Conveniently, this allows to differentiate between the concept of evidence relevance and its contribution to the final assessment. Our work was however limited in experiments with these coefficients, and we would like to analyze what they can learn, and how to inject features, such as satire assessment or source trustworthiness, through these coefficients in our future work.

Finally, it was shown that a hierarchical structure of our model allows making predictions on even finer language granularity, than the granularity the model was trained on. We believe the technique we proposed is transferable beyond fact-checking.

## Ethical Considerations

We built the system with the intended use for fact-checking support, providing rationales at various level to user for ease of understanding. These rationales include supporting and refuting evidences in the corpus. We see potential misuse of system might lie in spreading of fake news and propaganda by e.g., automatic detection of sources, which support or refute certain claims from the narrative. This could be followed by subsequent glorification/discreditation of statements in these sources. This could influence the human population, but also poison retrieval databases of similar fact-checking systems, influencing their decisions. Future work in this direction, such as Du et al. (2022), needs to study disinformation attacks and how to prevent them.

## References


Gabor Angeli and Christopher D. Manning. 2014. NaturalLI: Natural logic inference for common sense reasoning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 534–545, Doha, Qatar. Association for Computational Linguistics.

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243, Seattle, United States. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact : A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yibing Du, Antoine Bosselut, and Christopher D. Manning. 2022. Synthetic disinformation attacks on automated fact verification systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10581–10589.

Martin Fajcik, Pavel Smrz, and Lukas Burget. 2019. BUT-FIT at SemEval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1097–1104, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 103–108.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 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)*, pages 402–410, Online. Association for Computational Linguistics.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460, Online. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. *Advances in Neural Information Processing Systems*, 34.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. Proofver: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

Stephan Lewandowsky, Ullrich KH Ecker, Colleen M Seifert, Norbert Schwarz, and John Cook. 2012. Misinformation and its correction: Continued influence and successful debiasing. *Psychological science in the public interest*, 13(3):106–131.

Quanzhi Li, Qiong Zhang, and Luo Si. 2019. eventAI at SemEval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 855–859, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Jing Ma, Wei Gao, Shafiq Joty, and Kam-Fai Wong. 2019. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2561–2571, Florence, Italy. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 6859–6866.

Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. FaVIQ: FAct verification from information-seeking questions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5154–5166, Dublin, Ireland. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 22–32, Brussels, Belgium. Association for Computational Linguistics.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021a. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021b. Vera: Prediction techniques for reducing harmful misinformation in consumer health search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2066–2070.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. 2021. Joint verification and reranking for open fact checking over tables. 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)*, pages 6787–6799, Online. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643, Online. Association for Computational Linguistics.

Colleen M Seifert. 2002. The continued influence of misinformation in memory: What makes a correction effective? In *Psychology of learning and motivation*, volume 41, pages 265–292. Elsevier.

Darsh Shah, Tal Schuster, and Regina Barzilay. 2020. Automatic fact-guided sentence modification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8791–8798.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Dominik Stammbach. 2021. Evidence selection as a token-level prediction task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 14–20, Dominican Republic. Association for Computational Linguistics.

Dominik Stammbach and Guenter Neumann. 2019. Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 105–109, Hong Kong, China. Association for Computational Linguistics.

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical Evidence Set Modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809, Online. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 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)*, pages 3298–3309, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 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)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Joseph E Uscinski and Ryden W Butler. 2013. The epistemology of fact checking. *Critical Review*, 25(2):162–180.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wenpeng Yin and Dan Roth. 2018. TwoWingOS: A two-wing optimization strategy for evidential claim verification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 105–114, Brussels, Belgium. Association for Computational Linguistics.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6170–6180, Online. Association for Computational Linguistics.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901, Florence, Italy. Association for Computational Linguistics.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

# A Limitations

By manual analysis, we found that claim-dissector suffers from overconfidence in blocks with at least 1 relevant evidence. Then it seeks to select more relevant evidences inside, even when they are not. We believe this is connected to how irrelevant negatives are mined in FEVER — they originate only from blocks without relevant evidences.

On real data, the system often struggles to recognize what facts are refuting, and what are irrelevant (especially when applied out-of-domain). We demonstrate this in a case study on downstream application, where we replaced retrieval on Wikipedia with news-media in test-time. We tried to verify the claim "*Weapons are being smuggled into Estonia*". Our system discovered article with facts about "*Weapons being smuggled into Somalia*", and used it as a main refuting evidence to predict `REFUTE` veracity.

Lastly, CD is trained with evidence from Wikipedia, and do not considers other factors important for relevance assessment in practice, such as credibility of source, its reliability, or its narrative. This is the area of active research, as human fact-checkers also need to deal with lies (Uscinski and Butler, 2013).

# B Performance on FAVIQ-A

To asses more realistic performance of our system, we study its performance on FAVIQ-A (Park et al., 2022). We use the silver passage supervision from Asai et al. (2022), and feed the model with top-20 passages retrieved via DPR system (Karpukhin et al., 2020). We keep all the hyperparameters same as for FEVER, and use dev set only for early-stopping. We compare to evidentiality-guided generator (EGG), a t5-based Fusion-in-Decoder (FiD) (Izacard and Grave, 2021) with two decoders from Asai et al. (2022). We use hyperparameters from FEVER. The results are shown in Table 6. Our DebertaV3-based Claim-Dissector reaches state-of-the-art results on the dataset. The domain mismatch (measured by difference $\Delta$) between development and test set is likely caused by the domain shift of NaturalQuestions test set, from which FAVIQ's test set was created (see Appendix B in Min et al. (2020)). However, despite our best efforts, we have not uncovered the cause of massive degradation between dev and test set for `roberta-base` based Claim-Dissector (the standard deviation on test set was only $\pm 0.4$ accuracy points).[14]

| | Test | Dev | $\Delta$ | $\theta$ |
|---|---|---|---|---|
| BART$_{LARGE}$ (Park et al., 2022) | 64.9 | 66.9 | 2.0 | 374M |
| FiD (Asai et al., 2022) | 65.7 | 69.6 | 3.9 | 336M |
| CD$_{RoBERTa}$ | 58.6 | 69.8 | 11.2 | 127M |
| CD$_{RoBERTaL}$ | 66.9 | 73.3 | 6.4 | 360M |
| CD | 69.8 | 76.3 | 6.5 | 187M |
| CD$_{LARGE}$ | 72.0 | 79.7 | 7.7 | 439M |

Table 6: Performance on FAVIQ-A.

# C Detection of examples with bipolar evidence.

We manually analyzed whether we can take advantage of model's ability to distinguish between evidence, which is relevant because it supports the claim, and the evidence which is relevant because it refutes the claim. To do so, we try to automatically detect examples from the validation set, which contain both, supporting and refuting evidence (which we refer to as bipolar evidence). We note that there were no examples with explicitly annotated bipolar evidence in the training data.

We select all examples where model predicted at least $0.9$ probability for any supporting and any refuting evidence. We found that out of 72 such examples, $66\%(48)$ we judged as indeed having the bipolar evidence[15]. We observed that about half (25/48) of these examples had bipolar evidence because of the entity ambiguity caused by open-domain setting. E.g., claim "*Bones is a movie*" was supported by sentence article "*Bones (2001 film)*" but also refuted by sentence from article "*Bones (TV series)*" and "*Bone*" (a rigid organ).

[14] Author's note: *We would like to add these results to the 9-page version of the main paper, if reviewers would agree.*

[15] Annotations are available as supplementary material.

## D Structure of Single-layer Perceptron

Given a vector $\boldsymbol{x}$, the structure of single-layer perceptron from equation 1 is the following:

$$SLP(\boldsymbol{x}) = \mathrm{GELU}(\mathrm{dp}(\boldsymbol{W'}\,\mathrm{lnorm}(\boldsymbol{x}))). \quad (15)$$

The operator $\mathrm{dp}$ denotes the dropout (Srivastava et al., 2014) used in training, $\boldsymbol{W'}$ is a trainable matrix, GELU is the Gaussian Error Linear Unit (Hendrycks and Gimpel, 2016) and $\mathrm{lnorm}$ is the layer normalization (Ba et al., 2016).

## E Experiments with Different Model Parametrizations

Apart from parametrizations provided in the main paper, we experimented with several different parametrizations described below. We keep the training details the same as for our baseline (Section 2.3). Starting off with a baseline system formulation, we will consider replacing $L_{b0}$ with different objective functions.

$$\mathcal{L}_{b2} = \frac{1}{|\mathbb{A}|} \sum_{s_{i,j}, y \in \mathbb{A}} \log \mathrm{P}(s_{i,j}, y) \quad (16)$$

With $L_{b2}$, the annotation set $\mathbb{A}$ contains both relevant and irrelevant annotations. We found in practice this does not work - recall@5 during training stays at 0. This makes sense since if annotation exists, the final class is likely support or refute. Drifting the probability mass towards NEI for irrelevant annotations is adversarial w.r.t. total veracity probability.

$$\mathcal{L}_{b3} = \log \sum_{s_{i,j}, y \in \mathbb{A}_p} \mathrm{P}(s_{i,j}, y) \quad (17)$$

Instead of maximizing the multinomial probability, $L_{b3}$ objective marginalizes over relevant annotations.

$$\mathcal{L}_{b4} = \log \sum_{s_{i,j} \in \mathbb{A}_p} \sum_{y} \mathrm{P}(s_{i,j}, y) \quad (18)$$

Additionally to $L_{b3}$, $L_{b4}$ also marginalizes out the class label $y$.

The results in Table 7 reveal only minor differences. Comparing $L_{b3}$ and $L_{b4}$, marginalizing out label improves the accuracy, but damages the recall. Baseline parametrization achieves best accuracy but the worst recall. Claim-Dissector seems to work the best in terms of FS, but the difference to $L_{b3}$ is negligible, if any.

| | FEVER | | |
|---|---|---|---|
| **Model** | **FS** | **A** | **R@5** |
| CD | 76.2 | 79.5 | 91.5 |
| Baseline | 75.2 | 79.8 | 90.0 |
| $L_{b3}$ | 76.0 | 79.0 | 91.2 |
| $L_{b4}$ | 75.7 | 79.7 | 90.4 |

Table 7: Different model parametrizations.

## F The Continued Influence Effect: Retractions Fail to Eliminate the Influence of Misinformation

Lewandowsky et al. (2012) summarizes research paradigm, which focuses on credible retractions in neutral scenarios, in which people have no reason to believe one version of the event over another. In this paradigm, people are presented with a factious report about an event unfolding over time. The report contain a target piece of information (i.e. a claim). For some readers, the claim is retracted, whereas for readers in a control condition, no correction occurs. Then the readers are presented with a questionnare to assess their understanding of the event and the number of clear and uncontroverted references to the claim's veracity.

In particular, a stimulus narrative commonly used within this paradigm involves a warehouse fire, that is initially thought to have been caused by gas cylinders and oil paints there were negligently stored in a closet. A proportion of participants is then presented with a retraction such as "the closet was actually empty". A comprehension test follows, and number of references to the gas and paint in response to indirection inference questions about the event (e.g., "What caused the black smoke?") is counted.

Research using this paradigm has consistently found that retractions rarely, if ever, had the intended effect of eliminating reliance on misinformation, even when people remember the retraction, when later asked. Seifert (2002) have examined whether clarifying the correction might reduce the continued influence effect. The correction in their studies was strengthened to include the phrase "paint and gas were never on the premises". Results showed that this enhanced negation of the presence of flammable materials backfired, making people even more likely to rely on the misinformation in their responses. Some other additions to the correction were found to mitigate to a degree, but not eliminate, the continued influence effect. For instance, when participants were given a rationale about how misinformation originated, such as "a truckers' strike prevented the expected delivery of the items", they were less likely to make references to it. Even so, the influence of the misinformation could still be detected. The conclusion drawn from studies on this phenomenon show that it is extremely difficult to return the beliefs to people who have been exposed to misinformation to a baseline similar to those of people who have never been exposed to it. We recommend reading Lewandowsky et al. (2012) for broader overview of *the misinformation and its correction.*

## G Masker

**Model Description.** Our masker follows same DeBERTaV3 architecture as Claim-Dissector, except that the multiheaded layer from the equation (1) is omitted. It receives $K_1$ blocks at its input, encoded the very same way as for the Claim-Dissector. Instead of computing matrix $\boldsymbol{M}$—which contains three logits per evidence token, the masker predicts two logits $[l_0^i, l_1^i]$ — corresponding to keep/mask probabilities $[p_0^i, p_1^i]$ for $i$-th token in evidence of every block. The mask $[m_0^i, m_1^i]$ is then sampled for every token from concrete distribution via Gumbel-softmax (Jang et al., 2017). During training, $i$-th token embedding $e_i$ at the Claim-Dissector's input $e_i'$ is replaced with a linear combination of itself and a learned mask-embedding $e_m \in \mathbb{R}^d$, tuned with the masker.

$$e_i' = m_0^i e_i + m_1^i e_m \tag{19}$$

The masker is trained to maximize the Claim-Dissector's log-likehood of NEI class, while forcing the mask to be sparse via L1 regularization. Per-sample objective to maximize with sparsity strength hyperparameter $\lambda_S$ is given as

$$\mathcal{L} = \log \mathrm{P}(y = NEI) - \frac{\lambda_S}{L_e} \sum_i |m_0^i|. \tag{20}$$

**Training Details.** We keep most hyperparameters the same as for Claim-Dissector. The only difference is learning rate $2e-6$, adaptive scheduling on Gumbel-softmax temperature $\tau$ and training model/masker on different dataset split. Training starts with temperature $\tau = 1$ and after initial 100 steps, it is linearly decreasing towards $\tau = 0.1$ at step 700. Then we switch to hard Gumbel-softmax — sampling 1-hot vectors in forward pass, while computing gradients as we would use a soft sample with $\tau = 0.1$ at backward pass. We randomly split training set and we train model on 75% of data, and masker on remaining 25% of data.

| $K_1$+$K_2$ | FEVER | FEVER$_{MH}$ | FEVER$_{MH_{ART}}$ | #SaI |
|---|---|---|---|---|
| 35+0 | 94.2 | 52.0 | 45.8 | 239.9 |
| 100+0 | 95.1 | 58.5 | 53.1 | 649.4 |
| 35+10 | 95.2 | 61.9 | 57.0 | 269.6 |
| 35+20 | 95.9 | 69.0 | 65.2 | 309.0 |
| 35+30 | 96.7 | 77.5 | 74.7 | 388.6 |
| 35+35 | 97.5 | 84.1 | 82.3 | 506.7 |
| 35+50 | 97.7 | 86.5 | 85.0 | 624.3 |
| 35+100 | 98.4 | 93.0 | 92.4 | 1008.8 |
| 100+100 | 98.6 | 93.4 | 92.7 | 1431.0 |

Table 8: Retrieval performance in RaI on FEVER dev set and its subsets.

## H Retrieval Performance

We evaluate the retrieval method from Jiang et al. (2021) and the proposed hyperlink expansion method in Table 8. We use two metrics:

**Recall@Input (RaI).** We evaluate retrieval w.r.t. recall at model's input while considering different amount of $K_1$+$K_2$ blocks at the input, i.e. the score hit counts iff any annotated evidence group was matched in $K_1$+$K_2$ input blocks.

**Number of Sentences@Input (#SaI)** denotes average number of sentences at model's input under corresponding $K_1 + K_2$ setting.

We focus on analyzing the effect of hyperlink expansion, varying $K_2$, while keeping $K_1 = 35$ in most experiments, which is setting similar to previous work — Jiang et al. (2021) considers reranking top-200 sentences. We observe that setting $K_1 + K_2 = 35 + 10$ already outperforms retrieval without hyperlink expansion and $K_1 = 100$ blocks. Such observation is thus consistent with previous work which used hyperlink signal (Hanselowski et al., 2018; Stammbach and Neumann, 2019).

| $K_1$ | Recall | Recall$_{MH}$ | Recall$_{MH_{ART}}$ | #SaI |
|---|---|---|---|---|
| 10 | 90.4 | 40.1 | 33.0 | 68.8 |
| 20 | 93.4 | 48.0 | 41.5 | 132.9 |
| 30 | 94.1 | 51.3 | 45.0 | 196.8 |
| 35 | 94.2 | 52.0 | 45.8 | 239.9 |
| 50 | 94.5 | 54.3 | 48.4 | 325.4 |
| 100 | 95.1 | 58.5 | 53.1 | 649.4 |

Table 9: Retrieval performance on FEVER dev set.

An in-depth evaluation of retrieval method adopted from Jiang et al. (2021) is available in Table 9.

# I Token-level Annotation Guidelines

**Annotation Guidelines**
Welcome to the “Pilot annotation phase” and thank you for your help!
**How to start annotate**
If you haven’t done so, simply click on "Start Annotation" button, and the annotation will start.
**Annotation process & Guidelines**

- In pilot annotation, we are interested in annotator’s disagreement on the task. So whatever disambiguity you will face, do not contact the organizers but judge it yourself.

- Your task is to annotate 100 samples. In each case, you will be presented with list of sentences divided by | character. The sentences do not need to (and often do not) directly follow each other in text. Be sure that in each case you:

- read the claim (lower-right corner)

- read metadata - to understand the context, you also have access to other metadata (lower-right corner), such as
  - titles - Wikipedia article names for every sentence you are presented with, split with character |,
  - claim_label - Ground-truth judgment of the claim’s veracity.

- **highlight minimal part of text you find important for supporting/refuting the claim. There should be such part in every sentence (unless annotation error happened). PLEASE DO NOT ANNOTATE ONLY WHAT IS IMPORTANT IN THE FIRST SENTENCE.**

- Use **"RELEVANT"** annotation button highlight the selected text spans.

- In some cases, you can find **errors in the ground-truth judgment**, in other words, either document is marked as supported and it should be refuted according to your judgment or vice-versa. If you notice so, please annotate any part of the document with **CLAIM_ERROR** annotation.

- In case you would like to comment on some example, use comment button (message icon). If the comment is example specific, please provide specific example’s id (available in-between metadata).

**FAQ**

- *The example does not contain enough information to decide whether it should be supported or refuted. Should I label it as a CLAIM_ERROR?*
  No. In such case, please annotate parts of the input, which are at least partially supporting or refuting the claim. Please add comment to such examples. If there are no such input parts, only then report the example as CLAIM_ERROR.

**That is it. Good luck!**

# J Experimental Details

We base our implementation of pre-trained language representation models on Huggingface (Wolf et al., 2019). Unless said otherwise, we employ DeBERTaV3 (He et al., 2021) as LRM. In all experiments, we firstly pre-train model on MNLI (Williams et al., 2018). While we observed no significant improvement when using a MNLI-pretrained checkpoint, we found that without MNLI pretraining, our framework sometimes converges to poor performance. We train model on FEVER with minibatch size 64, learning rate $5e{-}6$, maximum block-length $L_x = 500$. We schedule linear warmup of learning rate for first 100 steps and then keep constant learning rate. We use Adam with decoupled weight decay (Loshchilov and Hutter, 2017) and clip gradient vectors to a maximal norm of 1 (Pascanu et al., 2013). In all experiments, the model is trained and evaluated in mixed-precision. We keep $\lambda_R = \lambda_2 = 1$. We use 8x Nvidia A100 40GB GPUs for training. We validate our model every $500$ steps and select best checkpoint according to FEVER-Score (see Subsection 4.2). We have not used any principled way to tune the hyperparameters.

To train model with SSE, we decrease the strength of block-level supervised $\mathcal{L}_R$ objective to $\lambda_R = 0.7$. We switch between vanilla objective and SSE objective randomly on per-sample basis. Training starts with replace probability $p_{sse} = 0$.

for first $1,000$ steps. The probability is then linearly increased up to $p_{sse} = 0.95$ on step $3,000$, after which it is left constant.

All results except for Table 3 and Table 4 were early-stopped based on the best FS. For Table 3, we report best result for each metric early-stopped independently, to be comparable with ablations where FS was not available. For Table 4, we record best F1 during training.

## K Statistical Testing on F1 Measure

To compare CD with masker in F1, we follow Goutte and Gaussier (2005), sum TPs, FPs, FNs across the dataset, estimate recall (R) and precision (P) posteriors, and sample F1 distributions. To obtain sample of average F1 from multiple checkpoints, we estimate the P and R posteriors for each checkpoint separately, sample F1 for each checkpoint and then average these. We estimate $p \approx \mathrm{P}(\mathrm{F1}_a > \mathrm{F1}_b)$ via Monte-Carlo, and consider significance level at $p > 0.95$.

## L Development Performance in Training

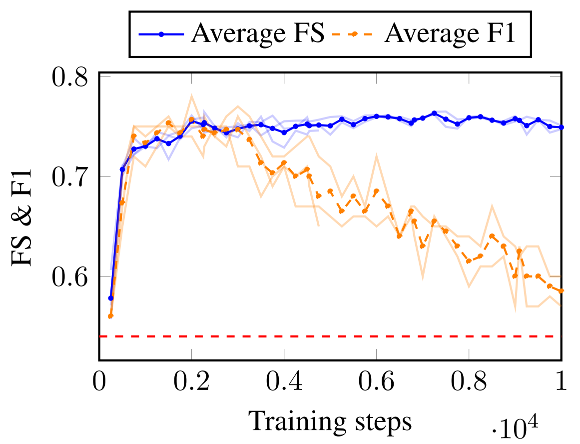


Figure 2: Average FEVER-Score (FS) and F1 performance on dev sets during training. Red dashed horizontal line marks the F1 performance when selecting all tokens (Select All Tokens) from Table 4. Opaque lines show the performance of individual checkpoints.

In Figure 2, we analyze how the performance on FEVER-Score and F1 changes over the course of training on FEVER and TLR-FEVER sets. We find our scores reach the top performance and then quickly detoriate. It is thus necessary do the early stopping on both, the performance metric and the interpretability metric.

| | Hops | Accuracy | EM |
|---|---|---|---|
| Baleen | 2 | - | 47.3 |
| | 3 | - | 37.7 |
| | 4 | - | 33.3 |
| | All | 84.5 | 39.2 |
| CD | 2 | 81.3 | 48.0 |
| | 3 | 80.1 | 16.9 |
| | 4 | 78.1 | 7.7 |
| | All | 79.9 | 23.3 |

Table 10: Results on HoVer dataset (dev split).

## M Multihop Experiments on HoVer

To study how well our model can deal with claims, which require multihop information, we trained our system on HoVer (Jiang et al., 2020). In particular, we follow the recipe for Baleen (Khattab et al., 2021) and retrieve $4\times25$ top articles using official quantized Baleen implementation[16] (which achieves about 2% lower retrieval@100 on supported samples than reported in paper). We split 5 starting documents from each iteration into blocks, padding input with further documents from first retrieval iteration when necessary. We keep input size at $K_1 = 35$, and we do not use hyperlink expansion. We compute the probability of not-supported class by summing NEI and REFUTE classes. Furthermore, we assume simplified conditions, we infuse inputs with oracle information when necessary (achieving RaI  100%) and predict as many evidences, as there was annotated. We refer reader to Khattab et al. (2021) for further information about setup and evaluation metrics.

Nevertheless, our system lags behind Baleen on 3 and 4 hop examples, as shown in Table 10. We hyphothesize that, similarly to Baleen, autoregressive process is necessary to match its performance. We leave the question of interpretable multi-hop fact-checking with Claim-Dissector open for our future work.

## N Example of Predicted Rationales

Interpretable refuting example (available between samples from supplementary material) is available in Figure 3. Example shows a top-6 refuting sentences ranked by their refuting relevance probability $\mathrm{P}^{i,j}(y = R)$. Each sentence is prefixed with its Wikipedia article title, refuting relevance probability (RS) and prediction score (PS). Prediction score

[16] https://github.com/stanford-futuredata/Baleen

American Sniper_RS(0.97)_PS[0.91]
-- American Sniper is a 2014 American biographical war drama film directed by Clint Eastwood and written by Jason Hall
American Sniper_RS(0.95)_PS[0.96]
-- The film follows the life of Kyle, who became the deadliest marksman in U.S. military history with 255 kills from four tours in the Iraq War, 160 of which were officially confirmed by the Department of Defense
American Sniper (book)_RS(0.93)_PS[0.79]
-- American Sniper : The Autobiography of the Most Lethal Sniper in U.S. Military History is an autobiography by United States Navy SEAL Chris Kyle, written with Scott McEwen and Jim DeFelice
American Sniper (book)_RS(0.92)_PS[1.00]
-- With 255 kills, 160 of them officially confirmed by the Pentagon, Kyle is the deadliest marksman in U.S. military history
American Sniper_RS(0.44)_PS[0.21]
-- The film became a
American Sniper (book)_RS(0.42)_PS[0.31]
-- Its film adaptation directed by Clint Eastwood and featuring Bradley Cooper as Kyle was released in 2014

Figure 3: Example of interpretable refuting evidence from Claim-Dissector for claim "*American Sniper (book) is about a sniper with 170 officially confirmed kills.*".

is the corresponding non-negative linear coefficient $C^{i,j}$ max-normalized between 0 and 1 based on maximum $C^{i,j}$ for this sample. The token-level relevance, sentence relevance, and sentence prediction score are highlighted on red-to-black scale (low relevance is black, high-relevance is red). Interestingly, the prediction score is highest for sentences containing crucial refuting evidence — number of confirmed kills.

## O Logit Proof

The link between equation 2 and equation 5 can be easily proved as follows. Applying logarithm to equation 2 we get

$$\log \mathrm{P}^{i,j}(w,y) = \boldsymbol{M}^{i,j}_{w,y} - \log \sum_{w'} \sum_{y'} \exp \boldsymbol{M}^{i,j}_{w',y'}. \tag{21}$$

Expressing $\boldsymbol{M}^{i,j}_{w,y}$, substituting $C^{i,j} = \sum_{w'} \sum_{y'} \exp \boldsymbol{M}^{i,j}_{w',y'}$, and merging the logarithms, we arrive to equation 5. We recommend Bishop (2006), chapter 4.2 for further information.

## P The Necessity of Baseline Modifications

The reason for the modification is that (i) the original model (Schlichtkrull et al., 2021) (without $L_{b1}$) could not benefit from NEI annotations present on FEVER, resulting in unfair comparison with our models and previous work, as TabFact does not contain such annotations (ii) the original model is not able to distinguish the attribution from the repeated relevant evidence, because equations (6)/(9) in their work just sum the probabilities of relevant items supervised independently — they do not use the supervision of overall veracity for the claim. This is problematic especially in FEVER setting comparable to ours, where the relevance of hundreds of sentences is considered (many of them possibly relevant) as compared to TabFact where only top-5 retrieved tables were considered, and often only single is relevant.