FACTKG: Fact Verification via Reasoning on Knowledge Graphs

Jiho Kim[1], **Sungjin Park**[1], **Yeonsu Kwon**[1], **Yohan Jo**[2]*, **James Thorne**[1], **Edward Choi**[1]
[1]KAIST [2]Amazon
{jiho.kim, zxznm, yeonsu.k, thorne, edwardchoi}@kaist.ac.kr
jyoha@amazon.com

Abstract

In real world applications, knowledge graphs (KG) are widely used in various domains (e.g. medical applications and dialogue agents). However, for fact verification, KGs have not been adequately utilized as a knowledge source. KGs can be a valuable knowledge source in fact verification due to their reliability and broad applicability. A KG consists of nodes and edges which makes it clear how concepts are linked together, allowing machines to reason over chains of topics. However, there are many challenges in understanding how these machine-readable concepts map to information in text. To enable the community to better use KGs, we introduce a new dataset, FACTKG: Fact Verification via Reasoning on Knowledge Graphs. It consists of 108k natural language claims with five types of reasoning: One-hop, Conjunction, Existence, Multi-hop, and Negation. Furthermore, FACTKG contains various linguistic patterns, including colloquial style claims as well as written style claims to increase practicality. Lastly, we develop a baseline approach and analyze FACTKG over these reasoning types. We believe FACTKG can advance both reliability and practicality in KG-based fact verification.[1]

1 Introduction

The wide spread risk of misinformation has increased the demand for fact-checking, that is, judging whether a claim is true or false based on evidence. Accordingly, recent works on fact verification have been developed with various sources of evidence, such as text (Thorne et al., 2018; Augenstein et al., 2019; Jiang et al., 2020; Schuster et al., 2021; Park et al., 2021) and tables (Chen et al., 2019; Wang et al., 2021; Aly et al., 2021). Unfortunately, knowledge graphs (KG), one of the large-scale data forms, have not yet been fully utilized as a source of evidence. A KG is a valuable knowledge source due to two advantages.

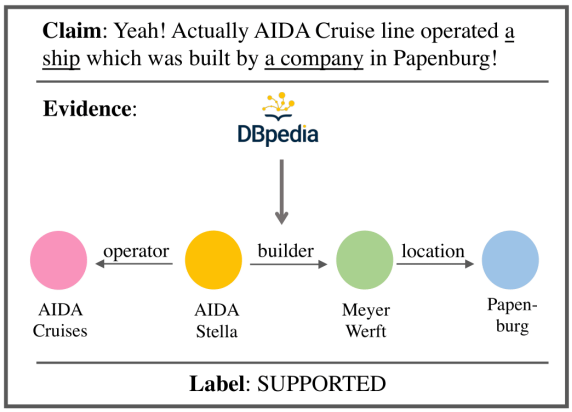


Figure 1: An example data from FACTKG. To verify the claim whether it is SUPPORTED or REFUTED, we use triples extracted from DBpedia as evidence.

Firstly, KG-based fact verification can provide more reliable reasoning: since the efficacy of real-world fact-checking hinges on this reliability, recent studies have focused on justifying the decisions of a fact verification system (Kotonya and Toni, 2020a). In most existing works, the justification is based on the extractive summary of text evidence. Therefore, the inferential links between the evidence and the verdict are not clear (Kotonya and Toni, 2020b; Atanasova et al., 2020a,b). Compared to text and tables, a KG can simply represent reasoning process with logic rules on nodes and edges (Liang et al., 2022). This allows us to categorize common types of reasoning with the graphical structure, as shown in Table 1.

Secondly, KG-based fact verification techniques have broad applicability beyond the domain of fact-checking. For example, modern dialogue systems (e.g. *Amazon Alexa* (Amazon Staff, 2018), *Google Assistant* (Kale and Hewavitharana, 2018)) maintain and communicate with internal knowledge graphs, and it is crucial to make sure that their content is consistent with what the user says and oth-

* This work is not associated with Amazon.
[1]Data available at https://github.com/jiho283/FactKG.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 16190–16206

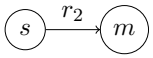
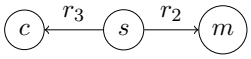
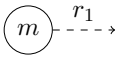
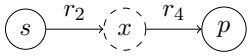
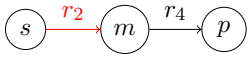
Reasoning Type	Claim Example	Graph
One-hop	AIDAstella was built by Meyer Werft.	
Conjunction	AIDA Cruise line operated the AIDAstella which was built by Meyer Werft.	
Existence	Meyer Werft had a parent company.	
Multi-hop	AIDAstella was built by a company in Papenburg.	
Negation	AIDAstella was not built by Meyer Werft in Papenburg.	

Table 1: Five different reasoning types of FACTKG. r_1: parentCompany, r_2: shipBuilder, r_3: shipOperator, r_4: location, m: Meyer Werft, s: AIDAstella, c: AIDA Cruises.

erwise update the knowledge graphs accordingly. If we model the user's utterance as a claim and the dialogue system's internal knowledge graph as a knowledge source, the process of checking their consistency can be seen as a form of KG-based fact verification task. More generally, KG-based fact verification techniques can be applied to cases which require checking the consistency between graphs and text.

Reflecting these advantages, we introduce a new dataset, FACTKG: Fact Verification via Reasoning on Knowledge Graphs, consisting of 108k textual claims that can be verified against DBpedia (Lehmann et al., 2015) and labeled as SUPPORTED or REFUTED. We generated the claims based on graph-text pairs from WebNLG (Gardent et al., 2017) to incorporate various reasoning types. The claims in FACTKG are categorized into five reasoning types: One-hop, Conjunction, Existence, Multi-hop, and Negation. Furthermore, FACTKG consists of claims in various styles including colloquial, making it potentially suitable for a wider range of applications, including dialogue systems.

We conducted experiments on FACTKG to validate whether graph evidence had a positive effect for fact verification. Our experiments indicate that the use of graphical evidence in our model resulted in superior performance when compared to baselines that did not incorporate such evidence.

2 Related Works

2.1 Fact Verification and Structured Data

There are various types of knowledge used in fact verification such as text, tables, and knowledge graphs. Research on fact verification has mainly focused on text data as evidence (Thorne et al., 2018; Augenstein et al., 2019; Jiang et al., 2020; Schuster et al., 2021; Park et al., 2021). FEVER (Thorne et al., 2018), one of the representative fact verification datasets, is a large-scale manually annotated dataset derived from Wikipedia. Other recent works leverage ambiguous QA pairs (Park et al., 2021), factual changes (Schuster et al., 2021), multiple documents (Jiang et al., 2020), or claims sourced from fact checking websites (Augenstein et al., 2019). Fact verification on table data is also studied (Chen et al., 2019; Wang et al., 2021; Aly et al., 2021). Table-based datasets such as SEM-TAB-FACTS (Wang et al., 2021) or TabFact (Chen et al., 2019) require reasoning abilities over tables, and FEVEROUS (Aly et al., 2021) validate claims utilizing table and text sources. We refer the reader to Guo et al. (2022) for a comprehensive survey.

There have been several tasks that utilize knowledge graphs (Dettmers et al., 2018). For example, FB15K (Bordes et al., 2013), FB15K-237 (Toutanova and Chen, 2015), and WN18 (Bordes et al., 2013) are built upon subsets of large-scale knowledge graphs, Freebase (Bollacker et al., 2008) and WordNet (Miller, 1995) respectively. These datasets only use a single triple as a claim, and thus the claims only require One-hop reasoning. However, FACTKG is the first KG-based fact verification dataset with natural language claims that require complex reasoning. In terms of the evidence KG size, FACTKG uses the entire DBpedia (0.1B triples), which is significantly larger than previous datasets (FB15K: 592K, FB15K-237: 310K, WN18: 150K).

2.2 WebNLG

As constructing a KG-based fact verification dataset requires a paired text-graph corpus, we utilized WebNLG as a basis for FACTKG. WebNLG

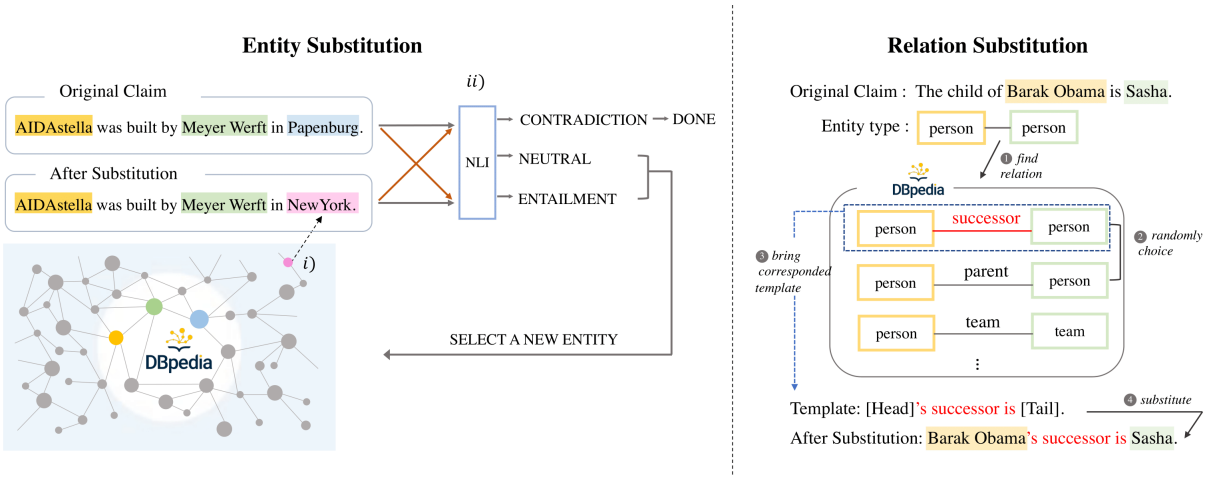


Figure 2: Two substitution methods utilized in FACTKG. In *Entity substitution*, we select a new entity located in outside 4-hops from all entities in the original claim. If the results of bidirectional NLI are both contradiction, we finish this process. In *Relation substitution*, we randomly extract a relation that takes the same entity types for the head and tail as the original relation. Then, substitution is performed based on a template specific to the selected relation.

is a dataset for evaluating triple-based natural language generation, which consists of 25,298 pairs of high-quality text and RDF triples from DBpedia. WebNLG contains diverse forms of graphs and the texts are created by linguistic experts, which gives it great variety and sophistication. In the 2020 challenge[2], the dataset has been expanded to 45,040 text-triples pairs. We used this 2020 version of WebNLG when constructing our dataset.

3 Data Construction

Our goal is to diversify the graph reasoning patterns and linguistic styles of the claims. To achieve this, we categorize five reasoning types of claims: One-hop, Conjunction, Existence, Multi-hop, and Negation. Our claims are generated by transforming the sentences in S_w, a subset of WebNLG's text-graph pairs (Section 3.1).[3] Next, we also diversified the claims with colloquial style transfer and presupposition (Section 3.2).

3.1 Claim Generation

3.1.1 One-hop

The most basic type of claim is one-hop, which covers only one knowledge triple. One-hop claims can be verified by checking the existence of a single corresponding triple. In the second row of Table 1, the claim is SUPPORTED when the triple (*AIDAstella*, *ShipBuilder*, *Meyer Werft*) exists.

We take the sentences that consist of a single triple in S_w as SUPPORTED claims. REFUTED claims are created by substituting SUPPORTED claims in two ways: *Entity substitution* and *Relation substitution*. In *Entity substitution*, we replace an entity e in SUPPORTED claim C with another entity $\tilde{e}$ of the same entity type. In order to ensure that the label of the substituted sentence $\tilde{C}$ is REFUTED, the entity $\tilde{e}$ should satisfy the following two conditions. *i)* To select $\tilde{e}$ that is irrelevant to C, $\tilde{e}$ is outside 4-hops from all entities in C on DBpedia, *ii)* the results of NLI (C, $\tilde{C}$) and NLI ($\tilde{C}$, C) are both CONTRADICTION.[4] In *Relation substitution*, we replace a relation in the SUPPORTED claim with another relation. We replace the relation of a triple in the claim with another relation that takes the same entity types for the head and tail as the original relation (e.g. currentTeam ↔ formerTeam). The four groups of compatible relations are listed in Table 6. The overall process of the substitution methods is illustrated in Figure 2.

[2] https://webnlg-challenge.loria.fr/challenge_2020/

[3] We found that 99.7% of claims in FEVER and FEVEROUS consist of a single sentence. To reflect this result, we extract a subset S_w containing only single sentences from WebNLG.

[4] We use a natural language inference (NLI) model, RoBERTa-base (Liu et al., 2019) finetuned on the MNLI dataset (Williams et al., 2018). The notation NLI (p, h) represents the result of NLI when p is assigned as the premise and h as the hypothesis.

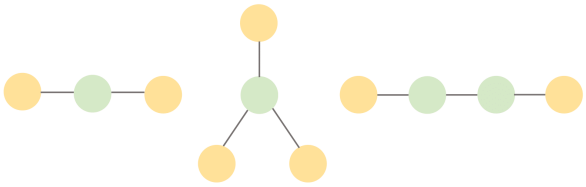


Figure 3: Graph patterns used in Conjunction and Multi hop claims.

3.1.2 Conjunction

A claim in the real world can include a mixture of different facts. To incorporate this, we construct a conjunction claim composed of multiple triples. Conjunction claims are verified by the existence of all corresponding triples. In the third row of Table 1, the claim can be divided into two parts: *"AIDA Cruise line operated the AIDAstella."* and *"AIDAstella was built by Meyer Werft."*. The claim is SUPPORTED when all the triples (*AIDAstella*, *ShipOperator*, *AIDA Cruises*), (*AIDAstella*, *ShipBuilder*, *Meyer Werft*) exist. To implement this idea, we extracted sentences consisting of more than one triple from S_w and used them as the SUPPORTED claims. To create REFUTED claims, we use *Entity substitution* method on these SUPPORTED claims.

3.1.3 Existence

People may make claims that assert the existence of something (e.g. *"She has two kids."*). From the view of a triple, this corresponds to the head or tail missing. To reflect this scenario, we formulate a claim by extracting only {*head*, *relation*} or {*tail*, *relation*} from a triple. Existence claims are generated using templates and they are divided into two categories: *head-relation* (e.g. template: {*head*} had a(an) {*relation*}.) and *tail-relation* (e.g. template: {*tail*} was a {*relation*}.). SUPPORTED claims are constructed by randomly extracting {*head*, *relation*} or {*tail*, *relation*} in triples from S_w. The REFUTED claims are constructed using the same type of entities as represented in the claim, but with different relations. However, it is possible that unrealistic claims may be generated in this manner. For example, *"Meyer Werft had a location."* or *"Papenburg was a location."* can be created from the triple (*Meyer Werft*, *location*, *Papenbug*). Hence, we selected 22 relations out of all relations that lead to realistic claims. Templates used for both categories and examples of generated claims are in Table 7.

3.1.4 Multi-hop

We also consider multi-hop claims that require the validation of multiple facts where some entities are underspecified. Entities in this claim can be connected by a sequence of relations. For example, the multi-hop claim in Table 1 is SUPPORTED if the triple (*AIDAstella*, *ShipBuilder*, x) and the triple (x, *location*, *Papenburg*) are present in the graph. The goal is to verify the existence of a path on the graph that starts from *AIDAstella* and reaches *Papenburg* through the relations *ShipBuilder* and *location*.

Figure 3 shows how a SUPPORTED multi-hop claim C_M can be generated by replacing an entity e of the conjunction claim C with its type name. First, an entity e is selected from the green nodes. Then, the type name t of the entity e is extracted from DBpedia. However, each entity e in DBpedia has several types $T = \{t_1, t_2, ..., t_N\}$, and it is not annotated which type is relevant when e is used in a claim. So it is necessary to select one of them. For each $t_n \in T$, we insert it next to the entity e in the claim C and measure the perplexity score of the modified claim using GPT2-large (Radford et al., 2019). Then we replace e in the claim with the type name that had the lowest score. The REFUTED claim is generated by applying *Entity substitution* to the SUPPORTED claim.

3.1.5 Negation

For each of the four methods for generating claims, we develop claims that incorporate negations.

One-hop We use the Negative Claim Generation Model (Lee et al., 2021) which was fine-tuned on the opposite claim set in the WikiFactCheck-English dataset (Sathe et al., 2020).[5] To ensure the quality of the generated sentences, we generate 100 opposing claims for each original claim, then only use those that preserve all entities, and contain negations (e.g. *'not'* or *'never'*). Also, similar to *Entity substitution* method, we only use sentences whose NLI relation with the original sentences are CONTRADICTION bidirectionally. When a negation is added, the label of the generated claim is reversed from the original claim.

Conjunction The use of negations (i.e., *'not'*) in various positions within conjunction claims allows the generation of a wide range of negative claim structures. We employ the pretrained language model GPT-J 6B (Wang and Komatsuzaki,

[5]WikiFactCheck-English consists of pairs of claims, with a positive claim and its corresponding negative claim.

2021) to attach negations to the claim. We construct 16 in-context examples, each with negations attached to the texts corresponding to the first or/and second relation. When a negation is added to the SUPPORTED claims, all the claims become REFUTED. However, when it is added to REFUTED claims, the label depends on the position of the negation. When negations are added to all parts with substituted entities, it becomes a SUPPORTED claim. Conversely, other cases preserve the label REFUTED since the negation is added to a place that is not related to entity substitution. A detailed labeling strategy is described in Appendix D.1.

Existence The claim is formulated by adding a negation within the templates presented in Section 3.1.3 (e.g. {*tail*} was not a {*relation*}.).

Multi-hop A claim is formulated using the GPT-J with in-context examples, similar to conjunction. The truth of this claim is dependent on the presence of a distinctive path that matches the claim's intent. For example, the negative claim *"AIDAstella was built by a company, not in Papenburg."* is SUPPORTED if x exists where the triples (*AIDAstella*, *ShipBuilder*, x) and (x, *location*, y) are in DBpedia and y is not *Papenburg*. A more detailed labeling strategy is in Appendix D.2.

3.2 Colloquial Style Transfer

We transform the claims into a colloquial style via style transfer using both a fine-tuned language model and presupposition templates.

3.2.1 Model based

Using a similar method proposed by Kim et al. (2021), we transform the claim obtained from 3.1 into a colloquial style. For example, the claim *"Obama was president."* is converted to *"Have you heard about Obama? He was president!"*.

We train FLAN T5-large (Chung et al., 2022) to generate a colloquial style sentence given a corresponding written style sentence from Wizard of Wikipedia (Dinan et al., 2019). However, using sentences generated by the model could have several potential issues: *i)* the original and generated sentences are lexically the same, *ii)* some entities are missing in the generated sentences, *iii)* the generated sentences deviate semantically from the original, *iv)* the generated sentences lack a colloquialism, as mentioned in Kim et al. (2021). To overcome this, we oversample candidate sentences and utilize an additional filtering process.

First, to make more diverse samples using the model, we set the temperature to 20.0 and generate 500 samples with beam search. *i)* To avoid generated sentences that are too similar to the original sentences, only sentences with an edit distance of 6 or more from the original sentence are selected among 500 samples. *ii)* Then, only those that have verbs and the named entities all preserved are selected.[6] *iii)* Finally, we use bidirectional NLI to preserve the original semantics. Candidate sentences survive when NLI (O, G) is ENTAILMENT and NLI (G, O) is not CONTRADICTION where O refers to the original sentence and G the generated sentence. On average, only 41.2 generated sentences survived out of 500 samples. Additionally, in cases where none of the 500 generated sentences pass the filtering process, we include the original claim in the final dataset as a written style claim. Following the filtering process, the AFLITE method (Sakaguchi et al., 2019), which utilizes adversarial filtering, is applied to select the most colloquial style sentence among the surviving sentences. We include the selected claim in the final dataset as a colloquial style claim.

3.2.2 Presupposition

A presupposition is something the speaker assumes to be the case prior to making an utterance (Yule and Widdowson, 1996). People often communicate under the presupposition that their beliefs are universally accepted. We construct claims using this form of utterance. The claims in FACTKG are focused on three types of presupposition: factive, non-factive, and structural presuppositions.

Factive Presupposition People frequently use verbs like *"realize"* or *"remember"* to express the truth of their assumptions. The utterance *"I remembered that* {Statement}." assumes that {Statement} is true. Reflecting these features, a new claim is created by appending expressions that contain presupposition (e.g. *"I realized that" or "I wasn't aware that"*) to the existing claim. We used eight templates to make factive presupposition claims: the details are appended in Table 8.

Non Factive Presupposition The verbs such as *"wish"* are commonly used in utterances that describe events that have not occurred. For example, people say *"I wish that* {Statement}." when {Statement} did not happen. Claims that are created by

[6] NLTK (Bird et al., 2009) POS tagger and Stanza (Qi et al., 2020) NER module are used. DBPedia entities are already tagged in each claim, but not all entities exist in the sentence in their raw form, so the NER module is used.

the non-factive presupposition method are labeled as the opposite of the original one. We used three templates to make these claims: the templates are appended in Table 8.

Structural Presupposition This type is in the form of a question that presumes certain facts. We treat the question itself as a claim. For example, *"When was Messi in Barcelona?"* assumes that *Messi was in Barcelona*. To create a natural sentence form, only claims corresponding to one-hop and existence are constructed. For the one-hop claim, a different template was created corresponding to each relation reflecting its meaning (e.g. "When did {*head*} die from {*tail*}?" for the relation *deathCause* and "When was {*head*} directed by {*tail*}?" for relation *director*). Existence claims are also generated based on templates (e.g. "When was {*tail*} {*relation*}?") using pairs of *head-relation* or *tail-relation*, similar to Section 3.1.3. The templates used are described in Table 9.

3.3 Quality Control

To evaluate the quality of our dataset, the labeling strategy and the output of the colloquial style transfer model are assessed.

Labeling Strategy When SUPPORTED claims are made in the manner described in Section 3.1, the labeling is straightforward, as all have precise evidence graphs. However, REFUTED claims are generated by random substitution, so there might be a small chance that they remain SUPPORTED (e.g. *"The White House is in Washington, D.C."* to *"The White House is in America."*). To evaluate this substitution method, randomly sampled 1,000 substituted claims were reviewed by two graduate students. As a result, 99.4% of generated claims were identified as REFUTED by both participants.

Colloquial Style Transfer Model We also evaluate the quality of the colloquial style claims generated by the model. A survey was conducted on all claims in the test set by three graduate students. As a result, only 9.8% of the claims were selected as *Loss of important information* by at least two reviewers. In addition, to ensure the quality of the test set, only claims that were selected as *All facts are preserved* by two or more reviewers are included in the test set. The survey details are in Appendix E.

Type	Written	Colloquial		Total
		Model	Presup	
One-hop	2,106	15,934	1,580	19,530
Conjuction	20,587	15,908	602	37,097
Existence	280	4,060	4,832	9,172
Multi-hop	10,239	16,420	603	27,262
Negation	1,340	12,466	1,807	15,613
Total	34,462	64,788	9,424	108,674

Table 2: Dataset statistics of FACTKG for all reasoning types.

4 Experiments

4.1 Dataset Statistics

Table 2 shows the statistics of FACTKG. We split the claims into train, dev, and test sets with a proportion of 8:1:1. We ensured that the set of triples in each split is disjoint with the ones in other splits.

4.2 Experimental Setting

We publish FACTKG with sets of claims, graph evidence and labels. The graph evidence includes entities and a set of relation sequences connected to them. For instance, when the claim is given as *"AIDAstella was built by a company in Papenburg."*, the entity *'AIDAstella'* corresponds to a set of relation sequence [*shipBuilder*, *location*] and *'Papenburg'* corresponds to [∼*location*, ∼*shipBuilder*].[7] In the test set, we only provide entities as graph evidence.

4.3 Baseline

We conduct experiments on FACTKG to see how the graphical evidence affects the fact verification task. To this end, we divided our baselines into two distinct categories based on the input type, *Claim Only* and *With Graphical Evidence*.

4.3.1 Claim Only

In the *Claim Only* setting, the baseline models receive only the claim as input and predict the label. We used three transformer-based text classifiers, BERT, BlueBERT, and Flan-T5. BERT (Devlin et al., 2018) is trained on Wikipedia from which DBpedia is extracted. So we expect that the model will use evidence memorized in its pretrained weights (Petroni et al., 2019) or exploit structural patterns in the generated claims (Schuster et al., 2019; Thorne and Vlachos, 2021). BlueBERT (Peng et al., 2019) is trained on biomedical corpus, such as Pubmed abstracts. We use

[7] '∼' indicates that the direction of the relation is reversed

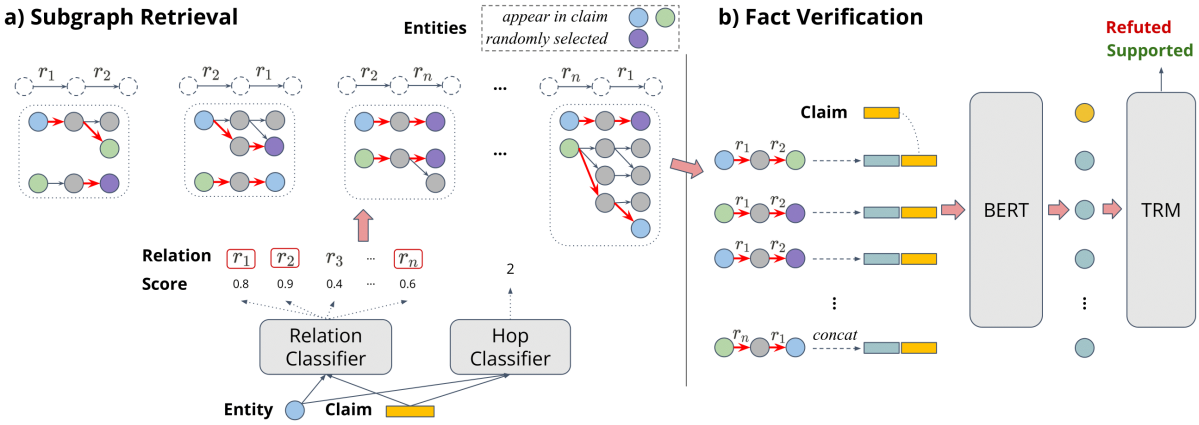


Figure 4: Overall process of our baseline. In the subgraph retrieval step, each classifier respectively predicts the relations and hops related to the given entity and the claim. Subsequently, we check all the n-hop relation sequences obtained from each classifier to find all evidence paths. In the fact verification step, the claim is verified by leveraging all outputs obtained from the subgraph retrieval step. In this figure, we denote Transformer Encoder as TRM.

BlueBERT as a comparator for BERT since it has never seen Wikipedia during its pre-training. Flan-T5 (Chung et al., 2022) is an enhanced version of T5 (Raffel et al., 2022) encoder-decoder that has been fine-tuned in a mixture of 1.8K tasks. In all experiments, we fine-tune BERT and BlueBERT on our training set. Different from BERT and BlueBERT, we use Flan-T5 in the zero-shot setting. For this setting, we use *"Is this claim True or False? Claim: "* as the prefix. Then, we measure the probability that tokens *True* and *False* will appear in the output. Among the two tokens, we choose the one with the higher probability.

4.3.2 With Graphical Evidence

In the *With Graphical Evidence* setting, the model receives the claim and graph evidence as input and predicts the label. The baseline we used is a framework proposed by GEAR (Zhou et al., 2019) that enables reasoning on multiple evidence texts. Since GEAR was originally designed to reason over text passages, we change components to suit KG. The modified GEAR consists of the subgraph retrieval module and the claim verification module. The pipeline of the modified GEAR is illustrated in Figure 4.

Subgraph retrieval We replace document retrieval and sentence selection in GEAR with subgraph retrieval. To retrieve graphical evidence, we train two independent BERT models, namely a relation classifier and a hop classifier. The relation classifier predicts the set of relations R from the claim c and the entity e. The hop classifier is designed to predict the maximum number of hops n to be traversed from e. We take the subgraph of G that are composed only of the relations in R and where the terminal nodes are entities in C and less than n hops apart from e, allowing for duplicates and considering the order. By traversing the knowledge graph starting from e along the relation sequences in P, we choose the paths that can reach another entity that appears in the claim. If none of the paths is reachable to other entities, then we randomly choose one of the paths. The strategy we used enables the model to retrieve supported evidence and counterfactual evidence for the given claim. The following example is presented to assist the understanding of our subgraph retrieval method. The example claim in Section 4.2 consists of two entities, *'AIDAstella'* and *'Papenburg'*. In this setting, the hop classifier must predict 2 since those entities are connected by a sequence of two relations, namely *shipBuilder* and *location*. In addition, the relation classifier must predict correctly predict those two relations. After that, we find all 2-hop paths starting from *'AIDAstella'* along the predicted relations in the knowledge graph. If there is a path that reaches *'Papenburg'*, we can use it as supporting evidence. If not, however, we randomly select a path.

Fact verification We directly employed the claim verification in GEAR and applied some changes to suit the KG setting. Since our evidence is a set of graph paths, we converted them to text by concatenating each triple with the special token <SEP>. We also found that ERNet in GEAR is identical

Input Type	Model	One-hop	Conjunction	Existence	Multi-hop	Negation	Total
Claim Only	BERT	69.64	63.31	61.84	70.06	63.62	65.20
	BlueBERT	60.03	60.15	59.89	57.79	58.90	59.93
	Flan-T5	62.17	69.66	55.29	60.67	55.02	62.70
With Evidence	GEAR	83.23	77.68	81.61	68.84	79.41	77.65

Table 3: Fact verification accuracy on FACTKG.

to the Transformer encoder, so we replaced it with a randomly initialized Transformer encoder. To make this paper self-contained, we provide further details about the claim verification of GEAR in Appendix F.

4.4 Results

Fact Verification Results We evaluated the performance of the models in predicting labels and reported the accuracy in Table 3 by different reasoning types.

As we expected, GEAR outperforms other baseline models in most of reasoning types because it used graph evidence. Especially, in existence and negation, GEAR substantially outperforms *Claim Only* baselines. Since the existence claims contain significantly less information than other types, having to search for evidence seems to increase fact verification performance. In addition, negation claims require additional inference steps compared to other types, thus logical reasoning based on graph evidence would help the model make correct prediction.

In the multi-hop setting, however, the accuracy of GEAR is lower than BERT, which may be due to the increased complexity of graph retrieval. When entities are far apart with many intermediate nodes being under-specified, it increases the probability of retrieving an incorrect graph. In GEAR, text and evidence paths are concatenated and used as input, so if many incorrect graphs are retrieved, they can lead to incorrect predictions. Also, the accuracy of BERT is the most superior in the multi-hop setting, which suggests that masked language modeling facilitates the model to robustly handle unspecified entities in the multi-hop claims.

In the *Claim Only* setting, all baselines outperform the Majority Class (51.35%), and the BERT model shows the highest performance. BlueBERT was pre-trained in the same manner, but BERT shows superior performance due to its pre-trained knowledge from Wikipedia.

Input Type	Model	W → W	W → C	C → C	C → W
Claim Only	BERT	71.75	63.85	68.10	69.43
	BlueBERT	64.76	56.28	58.77	63.92
With Evidence	GEAR	81.00	75.43	80.81	78.80

Table 4: **W** refers to written style claims and **C** refers to colloquial style claims. **W → C** means that the model is trained on the written style claim set and tested on the colloquial style claim set. Flan-T5 is not used in this experiment because we use it only in the zero-shot setting.

Cross-Style Evaluation We split the dataset into two disjoint sets, written style and colloquial style. We perform a cross-style fact verification task by using those datasets and the results are reported in Table 4.

Initially, we anticipated that using different styles for the train and test set would result in a significant decrease in verification performance. However, contradict our expectation, in **C**→**W** setting, BERT and BlueBERT show an improvement in performance over **C**→**C**. Even in GEAR, the performance score only dropped slightly. Therefore, the results demonstrate that colloquial style is constructed in various forms which can be beneficial for generalization.

5 Conclusion

In this paper, we present FACTKG, a new dataset for fact verification using knowledge graph. In order to reveal the relationship between fact verification and knowledge graph reasoning, we generated claims corresponding to a certain graph pattern. Additionally, FACTKG also includes colloquial-style claims that are applicable to the dialogue system. Our analysis showed that the claims in our dataset are difficult to solve without reasoning over the knowledge graph.

We expect the dataset to offer various research directions. One possible use of our dataset is as a benchmark for justification prediction. Most research on this task generate a text passage as justification, yet this approach lacked a gold reference. On the contrary, the interpretability of the knowl-

edge graph allows us to employ it as an explanation for the verdict, such as question answering in the medical domain where explainability is important. Furthermore, using the KG structure for the claim generation allows us to generate a dataset with more complex multi-hop reasoning by design without relying on annotator creativity.

Limitations

Since WebNLG is derived from 2015-10 version of DBpedia, FACTKG does not reflect the latest knowledge. Also, another limitation of our work is that the claims of FACTKG are constructed based on single sentences, like other crowdsourced fact verification datasets. If the claim is generated by more than one sentences, the dataset will be more challenging. We remain this challenging point as a future work.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2019-0-00075, No.2022-0-00984), and National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945), funded by the Korea government (MSIT).

References

Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. Feverous: Fact extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*.

Amazon Staff. 2018. How alexa keeps getting smarter.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020a. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020b. Generating fact checking explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online. Association for Computational Linguistics.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. Multifc: A real-world multi-domain dataset for evidence-based fact checking of claims. *arXiv preprint arXiv:1909.03242*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

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. *arXiv preprint arXiv:2011.03088*.

Ajinkya Gorakhnath Kale and Sanjika Hewavitharana. 2018. Knowledge graph construction for intelligent online personal assistant. US Patent App. 15/238,679.

Byeongchang Kim, Hyunwoo Kim, Seokhee Hong, and Gunhee Kim. 2021. How robust are fact checking systems on colloquial claims? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1535–1548, Online. Association for Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020a. Explainable automated fact-checking: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5430–5443, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Neema Kotonya and Francesca Toni. 2020b. Explainable automated fact-checking for public health claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7740–7754, Online. Association for Computational Linguistics.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21. Association for Computing Machinery.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal.

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. *CoRR*, abs/1907.11692.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Faviq: Fact verification from information-seeking questions. *arXiv preprint arXiv:2107.02153*.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from wikipedia. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France. European Language Resources Association.

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.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2021. Elastic weight consolidation for better bias inoculation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 957–964, 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.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Nancy XR Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. 2021. Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (sem-tab-facts). *arXiv preprint arXiv:2105.13995*.

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.

G. Yule and H.G. Widdowson. 1996. *Pragmatics*. Oxford Introduction to Language Study ELT. OUP Oxford.

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 Qualitative analysis

We report claims and the retrieved graphical evidence in Table A. We also report the correctness of the prediction of GEAR at the first column of our table, **Result**. We used subgraph retrieval to retrieve graph path visualize one of them. By checking the retrieved evidence, We can recognize why the model verdict the claims as refuted or supported. This shows that our graph evidence is fully interpretable.

B Relation Substitution

The four groups of compatible relations are listed in Table 6.

C Full List of Templates

C.1 Existence

The templates to generate existence claims are described in Table 7.

C.2 Factive and Non Factive Presupposition

Factive and Non Factive presupposition templates are in Table 8.

C.3 Structural Presupposition

Structural presupposition templates are in Table 9.

D Negation Labeling

D.1 Conjunction

When the negation is added to REFUTED claims, the label depends on the position of the negation. If negations are added to all parts with substituted entities, it becomes a SUPPORTED claim. Conversely, other cases preserve the label REFUTED since the negation is added to a place that is not related to entity substitution. Detailed examples are described in Table 10 and Table 11.

D.2 Multi-hop

The truth of this claim is dependent on the presence of a distinctive path that matches the claim's intent. For example, when verifying the claim in the fourth row of the Table 12, we check the existence of an entity which is connected to *'AIDAstella'* with relation *builder* and not connected to *'New York'* with relation *location*.

E Colloquial Style Claim Survey

A total of 9 graduate students participated in the survey to evaluate how much information was lost in the colloquial style claim compared to original claim. Since each person has different criteria for 'important information', the labels are divided into five rather than two. The labels are as follows, i) All facts are preserved, ii) Minor loss of information or minor grammatical errors, iii) Ambiguous whether the lost information is important, iv) It is ambiguous, but the lost information may be important, v) Loss of important information. And as a result, only 9.8% of the claims were selected as v) Loss of important information by at least two reviewers.

F Details of GEAR

To make this paper self-contained, we recall some details of the claim verification in GEAR (Zhou et al., 2019). The authors of GEAR (Zhou et al. (2019)) used sentence encoder to obtain representations for the claim and the evidence. Then they built a fully-connected evidence graph and used evidence reasoning network (ERNet) to propagate information between evidence and reason over the graph. Finally, they used an evidence aggregator to infer the final results.

Sentence Encoder

Given an input sentence, Zhou et al. (2019) employed BERT (Devlin et al., 2018) as a sentence encoder by extracting the final hidden state of the [CLS] token as the representation.

Specifically, given a claim c and N pieces of retrieved evidence $\{e_1, e_2, ..., e_N\}$, they fed each evidence-claim pair (e_i, c) into BERT to obtain the evidence representation $\mathbf{e_i}$. they also fed the claim into BERT alone to obtain the claim $\mathbf{c}$. That is,

$$\begin{aligned} \mathbf{e}_i &= \text{BERT}\left(e_i, c\right), \\ \mathbf{c} &= \text{BERT}\left(c\right). \end{aligned} \tag{1}$$

Evidence Reasoning Network

Let $\mathbf{h}^t = \{\mathbf{h}_1^t, \mathbf{h}_2^t, ..., \mathbf{h}_N^t\}$ denote the hidden states of the nodes in layer t, where $\mathbf{h}_i^t \in \mathcal{R}^{F\times 1}$ and F is the number of features in each node. The initial hidden state of each evidence node $\mathbf{h}_i^0$ was initialized by the evidence: $\mathbf{h}_i^0 = \mathbf{e}_i$. The authors proposed an Evidence Reasoning Network (ERNet) to propagate information among the evidence nodes. They first used an MLP to calculate the attention

Result	Claim	Retrieved Path
Correct	Yeah! Alfredo Zitarrosa died in a city in Uruguay I have heard that Mobyland had a successor.	(Uruguay, country, Montevideo, deathPlace, Alfredo_Zitarrosa) (Mobyland, successor, "Aero 2")
Wrong	I realized that a book was written by J. V. Jones and has the OCLC number 51969173	(J._V._Jones, author, A_Cavern_of_Black_Ice, 'oclc', "39456030")

Table 5: Examples of claims in FACTKG and retrieved graph path.

Group number	Head type	Tail type	Relation set
1	person	person	[child, children], [successor], [parent], [predecessor, precededBy], [spouse], [vicePresident, vicepresident], [primeminister, primeMinister]
2	person	team	[currentteam, currentclub, team], [debutTeam, formerTeam]
3	non-person	person	[chairperson, chairman, leader, leaderName], [manager], [founder], [director], [crewMembers], [producer], [discoverer], [creator], [editor], [writer], [coach], [starring], [dean]
4	non-person	non-person	[owningCompany, parentCompany, owner], [headquarter], [builder]

Table 6: Group information of *Relation Substitution*.

coefficients between a node i and its neighbor j ($j \in \mathcal{N}_i$),

$$p_{ij} = \mathbf{W}_1^{t-1}(\text{ReLU}(\mathbf{W}_0^{t-1}(\mathbf{h}_i^{t-1} \| \mathbf{h}_j^{t-1}))), \quad (2)$$

where $\mathcal{N}_i$ denotes the set of neighbors of node i, $\mathbf{W}_0^{t-1} \in \mathcal{R}^{H \times 2F}$ and $\mathbf{W}_1^{t-1} \in \mathcal{R}^{1 \times H}$ are weight matrices, and $\cdot \| \cdot$ denotes the concatenation operation.

Then, they normalized the coefficients using the softmax function,

$$\alpha_{ij} = \text{softmax}_j(p_{ij}) = \frac{\exp(p_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(p_{ik})}. \quad (3)$$

Finally, the normalized attention coefficients were used to compute a linear combination of the neighbor features and thus obtained the features for node i at layer t,

$$\mathbf{h}_i^t = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{h}_j^{t-1}. \quad (4)$$

The authors fed the final hidden states of the evidence nodes $\{\mathbf{h}_1^T, \mathbf{h}_2^T, ..., \mathbf{h}_N^T\}$ into their evidence aggregator to make the final inference.

Evidence Aggregator

The authors employed an evidence aggregator to gather information from different evidence nodes and obtained the final hidden state $\mathbf{o} \in \mathcal{R}^{F \times 1}$. We used the mean aggregator in GEAR.

The mean aggregator performed the *element-wise* Mean operation among hidden states.

$$\mathbf{o} = \text{Mean}(\mathbf{h}_1^T, \mathbf{h}_2^T, ..., \mathbf{h}_N^T). \quad (5)$$

Once the final state $\mathbf{o}$ is obtained, the authors employed a one-layer MLP to get the final prediction l.

$$l = \text{softmax}(\text{ReLU}(\mathbf{W}\mathbf{o} + \mathbf{b})), \quad (6)$$

where $\mathbf{W} \in \mathcal{R}^{C \times F}$ and $\mathbf{b} \in \mathcal{R}^{C \times 1}$ are parameters, and C is the number of prediction labels.

Type	Relation	Template	Example sentences
Head-Relation	successor, spouse, children, parentCompany, capital, garrison, nickname, mascot, youthclubs, predecessor, child, precededBy, religion, awards, award	{Head} had a(an) {Relation}.	Obama had a spouse.
		{Head} did not have a(an) {Relation}.	Apple did not have a parent company.
	college, university	{Head} attended {Relation}.	Obama attended university.
		{Head} did not attend {Relation}.	Obama did not attend college.
Tail-Relation	president, primeMinister, vicepresident, primeminister, vicePresident	{Tail} was a {Relation}.	Obama was a president.
		{Tail} was not a {Relation}.	Obama was not a vice president.

Table 7: Templates for Existence claims.

Presupposition type	Template	Claim Example
Factive	I forgot that {claim}.	I forgot that Obama was president.
	I realized that {claim}.	I realized that Obama was president.
	I wasn't aware that {claim}.	I wasn't aware that Obama was president.
	I didn't know that {claim}.	I didn't know that Obama was president.
	I remembered that {claim}.	I remembered that Obama was president.
	I explained that {claim}.	I explained that Obama was president.
	I emphasized that {claim}.	I emphasized that Obama was president.
	I understand that {claim}.	I understand that Obama was president.
Non Factive	I imagined that {claim}.	I imagined that Obama was president.
	I wish that {claim}.	I wish that Obama was president.
	If only {claim}.	If only Obama was president.

Table 8: Templates for factive, non factive presupposition.

Type		Relations	Template	Example claim
One-hop		leader, leaderName, mayor, senators, president, manager, generalManager, coach, chairman, dean	When was {tail} a {relaion} of {head}?	When was Elizabeth II a leader of Alderney?
		team, draftTeam, clubs, managerClub, managerclubs	When did {head} play for {tail}?	When did Aaron Boogaard play for Wichita Thunder?
		operator	When did {tail} operate {head}?	When did Aktieselskab operate Aarhus Airport?
		occupation, formerName	When was {head} a {tail}?	When was HBO a The Green Channel?
		almaMater	When did {head} graduate from the {tail}?	When did Ab Klink graduate from the Erasmus University Rotterdam?
		fossil	When was {tail} fossil found in {head}?	When was Smilodon fossil found in California?
		director	When was {head} directed by {tail}?	When was Death on a Factory Farm directed by Sarah Teale?
		producer	When was {head} produced by {tail}?	When was Turn Me On (album) produced by Wharton Tiers?
		foundation, foundedBy, founder	When was {head} founded by {tail}?	When was MotorSport Vision founded by Jonathan Palmer?
		deathCause	When did {head} die from {tail}?	When did James Craig Watson die from Peritonitis?
		creators, creator	When was {head} created by {tail}?	When was April O'Neil created by Peter Laird?
		starring	When was {head} starring {tail}?	When was Bananaman starring Graeme Garden?
		shipBuilder, builder	When was {head} built by {tail}?	When was A-Rosa Luna built by Germany?
		designer	When was {head} designed by {tail}?	When was Atatürk Monument (İzmir) designed by Pietro Canonica?
		shipCountry	When did {head} come from {tail}?	When did ARA Veinticinco de Mayo (V-2) come from Argentina?
		spouse	When was {head} married to {tail}?	When was Abraham A. Ribicoff married to Ruth Ribicoff?
		champions	When was {tail} champion at the {head}?	When was Juventus F.C. champion at the Serie A?
		recordedIn	When was {head} recorded in {tail}?	When was Bootleg Series Volume 1: The Quine Tapes recorded in San Francisco?
Existence	**Head-Relation**	successor, spouse, children, parentCompany, capital, garrison, nickname, mascot, youthclubs, predecessor, child, precededBy, religion, awards, award	What is the name of {head}'s {relation}?	What is the name of Obama's child?
		college, university	When did {head} attend {relation}?	When did Obama attend university?
	Tail-Relation	president, primeMinister, vicepresident, primeminister, vicePresident	When was {tail} {relation}?	When was Obama President?
			Where was {tail} {relation}?	Where was Biden Vice President?
			What country was {tail} {relation}?	What country was Obama President?

Table 9: Templates for structural presupposition.

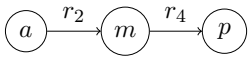
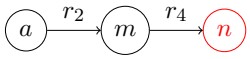
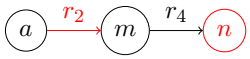
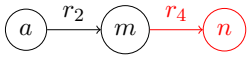
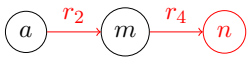
Graph	Claim Example	Label
	AIDAstella was built by Meyer Werft in Papenburg.	SUPPORTED
	AIDAstella was built by Meyer Werft in New York.	REFUTED
	AIDAstella was not built by Meyer Werft in New York.	REFUTED
	AIDAstella was built by Meyer Werft, not in New York.	SUPPORTED
	AIDAstella was not built by Meyer Werft, not in New York.	REFUTED

Table 10: r_2: shipBuilder, r_4: location, m: Meyer Werft, a: AIDAstella, n: New York, p: Papenburg.

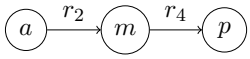
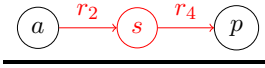
Graph	Claim Example	Label
	AIDAstella was built by Meyer Werft in Papenburg.	SUPPORTED
	AIDAstella was built by Samsung in Papenburg.	REFUTED
	AIDAstella was not built by Samsung in Papenburg.	REFUTED
	AIDAstella was built by Samsung, not in Papenburg.	REFUTED
	AIDAstella was not built by Samsung, not in Papenburg.	SUPPORTED

Table 11: r_2: shipBuilder, r_4: location, m: Meyer Werft, a: AIDAstella, p: Papenburg, s: Samsung.

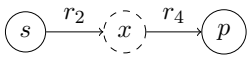
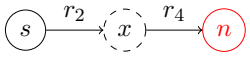
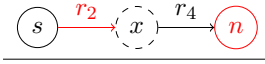
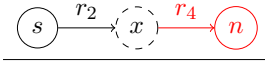
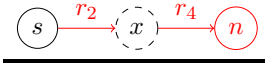
Graph	Claim Example	Label
	AIDAstella was built by a company in Papenburg.	SUPPORTED
	AIDAstella was built by a company in New York.	REFUTED
	AIDAstella was not built by a company in New York.	PATH CHECK
	AIDAstella was built by a company, not in New York.	PATH CHECK
	AIDAstella was not built by a company, not in New York.	PATH CHECK

Table 12: r_2: shipBuilder, r_4: location, s: AIDAstella, n: New York.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
after 5.conclusion, we added limitation

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
on the first page, 1. introduction

☑ A4. Have you used AI writing assistants when working on this paper?
grammarly and translation

B ☑ Did you use or create scientific artifacts?

publicly available webnlg

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

4

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

3.3

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.