CLAIM-DISSECTOR: AN INTERPRETABLE FACT-CHECKING SYSTEM WITH JOINT RE-RANKING AND VERACITY PREDICTION

Martin Fajcik Petr Motlicek Pavel Smrz

Idiap-Com-03-2022

SEPTEMBER 2022

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

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

Martin Fajcik[1, 2], **Petr Motlicek**[1, 2], **Pavel Smrz**[2]

[1]IDIAP Research Institute, Martigny, Switzerland
[2]Brno University of Technology, Brno, Czech Republic
martin.fajcik@vut.cz

Abstract

We present Claim-Dissector: a novel latent variable model for fact-checking and fact-analysis, which given a claim and a set of retrieved provenances allows learning jointly: (i) what are the relevant provenances to this claim (ii) what is the veracity of this claim. We propose to disentangle the per-provenance 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-provenance relevance probabilities. This way, it can be clearly identified the relevance of which sources contributes to what extent towards the final probability. We show that our system achieves state-of-the-art results on FEVER dataset comparable to two-stage systems typically used in traditional fact-checking pipelines, while it often uses significantly less parameters and computation.

Our analysis shows that proposed approach further allows to learn not just which provenances are relevant, but also which provenances lead to supporting and which toward denying the claim, without direct supervision. This not only adds interpretability, but also allows to detect claims with conflicting evidence automatically. Furthermore, we study whether our model can learn fine-grained relevance cues while using coarse-grained supervision. We show that our model can achieve competitive sentence-recall while using only paragraph-level relevance supervision. Finally, traversing towards the finest granularity of relevance, we show that our framework is capable of identifying relevance at the token-level. To do this, we present a new benchmark focusing on token-level interpretability — humans annotate tokens in relevant provenances they considered essential when making their judgement. Then we measure how similar are these annotations to tokens our model is focusing on. Our code, and dataset will be released online.

1 Introduction

Automated fact-checking systems today 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 the 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 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 errorneous 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 provenances[2] within these documents and then predict the claim's veracity from the top-K (usually K=5) relevant provenances.

This comes with several drawbacks; firstly *the multiple steps of the system lead to multi-step error propagation*, i.e. the input to the last system might often be too noisy to contain any information. Some previous work already targeted merging provenance reranking and veracity prediction into 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*, especially when there is a lot of repeated

[1] Discussion in Appendix C.

[2] For instance sentences, paragraphs, or larger text blocks.

evidence. Thirdly, again in open-domain setting, *sometimes there is a 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 may select the evidence from one group based on the in-built biases.

To further strengthen persuasive effect of the evidence, and understand the model's reasoning process, some of these systems are interpretable (Popat et al., 2018; Liu et al., 2020; Krishna et al., 2021). However, to our knowledge, the interpretability in these systems was considered an useful trait, which was evaluated only qualitatively.

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

1. jointly ranks top-relevant, top-supporting and top-refuting provenances and predicts veracity of the claim in an interpretable way, where the probability of the claim's veracity is given by the linear combination of per-provenance probabilities,
2. is able to detect claims with conflicting evidence (both supporting and refuting),
3. can provide unsupervisedly learned fine-grained (sentence-level or token-level evidence), while using only coarse-grained supervision (on block-level or sentence-level respectively),
4. 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 to provide a quantitative evaluation of our system.

2 Model Description

We present a 2-stage system composed from 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 provenance (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.

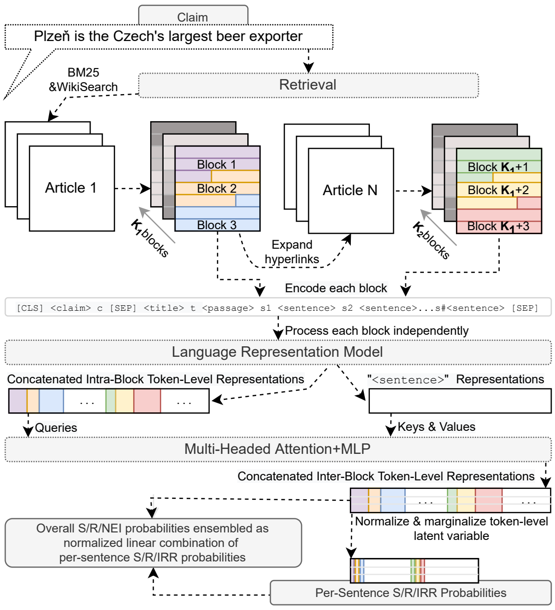
Claim
Plzeň is the Czech's largest beer exporter
BM25
&WikiSearch
Retrieval
Article 1
Block 1
Block 2
Block 3
K_1blocks
Expand
hyperlinks
Article N
Block $\mathbf{K_1}$+1
Block $\mathbf{K_1}$+2
Block $\mathbf{K_1}$+3
K_2blocks
Encode each block
[CLS] <claim> c [SEP] <title> t <passage> s1 <sentence> s2 <sentence>...s#<sentence> [SEP]
Process each block independently
Language Representation Model
Concatenated Intra-Block Token-Level Representations
"<sentence>" Representations
Queries
Keys & Values
Multi-Headed Attention+MLP
Concatenated Inter-Block Token-Level Representations
Normalize & marginalize token-level latent variable
Overall S/R/NEI probabilities ensembled as normalized linear combination of per-sentence S/R/IRR probabilities
Per-Sentence S/R/IRR Probabilities

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.

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 ranking function $\mathrm{rank} : \mathcal{C} \times \mathcal{D} \rightarrow \mathbb{R}$ that assigns a score to each document in the corpus. In this work, we focus on the verifier; therefore we take strong retriever from Jiang et al. (2021). This retriever interleaves documents ranked via BM25 (Robertson and Zaragoza, 2009) and Wikipedia API following Hanselowski et al. (2018), skipping duplicate articles. Each document is then split into non-overlapping blocks of size L_x, while respecting sentence boundaries[3]. Our verifier then computes its 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 the 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

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

is conditioned on other relevant documents. However, we stress that such approaches also mimic the way FEVER dataset was collected, and thus the improvements of such approach on "more naturally collected" datasets might not be as significant, if any.

2.2 Verifier

The verifier firstly processes each block independently by language representation model (LRM) and then aggregates cross-block information via multi-headed attention (Vaswani et al., 2017), computing matrix $\boldsymbol{M}$. This matrix is used to compute both, the probability of each provenance'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's 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 $t, s_1, s_2, ..., s_\#$ 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 a `<sentence>` token. Their respective embeddings are trained from scratch. Each input x_i is then encoded via LRM $\boldsymbol{E}_i = \text{LRM}(x_i) \in \mathbb{R}^{L_B \times d}$, where L_B is an input sequence length, and d is LRM's hidden dimensionality. Representations for 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 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 number of all tokens in all input sentences (without special tokens). Similarly, we index-select all representations at the same positions as the `<sentence>` tokens at then 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

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

The MHAtt operator is a multi-headed attention with queries $\boldsymbol{E}_s$, and keys and values $\boldsymbol{S}$. The SLP operator is a single layer perceptron described closely in Appendix A and $\boldsymbol{W} \in \mathbb{R}^{d \times 3}$ is a linear transformation, projecting resulting vectors to desired number of classes (3 in case of FEVER).

To compute the per-provenance probabilities we split the matrix $\boldsymbol{M}$ according to tokens belonging into each provenance. For instance for sentence-level provenance 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 normalize each such matrix of i-th provenance of j-th block as[4]:

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

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

$$\log \text{P}^{i,j}(\boldsymbol{y}) = \log \sum_{\boldsymbol{w}'} \text{P}^{i,j}(\boldsymbol{y}, \boldsymbol{w}') \qquad (3)$$

Then loss $\mathcal{L}_R$ is computed from the probabilities of annotated label set $\mathbb{A}$ for a single claim[5].

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

In training, we use maximum likelihood to maximize the log-probability $\log \text{P}(\boldsymbol{y} = \boldsymbol{y}^*); \boldsymbol{y}^* \in \{S, R\}$ based on the annotated label of the claim's veracity, and maximize $\boldsymbol{y}$ = IRR for irrelevant provenances. 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 minimizing loss for false negatives. In test-time, we rank the provenance (i, j) according to its combined probability of supporting or refuting relevance $score_{i,j} = \sum_{\boldsymbol{y} \in \{S,R\}} \text{P}^{i,j}(\boldsymbol{y})$.

[4] Note that the distribution 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 probability of the claim's veracity $\boldsymbol{y} \in \{\text{S, R, NEI}\}$. First notice that scores in $\boldsymbol{M}$ are logits

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

Therefore, we use an extra degree of freedom $K_{i,j}$ to compute a linear ensemble[6] producing the final probability

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

Lastly, we bias model to focus only on some tokens in each provenance by enforcing an L2 penalty over the scores in $\boldsymbol{M}$ by

$$\mathcal{L}_2 = \frac{1}{3L_e} ||\boldsymbol{M}||^2_F, \tag{7}$$

where $|| \cdot ||_F$ denotes Frobenius norm. We show empirically that this loss leads to significantly better unsupervised token-level interpretability (section 5). Therefore the final per-sample loss to maximize with hyperparameters λ_R, λ_2 is

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

2.3 Baseline

Apart from previous work, we propose a baseline bridging the proposed system and the recent work of Schlichtkrull et al. (2021). We normalize all scores in $\boldsymbol{M}$ to compute joint probability across all blocks

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

First, we marginalize out per-token probabilities in each provenance $\boldsymbol{s}_{i,j}$.

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

Using this sentence probability formulation, the loss is computed for every relevant provenance.

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

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

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

by computing their mean. Like CD, we use $\mathcal{L}_{b1}$ loss to take advantage of examples from NEI class for which we have no annotation $\mathbb{A}_p$ (and thus $\mathcal{L}_{b0}$ is virtually set to 0). Unlike CD, the annotations $\mathbb{A}_p$ in $\mathcal{L}_{b0}$ contain only sentences where $\boldsymbol{y} \in \{S, R\}$[7].

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

2.4 Transferring Supervision to Lower Language Granularity

The proposed model can benefit from annotation on different granularity of the language. For example, the provenance annotation can be done on document, block, paragraph or token-level. In section 5, we show despite the fact that model is trained on higher granularity level, the model still shows moderate performance of relevance prediction when evaluated on lower-level 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 Gaussian prior over per-token probabilities enforced by the loss $\mathcal{L}_2$ is crucial (see Table 5).

Secondly, *we assume only block-level annotation is available and we evaluate on sentence-level annotation.* Here we slightly alter the model, making it to rely more on its sentence-level representations. In section 5, we show this simple alteration significantly improves the performance at sentence-level. Note that to compute block-level probability, the block is the provenance, therefore the provenance index can be dropped. The probability of the j-th block $\boldsymbol{b}_j$ is obtained by marginalizing out the per-token (respectively per-sentence) probabilities.

[6] Assuming $\boldsymbol{y}$=IRR=NEI.

[7] We observed that 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 loss tries to allocate mass of joint space in NEI class, since most documents are irrelevant, whereas the latter loss tries to allocate the mass in the dimension of labelled veracity class.

$$\mathrm{P}(\boldsymbol{b}_j, \boldsymbol{y}) = \sum_{\boldsymbol{s}_{i,j} \in \boldsymbol{b}_j} \mathrm{P}(\boldsymbol{s}_{i,j}, \boldsymbol{y}) = \sum_{\boldsymbol{s}_{i,j} \in \boldsymbol{b}_j} \sum_{\boldsymbol{w}' \in \boldsymbol{s}_{i,j}} \mathrm{P}(\boldsymbol{w}', \boldsymbol{y}) \tag{13}$$

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

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

Intuitively, making a single-sentence estimate (SSE) forces model to rely more on its sentence-level probabilities during training. In $\mathcal{L}_R$ we then maximize the probabilities of positives blocks computed as in equation 14, and negative sentences[8] 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 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 D.

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. In future revisions, we plan to also validate our approach on HoVer.

Open-Domain Fact-Checking (ODFC) Unlike this work, most of the previous work includes 3-stage systems which 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 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 judgement of relevant evidence is crucial when computing final veracity. We stress that our system falls into joint judgement category. Although the relevance is computed per-sentence, these probabilities along with linear combination coefficients are computed jointly, with 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 block, and multi-headed attention layer propagates cross-block information. Our results (section 5) suggest our system is about on-par with these iterative approaches, while requiring only single forward computation.

Interpretability Popat et al. (2018); Liu et al. (2020) both introduced systems with an interpretable attention design, and they demonstrated its ability to highlight important words via 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 a data from dataset VitaminC. The dataset, constructed from factual revisions on Wikipedia, assumed that the revised part of facts is the most salient part of evidence. In contrast, we instruct annotators to manually annotate terms important to their judgement. The VitaminC dataset is not accompanied with the evidence corpus, thus we deemed it as unsuitable for open-domain knowledge processing.

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

Krishna et al. (2021) proposed a system which parses evidence sentences into a natural logic-based inferences (Angeli and Manning, 2014). These provide deterministic proof of 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 significant bias — 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 outcome, typically supported by single table. While our work started ahead of its publication, it can be seen as an extension of this system.

4 Experimental Setup

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 $\mathcal{L}_R$ loss 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.

	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%)

Table 1: FEVER dataset and its subsets.

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 he considered relevant). To analyze performance of our components on samples that need multi-hop reasoning, we further create subsets of training/development set:

- FEVER$_{MH}$ contains only examples where all annotators agreed on that more than 1 sentence is required for verification.
- FEVER$_{MH_{ART}}$ contains only examples, where all annotators agreed that multiple sentences from different articles are required for verification.

We include the subset statistics in Table 1.

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 F.

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 3 annotations, while the system is compared to 4 annotations (thus the performance of average annotator on 4 annotations would be equal or better). We measure performance via F_1 metric. We will expand the dataset size in future revisions.

4.2 Evaluation

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.

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 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[9]. When evaluating F_1, we select only tokens with scores greater than threshold τ. We tune the optimal threshold τ w.r.t. F_1 on TLR-FEVER.

5 Results

We now present our results. We report results of base-sized models based on 3-checkpoint average. We train only a single large model.

Retrieval. We evaluate the retrieval method from Jiang et al. (2021) and the proposed hyperlink expansion method in Table 2. 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 provide additional evaluation of the retrieval method with varying K_1 in Appendix E. 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+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 2: Retrieval performance in RaI on FEVER dev set and its subsets. #SaI denotes average number of sentences at model's input under corresponding $K_1 + K_2$ setting.

	System	**FS**	**A**	**R@5**	**#θ**
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 (Krishna et al., 2021)	**78.2**	80.2	-	515M
	Baseline$_{joint}$	75.2	79.8	90.0	187M
	Claim-Dissector$_{RoBERTa}$	74.6	78.6	90.4	127M
	Claim-Dissector	76.2	79.5	91.5	187M
	Claim-Dissector \w HE	76.9	79.8	93.0	187M
	Claim-Dissector$_{LARGE}$ \w HE	78.0	80.8	93.3	439M
	Claim-Dissector$_{LARGE}$ \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 (Krishna et al., 2021)	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
	Claim-Dissector$_{LARGE}$ \w HE	76.5	79.3	-	439M

Table 3: Performance on development and test splits of FEVER. #θ denotes number of parameters in the model. Model names suffixed with [OR](as Oracle) inject missing golden evidence into its input. Model version with hyperlink expansion is suffixed as (\w HE).

Main Results. We compare the performance of our system with previous work in Table 3. 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

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

	FEVER			FEVER$_{MH}$		
System	**FS**	**A**	**R@5**	**FS**	**A**	**R@5**
CD$_{LARGE}$ \w HE [OR]	78.9	81.2	94.7	47.1	76.8	57.4
CD$_{LARGE}$ \w HE	78.0	80.8	93.3	41.9	75.7	51.5
CD \w HE	76.9	79.8	93.0	38.1	74.1	48.3
CD \w HE \wo MH	76.5	79.5	92.7	38.4	75.5	47.7
Baseline	75.2	79.8	90.0	26.1	74.2	33.9
CD	76.2	79.5	91.5	26.3	68.9	35.5
CD \wo $\mathcal{L}_2$	76.0	79.6	91.5	28.8	74.7	35.9
CD \wo VC	-	-	91.9	-	-	37.2
CD \wo RC	-	79.9	-	-	76.8	-

Table 4: Ablation Study.

System	F1
Select All Tokens	52
Select Claim Overlaps	63
Masker	73
Claim-Dissector \wo $\mathcal{L}_2$	61
Claim-Dissector	75
Human Performance LB	85

Table 5: Token-level relevance.

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 6: Sentence-level performance under different kinds of supervision.

paired with original FEVER claims (Thorne and Vlachos, 2021).

We observe that (i) even our base-sized RoBERTa model outperforms HESM on dev data, which in turn outperformed KGAT and DREAM in FS on test data, (ii) our base sized DebertaV3-based model Claim-Dissector outperforms DREAM (Zhong et al., 2020) and even KGAT with oracle inputs (Liu et al., 2020), (iii) model version with hyperlink expansion (suffixed \w HE) significantly improves the overall performance, (iv) increasing size of the model to *LARGE* improves mostly its accuracy, (v) Claim-Dissector$_{LARGE}$ \w HE achieves better FEVER score than T5-based approach (Jiang et al., 2021) (with two 3B models) and better accuracy than LongFormer+DebertaXL (Stammbach, 2021) and ProofVer (Krishna et al., 2021), but it is not pareto optimal to these previous SOTA. 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 golden evidence into inputs of Claim-Dissector$_{LARGE}$ \w HE at random positions and measure its performance (suffixed [OR]). We observe that items missed by retrieval are not just noisy examples and are still beneficial to the system performance.

Ablations. We ablate components of the Claim-Dissector (CD) in Table 4. Firstly, we resort to single-task training. We drop veracity classification (VC) loss or relevance classification (RC) loss from the training. We observe an overall trend — single-task model performs better to multi-task model.

Next we analyze the effect of dropping the $\mathcal{L}_2$ loss from the total objective. We observe no significant difference on FEVER, but we observe large drop in accuracy on FEVER$_{MH}$. The experiments show large variance (std ±2)[10]. We hypothesize that while $\mathcal{L}_2$ loss doesn't seem cause any damage to performance on average, the model could be focusing on different dataset cues in each training run. We seek to further investigate the phenomena in future revisions of this work.

Further, we study the effect of hyperlink expansion (HE) and the effect of multi-headed (MH) attention layer. As expected, hyperlink expansion dramatically increases performance on FEVER$_{MH}$. The multi-headed attention also brings marginal improvements to results on FEVER. However, contrary to our expectations, the effect of MH layer on FEVER$_{MH}$ is not significant.

Transferring sentence-level supervision to token-level performance. We evaluate the performance of token-level rationales[11] on our dataset in Table 5. We considered two trivial baselines. First was to select all tokens in golden evidences (Select All Tokens). Second was to select only tokens which overlap with claim tokens (Select Claim Overlaps). As we have not done early stopping for token-level relevance (but with FEVER score), we report best out of 3 runs result for Claim-Dissector. We found that our model with unsupervised objective produces rationales equal or better than the masker — a separate model trained explicitly to identify tokens important to model's prediction. Furthermore, the results demonstrate the importance of $\mathcal{L}_2$ objective. However the human performance lower-bound is still far beyond the performance of our unsupervised approach.

[10] Another set of 3 checkpoints had A 73.4 on FEVER$_{MH}$.

[11] We visualized token-level rationales on 100 random dev set examples at shorturl.at/beTY2.

Transferring block-level supervision to sentence-level performance. The performance of our model on the sentence-level provenances is evaluated in Table 6. We notice that even our vanilla Claim-Dissector trained with block supervision reaches competitive recall@5 on sentence-level. However, adding SSE leads to further improvements both in recall, but also in accuracy. We hypothesized that the recall will be improved, because model with now focus on assigning high probability mass only to some sentences within block, since high-entropy of the per-sentence distribution would lead to poor loss performance. However, we have not foreseen the damaging effect on accuracy, which block-level supervision causes. Interestingly, the accuracy without any provenance supervision from Table 4 was increased.

Detection of examples with conflicting evidence. Finally, we manually analyze whether we can take advantage of model's ability to distinguish between provenance, which is relevant because it supports the claim, and the provenance 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 conflicting evidence). We note that there were no examples with explicitly annotated conflicting evidence in the training data.

We select all examples where model predicted at least 0.9 probability for any supporting and any refuting provenance. Formally we select every example for which the following condition holds: $\exists a, b, x, y : \mathrm{P}^{a,b}(\boldsymbol{y} = S) > 0.9 \wedge \mathrm{P}^{x,y}(\boldsymbol{y} = R) > 0.9$. We found that out of 72 examples, $66\%(48)$ we judged as indeed having a conflicting evidence[12]. We observed that about half (25/48) of these examples had conflicting evidence because of the entity ambiguity caused by open-domain setting. For instance 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).

6 Known Problems and Limitations

By manual analysis, we found that claim-dissector suffers from overconfidence in blocks with at least 1 relevant provenance. Then it seeks to select more relevant provenances 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 provenances.

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 trustworthiness, or its narrative. This is the area of active research, as human fact-checkers also need to deal with lies (Uscinski and Butler, 2013).

7 Conclusion & Future Work

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 provenances into ranking of supporting evidence and ranking of refuting evidence. This allows detecting conflicting evidence without being exposed to such conflicting evidence sets during training. Thirdly, it combines the per-token relevance probabilities via linear combination into final veracity assessment. This allows to understand, 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.

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 used is transferable beyond fact-checking.

In future revisions, we seek to extend our results to another dataset (possibly HoVer), increase sample size of `TLR-FEVER`, and verify marginal differences in our results with statistical tests.

[12] Annotations are available at shorturl.at/qrtIP.

Acknowledgements

This work was supported by CRiTERIA, an EU project, funded under the Horizon 2020 programme, grant agreement no. 101021866.

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.

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

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.

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.

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*.

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.

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. 2021. Proofver: Natural logic theorem proving for fact verification. *arXiv preprint arXiv:2108.11357*.

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.

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 Structure of Single-layer Perceptron

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

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

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

B 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 loss functions.

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

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.

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

	FEVER			FEVER$_{MH}$		
Model	**FS**	**A**	**R@5**	**FS**	**A**	**R@5**
CD	76.2	79.5	91.5	26.3	68.9	35.5
Baseline	75.2	79.8	90.0	26.1	74.2	33.9
L_{b3}	76.0	79.0	91.2	20.2	71.8	26.3
L_{b4}	75.7	79.7	90.4	23.4	72.3	31.4

Table 7: Different model parametrizations.

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

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

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

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.

C 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.*

D Masker

Model Description. Our masker follows same DeBERTaV3 architecture as Claim-Dissector. 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.

$$e_i' = m_0^i e_i + m_1^i e_m \qquad (19)$$

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 loss to maximize with sparsity strength hyperparameter λ_S is given as

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

Training Details. We keep most hyperparameters the same as for Claim-Dissector. The only difference is learning rate $2e{-}6$, and an adaptive scheduling on Gumbel-softmax temperature τ. 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.

E Retrieval Performance

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 8: Retrieval performance.

An in-depth evaluation of retrieval method adopted from Jiang et al. (2021) is available in Table 8.

F 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!