Evidence Decomposition Graph Network for Fact Verification

Anonymous ACL submission

Abstract

Fact verification is the task to verify a given claim according to extracted evidence sentences. Most existing works use whole evidence sentences or break them into phrases to perform evidence interaction, where evidence is treated either too coarsely or over fragmented. We also find that many models suffer from exposure bias, which finally leads to them only paying attention to the evidence ranked higher by previous steps while failing to recognize crucial pieces from all candidates. In this paper, we propose an Evidence Decomposition Graph Network (EDGN), which decomposes each evidence sentence, especially the complex ones, into several simple sentences, highlighting the required key information without losing sentence structure and meaning. EDGN also absorbs a simple but effective evidence shuffling method to mitigate exposure bias. Experiments on the FEVER benchmark show our model can take all evidence candidates into account, distill necessary key information from complex evidence, and outperform existing methods in the literature. We will release our code to the community for further exploration.

1 Introduction

FEVER (Thorne et al., 2018) is a Fact Extraction and Verification task, where a system is asked to predict whether a given claim is supported, refuted, or can not be verified based on a Wikipedia dump. Most existing works follow a three-step pipeline (Thorne et al., 2018): (i) Retrieve relevant pages from Wikipedia dump. (ii) Extract evidence sentences from retrieved pages. (iii) Verify the given claim based on the extracted evidence candidates. Hanselowski et al. (2018) and Liu et al. (2020) have contributed efficient and effective methods for the retrieval and extraction steps respectively. Table 1 shows an example with extracted evidence candidates present in a descending order according to their ranking scores by Liu et al. (2020).

Many subsequent works simply use the evidence candidates they provided and focus on the claim verification step. They usually perform evidence interaction at sentence or phrase level, which is either too coarse or over-fragmented. The length of sentences in Wikipedia varies a lot, so as the extracted evidence sentences. If we represent these evidence pieces with sentence vectors of a fixed size, some key information in complex sentences may be lost or overwhelmed. On the other hand, simply breaking evidence sentences into phrases and organizing as graphs may also introduce noise information, which may even make them contradictory to the original evidence. Previous work (Portelli et al., 2020) finds that the key information in an evidence sentence is generally in continuous spans, indicating a coarse-grained decomposition method could refine the evidence effectively. Therefore, we propose to decompose and re-organize an original evidence sentence into sub-evidence pieces, namely, several simple sentences. As shown in Table 1, the gold evidence sentence is decomposed into two sub-evidence pieces, ***[Love the Way You Lie]*** *"Love the Way You Lie" is a song recorded by the American rapper Eminem.* and ***[Love the Way You Lie]*** *featuring the Barbadian singer Rihanna, from Eminem's seventh studio album Recovery (2010).* Compared to models operating at sentence-level or phrase-level, our method can maintain a relatively complete but focused meaning regarding the original evidence sentence.

Meanwhile, previous works simply concatenate the claim and extracted evidence pieces as the input sequence fed to a classifier, while paying little attention to the order of those evidence candidates in the sequence. They often put sentences with higher evidence extraction scores closer to the given claim. Some works (Liu et al., 2020) even purposely insert gold evidence sentences next to

Claim: Recovery features Rihanna on the track Love the Way You Lie.
Label: SUPPORT
Evidence Candidates: **[Love the Way You Lie/0]** "Love the Way You Lie" is a song recorded by the American rapper Eminem, featuring the Barbadian singer Rihanna, from Eminem's seventh studio album Recovery (2010). **[Recovery (Eminem album)/0]** It spawned four singles; "Not Afraid ", "Love the Way You Lie", "No Love", and "Space Bound", with the former two both reaching number one on the Billboard Hot 100. **[Love the Way You Lie/1]** Interscope Records released the song in August 2010 as the second single from Recovery .
Gold Evidence: [Love the Way You Lie/0]
Sub-evidence Pieces of the gold Evidence: **[Love the Way You Lie]** "Love the Way You Lie" is a song recorded by the American rapper Eminem . **[Love the Way You Lie]** featuring the Barbadian singer Rihanna , from Eminem 's seventh studio album Recovery (2010) .

Table 1: An instance with the decomposed gold evidence of FEVER dataset.

the claim, in the hope of better training the verification model. However, this may make the verification model learn shortcuts, e.g., only concentrating on evidence pieces close to the claim in the input sequences, but probably failing to recognize the gold evidence from imperfect evidence extractions. This exposure bias will limit the model's generalization ability at real-world scenarios, where the positions of the gold evidence are relatively scattered. We think that one way to prevent learning such shortcuts may be to shuffle the candidates during training, and push the model to learn to recognize crucial evidence from tough cases, thus improve its generalization ability.

In this paper, we propose an Evidence Decomposition Graph Network (EDGN), a sub-sentence level graph network with the evidence decomposition and reordering mechanism. Specifically, EDGN decomposes each evidence candidate into several clause-level sub-evidence pieces, highlighting key information in evidence candidates without losing much syntactic and semantic information in the original sentences. We also connect sub-evidence pieces from the same evidence to form an sub-evidence interaction graph. With a random shuffle mechanism, we reorder the evidence candidates to help EDGN learn to recognize crucial evidence. Then, we apply a multi-layer Graph Attention Network (GAT) to the sub-evidence interaction graph, which accumulates context information for each sub-evidence piece. With the claim and all contextualized sub-evidence pieces, EDGN predicts the veracity label for the given claim.

Experiments on FEVER show that our EDGN outperforms other published papers in both base and large PLM settings. The evidence shuffling method is also proved to be effective to cope with the exposure bias and improve the robustness of the fact verification model. Even equipped with a base version pretrained language models, our EDGN still achieves comparable results to recent works that based on RoBERTa-large.

Our main contributions can be summarized as:

• we propose a novel evidence decomposition and re-organization strategy, which can maintain a focused but complete meaning regarding original evidence, benefiting fact verification performance on FEVER benchmark.

• we propose a simple but effective evidence reordering mechanism to avoid the exposure bias introduced by the overfitting and gold evidence insertion in the evidence extraction step of FEVER pipeline, which many previous works suffer from.

2 Methodology

In this paper, we focus on the last step of FEVER pipeline, namely claim verification as many recent works do.

The first two steps obtain a set of evidence sentences extracted from the Wikipedia dump. As a result, for each instance of the claim verification step, the input consists of a claim sentence C and an evidence set $E = \{(t_j, e_j)\}_{j=1}^{k}$, which contains k evidence candidates $\{e_j\}$ with corresponding Wikipedia titles $\{t_j\}$.

2.1 Overview

Figure 1 shows an overview of our model. First, each evidence sentence is split and re-organized into a collection of sub-evidence (§ 2.2). In this work, a ***sub-evidence*** piece is a simple sentence that represents a specific perspective or an assertion of the original evidence sentence. It is typically more fine-grained than the original sentence.

We encode the sub-evidence via pre-trained language models (§ 2.3). Then, we construct a sub-evidence interaction graph to obtain the contextualised representations of each sub-evidence (§ 2.4). Finally, we predict the veracity label (§ 2.5) with an auxiliary task of gold evidence prediction (§ 2.6).

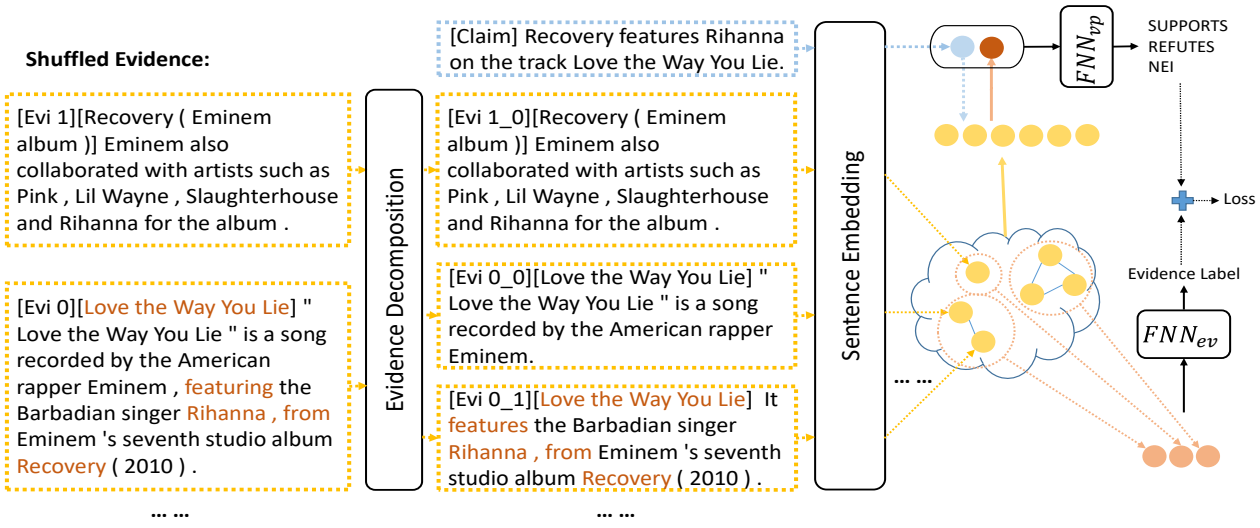

Figure 1: Architecture of our approach. Each yellow node represents a sub-evidence piece, and each orange node an evidence sentence. The blue node is the claim.

2.2 Evidence Decomposition

We split and rephrase each evidence sentence e_j into several sub-evidence pieces to make the evidence more concise while keeping the similar granularity to the claims. We use an off-the-shelf sentence splitting and rephrasing tool called LaserTagger(Malmi et al., 2019), which is trained on the WikiSplit dataset(Botha et al., 2018). LaserTagger treats the sentence split as a text editing task, and uses a BERT encoder with an auto-regressive sequence labeling decoder.

We find that LaserTagger is prone to splitting out long entities, e.g., *the NAACP Image Award for Outstanding Supporting Actor* will be treated as a sub-sentence. However, in our circumstance, these entities can not be proper sub-evidence since they do not make clear assertions. Therefore, we identify entities longer than two words with Spacy[1], and replace these entities with anonymous tokens, *entx*, where x is the entity id. After evidence splitting, these anonymous tokens are recovered.

2.3 Evidence Encoding

We further concatenate each sub-evidence sentence with its Wikipedia title to build a topic background. This also potentially resolves the co-reference issues. The concatenated sequence are fed into a pre-trained language model (PLM) to obtain the evidence representations. The input of the PLM can be formulated as: $\langle s \rangle$ claim $\langle /s \rangle$ $\langle /s \rangle$ sub_e$_1$ $\langle /s \rangle$ $\langle /s \rangle$ sub_e$_2$ $\langle /s \rangle \cdots$, where claim represents the tokens in the given claim, and sub_e$_i$ is the concatenation of the title and content of the i_{th} sub-evidence. The representation of the i_{th} sub-evidence is the average of the $\langle /s \rangle$'s representations at the start and end of the sub-evidence. The claim-aware evidence encoding results are formulated as:

$$\hat{E} = \{\hat{\boldsymbol{e}}_{jr} \in \mathbb{R}^{h_p}\}_{j=1,r=1}^{k,m_j} \quad (1)$$

, where an original evidence sentence e_j is split into m_j sub-evidence fragments, h_p is the hidden size of the PLM.

evidence shuffling Before each epoch starts, we randomly shuffle the original evidence sentences' order of all training and validation examples. By doing so, we make the location distribution of gold candidates in all set become more similar, which will prevent exposure bias introduced by the evidence extraction step and ensure the verification model learn to considerate all evidence candidates when making predictions.

2.4 Sub-evidence Interaction

We construct an undirected graph G to facilitate interactions among sub-evidence pieces from the same original evidence sentence. Each node in G denotes a sub-evidence, and is initialized with its encoding, $\hat{e}_{jr} \in \hat{E}$ from Eq. 1. There is an edge between the node pair $\hat{e}_{j_1 r_1}$ and $\hat{e}_{j_2 r_2}$ only if $j_1 = j_2$. We also add a self-loop edge for every node in the graph.

Then a two-layer Graph Attention Network (GAT) is applied to this graph to facilitate the

[1] https://spacy.io

sub-evidence nodes collecting context information from the neighborhood. We follow (Velickovic et al., 2018) and implement a two-layer GAT with multi-head attention. The contextualized node representations of the sub-evidences after GATs can be formulated as:

$$\left\{\hat{\boldsymbol{q}}_{jr}^{E}\right\} = \mathrm{GAT}(\{\hat{e}_{jr}\}, G) \in \mathbb{R}^{(\sum m_j)\times h} \quad (2)$$

, where $\hat{\boldsymbol{q}}_{jr}^{E}$ is the representation of the r_{th} sub-evidence of the j_{th} evidence candidate.

2.5 Veracity Prediction

We also get the claim encoding from the PLM's outputs described in § 2.3. Specifically, note the encoding output of PLM for a claim with length of m^c as $\{\boldsymbol{c}_i\}_{i=1}^{m^c} \in \mathbb{R}^{m^c \times h_t}$, the claim representation $\boldsymbol{q}^C$ satisfies:

$$\alpha_i^c = \mathrm{softmax}\left(\mathrm{FNN}_c\left(\boldsymbol{c}_i\right)\right)$$

$$\boldsymbol{q}^C = \sum_{i=1}^{m^c} \alpha_i^c \cdot \boldsymbol{c}_i$$

FNN_c is a two-layer feedforward neural network.

The claim representation is then used to perform a cross attention with sub-evidence representations to obtain the refined evidence vector $\boldsymbol{q}^E$:

$$\alpha_{jr} = \mathrm{softmax}\left(\mathrm{FNN}_{\mathrm{ce}}\left([\boldsymbol{q}^C; \hat{\boldsymbol{q}}_{jr}^{E}]\right)\right)$$

$$\boldsymbol{q}^E = \sum_{j=1}^{k}\sum_{r=1}^{m_j} \alpha_{jr} \cdot \hat{\boldsymbol{q}}_{jr}^{E}$$

$\mathrm{FNN}_{\mathrm{ce}}$ is a two-layer feedforward neural network, and $[\boldsymbol{x}; \boldsymbol{y}]$ means the concatenation of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

Finally, we obtain the predicted veracity probability distribution with FNN_{vp} and a softmax layer based on the concatenation of the claim embedding $\boldsymbol{q}^C$ and the refined evidence vector $\boldsymbol{q}^E$:

$$\mathrm{p}\left(\hat{y}|C, E\right) = \mathrm{softmax}\left(\mathrm{FNN}_{\mathrm{vp}}\left([\boldsymbol{q}^C; \boldsymbol{q}^E]\right)\right)$$

, where $\mathrm{p}\left(\hat{y}|C, E\right)$ is the probability of the predicted label $\hat{y}$ given the claim C and evidence candidates E.

We use maximum likelihood estimation to optimize our veracity prediction model. The negative log-likelihood loss L_{vp} satisfies:

$$L_{\mathrm{vp}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\mathrm{p}\left(\hat{y} = y_i|C, E\right)\right)$$

y_i is the true label of the i_{th} instance. N is the number of instances for training.

2.6 Auxiliary Gold Evidence Prediction

We use gold evidence prediction as an auxiliary task to guide the model to identify and focus on more helpful evidence candidates. Specifically, we apply an average pooling over the sub-evidence representations $\{\hat{\boldsymbol{q}}_{jr}^{E}\}$ (from Eq. 2) to obtain representations of k evidence sentences $\{\tilde{\boldsymbol{q}}_{j}^{E}\}_{j=1}^{k}$.

$$\tilde{\boldsymbol{q}}_{j}^{E} = \frac{1}{m_j}\sum_{r=1}^{m_j} \hat{\boldsymbol{q}}_{jr}^{E}$$

The evidence verification prediction is obtained via a two-layer feed-forward neural network according to the evidence sentence representation:

$$\mathrm{p}\left(\hat{y}_j^E|C, E\right) = \mathrm{softmax}\left(\mathrm{FNN}_{\mathrm{ev}}\left(\tilde{\boldsymbol{q}}_{j}^{E}\right)\right)$$

The evidence verification is optimised by maximum likelihood estimation, the negative log likelihood L_{ev} is:

$$L_{\mathrm{ev}} = -\frac{1}{N}\frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{k} \log\left(\mathrm{p}\left(\hat{y}_{ij}^E = y_{ij}^E|C, E\right)\right)$$

y_{ij}^E is the label of the j_{th} evidence in the i_{th} training instance. Particularly, for those instances with veracity labels SUPPORTS and REFUTES, we label the evidence in their annotated gold evidence set as RELEVANT. The rest evidence candidates, including all evidence of the instances with veracity label of NOT ENOUGH INFO, are labeled as IRRELEVANT.

Finally, the loss function is the combination of the veracity prediction loss and the evidence verification loss: $L = L_{\mathrm{vp}} + \lambda \cdot L_{\mathrm{ev}}$, where λ is the scaling weight, which is set to 0.5 in our model.

3 Implementation Details

For the first two steps of FEVER, we use the method of Hanselowski et al. (2018) to retrieve documents and the evidence extraction approach proposed by Liu et al. (2020). Detailed statistics of FEVER and the evidence extraction results are shown in Appendix A. We keep the top 5 evidence candidates as input to our model. These settings are consistent with most existing work, ensuring a fair comparison with the baseline.

We obtain 9 sub-evidence pieces from each instance on average. Averagely, each sub-evidence piece contains 16 words, which is much closer to the number of words contained by a claim (8 words) than that of the original evidence (28 words).

For the hyper-parameters, the batch size is set to be 8, with a gradient accumulation step of 8. We use Adam as our optimizer, and train the model for 3 epochs in total. A linear scheduler is applied and the warm-up rate is 20%. The pre-trained language model is initialized with RoBERTa-large. The peak learning rate for parameters in RoBERTa is 2e-5, and 2e-3 for other parameters introduced by our model.

4 Experiments

We compare our model with the following competitive works: (1) **BERT Concat/Pair**[2] are both vanilla PLM models with the Bert-base checkpoint. BERT Concat concatenates the claim and all evidence as the input sequence, while Bert Pair's input is pairs of the claim and each evidence piece. (2) **GEAR** (Zhou et al., 2019) constructs fully-connected sentence-level graphs to perform evidence interaction. (2) **DOMLIN** (Stammbach and Neumann, 2019) adopts a two-staged sentence selection strategy to enhance the evidence extraction step and use Bert Concat for claim verification[3]. (3) **KGAT** (Liu et al., 2020) introduces node kernels and edge kernels to conduct fine-grained evidence propagation on sentence-level graph. (4) **DREAM** (Zhong et al., 2020) constructs phrase-level graphs with an SRL parser for fine-grained interactions. (5) **TARSA** (Si et al., 2021) proposes to perform topic-aware evidence reasoning and stance-aware evidence aggregation for fact verification. (6) **HESM** (Subramanian and Lee, 2020) proposes that the claim and evidence set should be encoded and attended to at various levels of hierarchy. (7) **CorefRoberta/CorefBERT** (Ye et al., 2020) add a mention reference prediction task in pre-training to enhance PLM's ability to capture coreferential information. (8) **MLA** (Kruengkrai et al., 2021) combines token-level and sentence-level self-attention on the evidence candidates. We report its result based-on the same evidence candidates as ours.

4.1 Metrics

There are two metrics to evaluate model performance on the FEVER dataset, i.e., label accuracy and FEVER score (Thorne et al., 2018). Label accuracy only considers the accuracy of veracity labels, while FEVER score involves the evaluation of the extracted evidence.

[2]The results of base version Bert Concat and Bert Pair are directly taken from Zhou et al. (2019). The large version Bert Pair's results are taken from Soleimani et al. (2020).

[3]We do not compare with DOMLIN++ (Stammbach and Ash, 2020), since it introduces external dataset MultiNLI (Williams et al., 2018) and ensemble tricks.

Models	Validation		Test	
	ACC.	F.S.	ACC.	F.S.
Base Size Settings				
BERT Pair	73.30	68.90	69.75	65.18
BERT Concat	73.67	68.89	71.01	65.64
GEAR	74.84	70.69	71.60	67.10
DOMLIN	72.10	–	71.50	68.46
MLA	77.54	74.41	–	–
KGAT	78.02	75.88	72.81	69.40
CorefBERT	–	–	72.88	69.82
HESM	–	–	73.18	70.07
MLA	77.54	74.41	–	–
Our Model	**80.71**	**80.51**	**75.17**	**71.45**
Large Size Settings				
BERT Pair	74.59	72.42	71.86	69.66
KGAT	78.29	76.11	74.07	70.38
DREAM	79.16	–	76.85	70.60
TARSA	81.24	77.96	73.97	70.70
HESM	75.77	73.44	74.64	71.48
CorefRoberta	–	–	75.96	72.30
MLA	–	–	76.30	72.83
Our Model	**82.67**	**82.46**	**76.97**	**73.45**

Table 2: Model performance on FEVER. F.S. is FEVER score.

4.2 Experimental Results

Table 2 shows that, on both base-size and large-size PLM settings, our model outperforms all published results on validation and test set at both metrics. On base-size settings, EDGN outperforms the best baseline model by 1.99% and 1.38% on accuracy and FEVER scores, respectively. On large settings, the improvements over the best baseline are 0.67% and 0.62%. These results suggest the effectiveness of our proposed evidence decomposition and shuffling for the FEVER task.

Specifically, EDGN outperforms GEAR, KGAT, and DREAM, which adopt interactions in sentence or phrase level. This result demonstrates that sub-evidence is a potentially more effective granularity to process and aggregate information from evidence. Processing in sub-evidence level, EDGN is more efficient than MLA, which adopt token-level interactions. Furthermore, our base-size model even performs comparably with some recent large-setting efforts, like HESM (Subramanian and Lee, 2020) and CorefRoberta (Ye et al., 2020). We think the reason is our model makes the key information in longer evidence sentences more prominent, which allows light-weight models to obtain remark-

Models	Validation	
	ACC.	F.S.
Our Model	82.67	82.46
w/ Dep-based Decomposition	82.39	82.18
w/o Evidence Decomposition	81.88	81.68
w/o Graph Interaction	81.86	81.65
w/o Auxiliary Task	81.98	81.77

Table 3: Results of Ablation study. w/o means without a specified module. w/ means replace an original module with a simpler module with the same function.

Train Shuffle	Valid Shuffle	Validation Accuracy	Validation FEVER score
✔	✗	82.63	82.42
✔	✔	82.67	82.46
✔	Rev.	82.38	82.17
✗	✗	83.28	83.07
✗	✔	76.28	76.07
✗	Rev.	72.56	72.35

Table 4: The accuracy of FEVER score of the validation set with different evidence shuffling settings. Rev. means reverse the order of evidence candidates.

able performance.

4.3 Ablation Study

For ablation, we compare with the following model variants to examine the effectiveness of each component. (1) w/ Dep-based Decomposition. We use simple rule-based evidence decomposition method with the "conjunction" relationship in dependency parsing instead of LaserTagger (elaborated in Appendix C). (2) w/o Evidence Decomposition. We use the original retrieved evidence without splitting them into sub-evidence pieces. (3) w/o Graph Interaction. We remove the graph interaction module, and directly use the outcomes of PLM for veracity prediction and gold evidence prediction. (4) w/o Auxiliary Task. We remove the auxiliary task, gold evidence prediction, and optimize the model solely with the veracity prediction target.

In Table 3, without each component, the performance on the validation set drops consistently, demonstrating the effectiveness of our proposed modules. Particularly, without evidence decomposition, our model performs worse on the validation set by 0.79% and 0.78% in accuracy and FEVER score. Organizing the evidence with similar granularity with the given claim, our model could capture the relation between evidence and the claim more precisely. With a rule-based evidence splitter, the performances slightly drop, but are still better than the setting without evidence decomposition. Even rule-based evidence decomposition works for our proposed method, and more sophisticated evidence splitter brings further improvements. Without the auxiliary gold evidence prediction task, both the label accuracy and FEVER score drop by 0.69%. This demonstrates the auxiliary task indeed push the model to learn focusing on crucial evidence, which is beneficial to veracity prediction.

5 Analysis

On both training and validation set, experiments show an original evidence sentence is split into 1.8 sub-evidence pieces on average, which means this evidence decomposition clarifies the vital information without overly splitting the original sentence. AS the decomposition results are still at sentence level, the idea of evidence decomposition can be simply extended to other models.

5.1 Evidence Shuffling

In this section, we further explore the impact of the evidence shuffling method, and the results are shown in Table 4. When training with shuffled evidence, regardless of whether we use evidence shuffling in the validation stage, the results are stable. However, if we keep the evidence order in the training step but shuffle evidence in the validation set, all metrics drop significantly, with a decrease of 7% on both accuracy and FEVER score. When we further reverse the order of the evidence candidates, the gap expands to 10.73%.

After examining the evidence candidates, we discovered that gold evidence in the training set consistently receiving extreme high scores and others are very low. Because the gold evidence in the training set have been used as the ground truth in the second step of FEVER, the evidence extraction model is overfitting to them. The evidence extraction model gives all of the gold evidence pieces a score of over 0.999, while others are always below 0.5. In the training phase, if a claim needs s evidence to be verified, these s sentences are ranked the top-s according to the ranking scores provided by the evidence extraction model. Without shuffle, the verification model learns to only concentrate on the evidence close to the given claim, which will be harmful to the model's generalization ability.

People are inserting gold evidence to the extracted evidence collection and giving them the highest ranking score, making things even worse. The validation set contains the same position bias as in the training set. That is why we get higher accuracy and FEVER score on the validation set when the training and validation set are both unshuffled. Models have no chance to learn in hard mode and the validation set fails to be a reasonable selection criterion.

Therefore, evidence shuffling changes the location distribution of gold evidence in the evidence candidates, guiding the verification model to consider all evidence candidates. It also makes the validation set a more reasonable criterion to select the final model. Meanwhile, it allows the evidence extraction model to insert the gold evidence pieces to the evidence candidates, which alleviates the mismatch between the extracted evidence set and the veracity label.

5.2 Influence of Gold Evidence Lengths

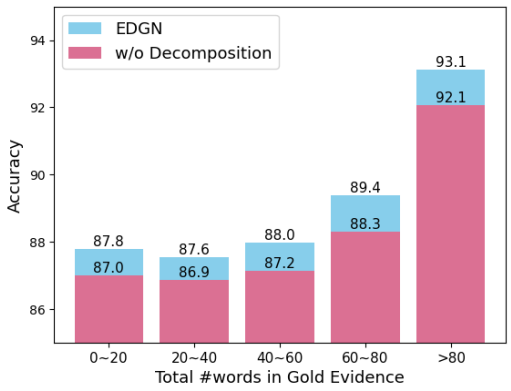

Figure 2: Model performance with respect to different lengths of gold evidence on the validation set. Blue bar is our full model, while red bar is our model without evidence decomposition.

We compare the performance of our model with and without evidence decomposition on instances with gold evidence of different lengths. The results are shown in Figure 2. As there is no gold evidence set for instances labeled NOT ENOUGH INFO, we only care about instances that can be verified, namely the SUPPORTS and REFUTES instances. Our model achieves higher results on all gold evidence lengths. When longer evidence is required, the accuracy improvement is more obvious, with an increase of 1.1% on instances required 60 to 80 words and 1.0% on instances required more than 80 words to verify respectively. Although in these cases, the total text length a model should concentrate on is very long, informative spans in each evidence piece are shorter (Portelli et al., 2020). The evidence decomposition mechanism allows models to ignore these sub-evidence pieces without key information. For claims that only need one or two evidence pieces to be verified, evidence decomposition can also highlight the key information, reaching an average of 0.77% improvement on instances with fewer than 60 evidence words needed. The evidence decomposition mechanism shows more advantages when the gold sentence is longer or more complex.

5.3 Impact of Evidence Decomposition Quality

We use Lasertagger (Malmi et al., 2019) to decompose evidence candidates without further fine-tuning on the FEVER dataset. The Exact Score and SARI on the WikiSplit dataset (Botha et al., 2018) is 14.42% and 61.11% respectively. Exact Score is the percentage of exactly correctly predicted fusions, and SARI (Xu et al., 2016) computes the average F1 scores of the added, kept and deleted n-grams. Detailed results of the decomposition model are shown in Table 8 in the Appendix B. The quality of evidence decomposition has a great impact on the performance of our verification model.

For the example as shown in Table 5, evidence decomposition highlights the key information in long sentences without much loss of sentence meaning. However, LaserTagger does not replace pronouns with the exact name of the subject. As shown in the second sub-evidence, the pronoun "they" are referred to without specification. In most cases, it is reasonable to assume that free pronouns referred to the title of the Wikipedia page. However, models will easily make mistakes when it is not the case. When we replace the claim with "Wildfang previously worked at Nike", the model still predicts SUPPORT.

Meanwhile, as shown in Table 8 of Appendix B, the overall metrics of the decomposition results have a lot of room for improvement. For example, the best sub-evidence pieces of the gold evidence in Table 1 should be *"Love the Way You Lie" is a song recorded by the American rapper Eminem.*, *"Love the Way You Lie" features the Barbadian singer Rihanna.* and *"Love the Way You Lie" is from Eminem's seventh studio album Recovery (2010).* If

a better sentence splitting method and appropriate fine-tuning methods are introduced, EDGN has an essential of getting more improvement.

Claim: Wildfang was founded in 2010.
Label: SUPPORT
Gold Evidence: **[Wildfang]** The company was founded in 2010 by Emma Mcilroy and Julia Parsley , who previously worked at Nike , Inc. in Portland , Oregon.
Decomposed Gold Evidence: **[Wildfang]** The company was founded in 2010 by Emma Mcilroy and Julia Parsley . **[Wildfang]** they previously worked at Nike , Inc. in Portland , Oregon.

Table 5: An example with decomposed gold evidence.

6 Related Works

Many earlier attempts in FEVER view the task as an extension of the natural language inference (NLI) task. Previous works solves it following the NLI pattern (Hanselowski et al., 2018; Nie et al., 2019; Soleimani et al., 2020). The given claim is analogied to the premise, and a piece of evidence the hypothesis. They predict a label for each claim and evidence pair and synthesis them using rule-based or NN-based methods to get the final prediction. However, methods at this pattern disallow evidence-evidence interaction.

Zhou et al. (2019) introduce graphs to this task. They denote each evidence sentence with one node and construct a fully-connected graph with these nodes. Liu et al. (2020) try a mixture of token-level and sentence-level kernel graph attention to allow fine-grained interaction. They consider all evidence candidates at once, but graphs at the sentence level are too coarse-grained for the long extracted evidence sentences. Fine-grained, namely an SRL-based phrase-level graph construction method is presented by Zhong et al. (2020). However, it is over-fragmented and leaves some semantic and syntactical information in the original evidence sentence, especially with the fully-connected edges within each verb and all its arguments.

Researchers also try to manipulate the claim and evidence more skillfully. Chen et al. (2020) notice that there are many aspects in a claim which could be verified respectively. They perform task type transformation and transform the task into a pipeline of Question Generation, Question-Answer, and other tasks. Portelli et al. (2020) find some spans in evidence are crucial for claim verification. They identify these spans and concatenate them to the original evidence sentence for emphasis. Kruengkrai et al. (2021) advance multi-level self-attention within evidence pieces perform well. However, no previous works attempt to decompose the evidence candidate into multiple relatively complete parts explicitly.

From another perspective, after Zhou et al. (2019) transform fact verification from a synthesis of NLI results, there are two main ways to form the input sequence, from which we can get the evidence embedding vectors. One way is to form a set of claim-evidence piece pair (Nie et al., 2019; Zhou et al., 2019; Liu et al., 2020), and another is to concatenate the claim and all evidence candidates (Zhong et al., 2020; Kruengkrai et al., 2021). Experiments show models with the later input method get better results for it allows early interaction between the claim and evidence pieces. However, previous researchers pay no attention to the evidence order and only arrange them according to the ranking score provided by the evidence extraction model.

Contributing to the development of PLMs, larger and stronger PLM is introduced to this task. T5 Listwise (Jiang et al., 2021) propose to apply T5 to the concatenated sequence of the given claim and the candidates in all three steps of FEVER. Because its evidence candidates set is totally different from ours and the T5 is a stronger PLM, it is unfair to compare the results directly.

7 Conclusion

In this paper, we propose EDGN, an evidence decomposition graph network for fact verification. It decomposes evidence candidates into sub-evidence pieces and re-organizes them in a graph to encourage their further interactions. The evidence decomposition method highlights the key information in longer evidence sentences without much expense of the original sentence structure and meaning. Meanwhile, we find the exposure bias issues introduced by the evidence extraction step in the FEVER pipeline and propose evidence shuffling, a simple but effective approach to help our EDGN learn to recognize crucial pieces from the evidence candidates. Experiments show that our method can make better and full use of evidence candidates, especially those longer or more complex ones, thus achieving state-of-the-art performance.

References

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

Jiangjie Chen, Qiaoben Bao, Jiaze Chen, Changzhi Sun, Hao Zhou, Yanghua Xiao, and Lei Li. 2020. LOREN: logic enhanced neural reasoning for fact verification. *CoRR*, abs/2012.13577.

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, Brussels, Belgium. Association for Computational Linguistics.

Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with T5 for fact verification. In *ACL/IJCNLP (2)*, pages 402–410. Association for Computational Linguistics.

Canasai Kruengkrai, Junichi Yamagishi, and Xin Wang. 2021. A multi-level attention model for evidence-based fact checking. In *ACL/IJCNLP (Findings)*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2447–2460. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6859–6866. AAAI Press.

Beatrice Portelli, Jason Zhao, Tal Schuster, Giuseppe Serra, and Enrico Santus. 2020. Distilling the evidence to augment fact verification models. In *Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER)*, pages 47–51, Online. Association for Computational Linguistics.

Jiasheng Si, Deyu Zhou, Tongzhe Li, Xingyu Shi, and Yulan He. 2021. Topic-aware evidence reasoning and stance-aware aggregation for fact verification. In *ACL/IJCNLP (1)*, pages 1612–1622. Association for Computational Linguistics.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. BERT for evidence retrieval and claim verification. In *ECIR (2)*, volume 12036 of *Lecture Notes in Computer Science*, pages 359–366. Springer.

Dominik Stammbach and Elliott Ash. 2020. e-fever: Explanations and summaries forautomated fact checking. In *TTO*.

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, 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.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. 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.

A FEVER Statistics

The numbers of instances in the training, validation and test set of each class are shown in Table 6.

	SUPPORTED	REFUTED	NEI
Train	80,035	29,775	35,639
Dev	6,666	6,666	6,666
Test	6,666	6,666	6,666

Table 6: Statistics of FEVER dataset.

The Evidence extraction results from Liu et al. (2020) of each set are shown in Table 7.

	Top-5 Precision	**Top-5 Recall**	**Top-5 F1**
Train	32.14	99.59	48.59
Dev	27.29	94.37	42.34
Dev*	30.58	99.66	46.82
Test	25.21	87.47	39.14

Table 7: Evidence extraction results of Liu et al. (2020). Dev* is that after gold evidence insertion.

B Detailed Results of the Decomposition Model

The detailed results of LaserTagger, our sentence decomposition model, on the WikiSplit dataset are shown in Table 8. Since no checkpoint is provided by its author, we re-train the model with their code and dataset. The Exact score and SARI score are less than 1% lower compared to results reported in their paper.

Exact score	14.420
SARI score	61.112
KEEP score	93.033
ADDITION score	31.218
DELETION score	59.086

Table 8: Sentence splitting and rephrasing results on WikiSplit dataset.

C Rule-based Decomposition Method

We get the dependency structure of each evidence sentence with Spacy. Then we find several conjunction sets in the dependency structure. Each conjunction set contains several words connected with the "conj" edge. As we want to decompose the evidence sentence at clause level, we keep the conjunction set with the max distance of words in it. When we get a sub-evidence piece, we keep one word and remove others in the selected conjunction set. For words not in the conjunction set, we remove a word if it depends on a removed word and the relationship is not "conj", recursively, and keeps other words. Therefore, if there are n words in the kept conjunction set, the original evidence sentence is decomposed to n sub-evidence pieces.