Improving Evidence Retrieval on Claim Verification Pipeline through Question Enrichment

Svetlana Churina[*a], **Anab Maulana Barik**[*ab], **and Saisamarth Rajesh Phaye**[*]
[a] Centre for Trusted Internet and Community, National University of Singapore
[b] School of Computing, National University of Singapore

Abstract

The AVeriTeC shared task introduces a new real-word claim verification dataset, where a system is tasked to verify a real-world claim based on the evidence found in the internet. In this paper, we proposed a claim verification pipeline called QECV which consists of two modules, Evidence Retrieval and Claim Verification. Our pipeline collects pairs of <Question, Answer> as the evidence. Recognizing the pivotal role of question quality in the evidence efficacy, we proposed question enrichment to enhance the retrieved evidence. Specifically, we adopt three different Question Generation (QG) technique, muti-hop, single-hop, and Fact-checker style. For the claim verification module, we integrate an ensemble of multiple state-of-the-art LLM to enhance its robustness. Experiments show that QECV achieves 0.41, 0.29, and 0.42 on Q, Q+A, and AVeriTeC scores. Code is available here.

1 Introduction

Claim Verification has become critical in the past few years due to the widespread of false information. This highlight the needs for robust automated system for claim verification. To advance the research area, benchmark datasets and challenges such as FEVER (Thorne et al., 2018) and FEVEROUS (Aly et al., 2021) have been introduced and subsequent systems (Zhou et al., 2019; Liu et al., 2020; Zhong et al., 2020; Barik et al., 2022; Chen et al., 2022; Bouziane et al., 2021; Gi et al., 2021) have demonstrated progress in claim verification. Nevertheless, given the artificial claims and structured Wikipedia evidence in FEVER and FEVEROUS, those systems have been optimized primarily under this condition. Verifying real-world claim such as news claim still poses a significant challenge due to the complexity of sources, varying contexts, and the potential for misleading or evolving information.

Recently, a new claim verification benchmark on real-world called AVeriTeC (Schlichtkrull et al., 2024) was introduced. In this benchmark, the system is required to retrieve relevant document from articles across the internet and extract essential information from the articles that can debunk the claim. Then, the system must classify the claim as Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking.

Compare with previous datasets that relies on synthetic claims derived from Wikipedia, AVeriTeC focused on real-world claims. Additionally, question-answer pairs have been introduced to capture reasoning steps and include annotations for conflicting evidences, offering a more nuanced approach to claim verification.

In this dataset, question generation is a structured process aimed at deconstructing the reasoning used in fact-checking. Annotators identify key aspects of a claim that require verification by reading original claim, relevant fact-checking source(s) and original source of the claim. They have been tasked to generate questions that would help break verification into the smaller steps. These questions need to be designed to extract specific pieces of evidences that would be required to verify claim.

In this paper, we propose Question Enrichment Claim Verification (QECV) consisting of 2 modules, Evidence Retrieval and Claim Verification. To enhance the quality of the retrieved evidence, we adopt three different question generation approaches; multi-hop, single-hop, and fact-checker style. Single-hop aims to retrieve more general evidence to verify the claim, while multi-hop targets more detailed evidence for each component of the claim. Fact-checker style mimic how human fact-checker generate questions by conditioning on both the claim and the content article. In contrast, single-hop and multi-hop solely rely on the claim for

[*]All authors have contributed equally.
Correspondence emails: *churinas@nus.edu.sg*
anabmaulana@u.nus.edu

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 64–70

question generation. Our claim verification module combine two different approaches: evidence-level verifier and claim-level verifier. The former classify intermediate label to individual piece of evidence, which are subsequently aggregated to determine the claim label. Conversely, the latter directly classifies the claim label based from all the retrieved evidence. To leverage the strength of each approach, we employ a voting-based ensemble model to aggregate the output and obtain the final label. Our pipeline achieves 0.41, 0.29, and 0.42 on Q, Q+A, and AVeriTeC scores respectively, which outperforms the baseline model with a substantial margin.

2 Pipeline

As shown in Figure 1, our pipeline consists of two modules: Evidence Retrieval and Claim Verification. The input claim first passes through our three variants of evidence retrieval to retrieve relevant pairs of <Question, Answer>. Each variant generate questions from the claim and retrieve relevant articles through Faiss: a library for efficient similarity search (Douze et al., 2024). Then, it outputs list of <Question, Answer> which later combined to become the retrieved evidence. Thereafter, the claim sentence and the retrieved evidence are fed to the claim verification module to predict the final label. The detail of each module will be elaborated in subsequent subsections.

2.1 Evidence Retrieval

The evidence retrieval module processes a claim sentence through a sequential of sub-modules to extract relevant pairs of <Question, Answer> evidence.

2.1.1 Question Generation

Crafting effective questions is crucial in the question generation process, especially for claim verification. The quality of the questions can significantly influence the verification outcome, guiding it towards uncovering the truth or leading to ambiguity. Therefore, we place great importance on designing these questions carefully. Specifically, we propose three different question generation strategies: multi-hop, claim as a question, and FC-style question generation.

Multi-hop Question Generation Following QACheck methodologies (Pan et al., 2023), we employ two different question types in this strategy, initial question and follow-up question. The initial

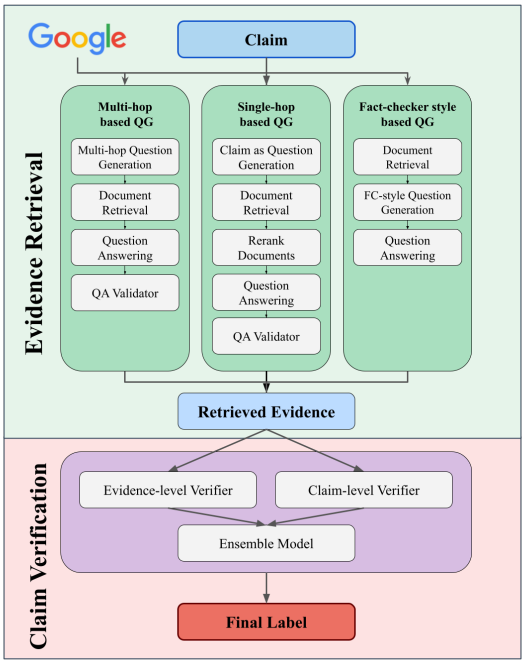

Figure 1: QECV Pipeline

question serves as the starting point for verification. Here is the prompt structure for generating initial question:

```
Claim = CLAIM
What kind of question need to be
asked to start fact checking
process?
```

Follow-up questions build on the initial question and any previous responses to further validate the claim. Here is the prompt structure for generating follow-up questions:

```
Claim = CLAIM
We already know the following:
CONTEXT = Prev. QA Pairs
Given a claim and previous
questions, what follow-up
question need to be asked to
verify the claim?
```

Claim as Question Generation Unlike the multi-hop question, this strategy leverages the entire claim as a question to better grasp the overall context and nuances of the claim. Specifically, a question "Is it true that CLAIM?" is manually constructed and subsequently paraphrased using OpenAI's GPT4o.

Fact-checker Style Question Generation

After manually reviewing the questions generated by annotators, we discovered that most of these questions are more sophisticated than those based solely on the claim. Generating such sophisticated questions requires additional knowledge, including details from the source text, information about where the claim was published, and the nature of the publishing company. Often, this information might not seem directly connected to the claim at first glance.

To generate these types of questions, we need to provide more comprehensive information to the model and tailor the question generation process accordingly.

Here is the prompt structure for generating fact-checker style questions:

```
Claim = CLAIM

Article text = TEXT Is this
article relevant to our claim?
If yes - what question need to
be asked based on the article
text that will be required to
verify claim?
```

By systematically asking well-structured questions, our system aims to facilitate a thorough and accurate verification process.

2.1.2 Document Retrieval

This module accepts a question as input to extract relevant documents. We leverage the provided document collections from the dataset provided in the challenge. However, given the substantial proportion of empty documents (exceeding 50%) within these collections, we augmented more documents by querying the claim itself with Google API. We also scraped a few hundred URLs manually for which document-text field was empty.

To match any question with the corresponding documents, we tried multiple techniques. In summary, we create an embedding vector for each document and also the question, using the Sentence Transformer library (Reimers and Gurevych, 2019). Considering the resource constraints, we used "all-MiniLM-L6-v2" model to get the encodings. We found that Faiss yields fast indexing and best similarity results even for extremely long texts, partly due to the quality of encodings by Sentence Transformers. We get the 20 best matches with the question and pass it to the Reranking module which is described below.

2.1.3 Rerank Documents

In our manual analysis, we noticed that some of the URLs could be from inauthentic sources, and could include wrong information. However, the gold labeled URLs in training data seemed to have authentic information. To leverage this, we devise a simple reranking algorithm, based on the training data's Gold standard websites (retrieved from the URLs). We calculate the frequency based weighting for the training data's ground truth websites which are of type "gold", and also for the rest, which we call "normal" website weight. Now, for the test stage, we check every URL's Faiss score, and multiply it with the corresponding website weight. Gold websites are always prioritised above the normal weighted websites. This reranking multiplication considers only the top 20 documents and not all, because considering all URLs could result in dissimilar documents being at the top.

Post-reranking, we take the top 5 documents retrieved and pass them to the Question Answering stage, which is described below. This reranking stage yielded us best results for Claim-as-question generation. However, it didn't yield significantly better results for the Multi-hop based QG. By adding URL weightings (and using no claim-as-questions yet) on the development dataset, our Q and Q+A score slightly go down from 31.35 and 21.67 to 30.64 and 20.32 respectively. Our hypothesis for this observation is that, multiple questions retrieve multiple documents. As a result, those retrieved documents already cover a number of authentic websites. Hence, URL weighting might hinder more than help in multi-hop stage.

2.1.4 Question Answering

Once we retrieve the five most relevant documents, the first step is to generate a summary tailored to the question at hand. For summary generation, we utilize OpenAI's GPT4o, providing it with the question, the claim, and the text of the document as input.

Since the summary is generated with the specific goal of addressing the question based on the document's content, it is subsequently treated as the answer in the following modules. The prompt used for generating the summary is as follows:

```
Claim = CLAIM

Question= QUESTION

Text = TEXT
```

```
Provide a brief summary of the
text, focusing on information
relevant to the question. The
summary should aim at answering
the question.
```

2.1.5 QA Validator

The QA Validator module plays a crucial role in our fact-checking system, as it determines the direction of subsequent verification processes. Given that some questions may yield conflicting answers (which could lead to cherry-picking the final label), it is essential to determine differences in answers before proceeding. To address this, we assign individual labels to each QA pair based on their content.

Each QA pair can be assigned one of three labels: *Supported*, *Refuted*, or *Not Enough Evidence*. Once each QA pair is labeled, we group them based on these three categories. The logic for handling the labels is as follows:

- If a question has both *Supported* and *Not Enough Evidence* pairs, we only consider the *Supported* pairs.
- If a question has both *Refuted* and *Not Enough Evidence* pairs, we only consider the *Refuted* pairs.
- If a question has both *Supported* and *Refuted* pairs, we retain both and generate follow-up questions based on these two paths.
- If a question only has *Not Enough Evidence* pairs, we proceed with that label.

After selecting the pairs to continue with, we must choose the best QA pair within each category. Using OpenAI's GPT4o, we analyze each QA pair and select the one that provides the most informative response to the question.

2.2 Claim Verification

The claim verification module is given a claim sentence and evidence as input, it tasked to classify the label of the claim. The module is a combination of two claim verification system variants, namely Evidence-level verifier and Claim-level verifier.

Evidence-level Verifier In this variant, the model was trained to independently classify the label of a claim w.r.t a piece of evidence. The evidence is a concatenation of a question and an answer following this format: *"Question: [Question]. Answer: [Answer]"*. Claims are classified as Supported, Refuted, or Not Enough Evidence, constituting a fine-grained label. Ultimately, the claim label was determined through applying deterministic function to the fine-grained labels:

- **Supported**: If all the fine-grained labels are Supported.
- **Refuted**: If all the fine-grained labels are Refuted.
- **Conflicting Evidence/Cherrypicking**: If both Supported and Refuted are presents in the fine-grained labels
- **Not Enough Evidence**: Otherwise

Claim-level Verifier In this variant, we follow a conventional claim verification model, in which, the model is tasked to classify the label of the claim given all pieces of evidence. The evidence is the concatenation of questions and answers following this format: Question-1: *[Question-1]. Answer-1: [Answer-1]. ... Question-N: [Question-N]. Answer-N: [Answer-N]*. The claim is classified either Supported, Refuted, Not Enough Evidence, or Conflicting Evidence/Cherrypicking.

For each variant, we experimented with different LLMs as the backbone and we combine the output of these models through a voting-based ensemble model to obtain the final claim label. A comprehensive description of each LLM is presented in the next section.

2.2.1 Training Detail

We fine-tuned five LLMs: (1) flan-t5-Large (Chung et al., 2024), (2) Mistral-7B-Instruct-v0.1 (Jiang et al., 2023), (3) Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024), (4) gpt-3.5-turbo-0125, and (5) gpt-4o-mini. For T5, Mistral, and Mixtral, we set the learning rate to $1e-4$ and fine-tuned it for 2 epochs. We use LoRA with rank, alpha, and dropout are set to 8, 32, and 0.05. Meanwhile, for GPT3.5 and GPT4, we use 4 epochs. We set the other hyperparameters as default.

Evidence-level Verifier: to obtain the training data for this variants, we first filter out all claims with label $Conflicting\ Evidence/Cherrypicking$. Then, quadruplets of <claim, question, answer, label> are obtained from the training set. For **Claim-level Verifier**, we collect quadruplets of <claim, list of

Model	Q	Q+A
baseline	0.24	0.19
Single	0.23	0.16
Single+Multi	0.39	0.27
Single+Multi+FC	**0.44**	**0.31**

Table 1: Evidence Retrieval Result on Development Set. Comparison of results from different question generation types.

questions, list of answers, label> from the whole training set.

We employ majority voting for the ensemble models. Based from the experiments on the dev set, our final claim verification is an ensemble of 4 different models: GPT4 on Evidence-level verifier and Mistral, GPT3.5, and GPT4 on Claim-level verifier.

3 Results

3.1 Evidence Retrieval

Table 1 reports the results evidence retrieval performance of QECV compared to the baseline models on the development set. Among the investigated Question Generation style, the single-hop approaches yield the lowest score among other variants. This shows that claim as question is not sufficient to retrieve enough evidence to verify the claim. Nevertheless, the claim as question is competitive with the baseline models. Augment the evidence through multi-hop question led to a substantial improvement, which improves 0.13 on Q and 0.11 on Q+A. This suggest that Q+A effectively capture more detailed and relevant evidence. Finally, adding FC-style question improve additional performance gain by 0.5 on Q and 0.4 on Q+A, emphasizing the efficacy of this approach to collect evidence that are hardly mention by the claim.

3.2 Claim Verification

Table 2 reports the Evidence-level Verifier, and Table 3 reports the Claim-level Verifier on the development set using various fine-tuned LLM.

Effect on LLMs size: Through the experiments, we can see that on evidence-level verifier, bigger model such as mixtral, GPT3.5, and GPT4 outperforms smaller models on AVeriTeC score. Meanwhile on claim-level verifier, mistral, GPT3.5 and GPT4 outperforms smaller models on AVeriTeC score. Moreover, GPT3.5 and GPT4 are consistently achieved the highest performance across both variants.

Effect on Different Variants: Experimental results demonstrate that claim-level verifier are superior than the evidence-level verifier, both in macro F1 and AVeriTeC score. The under performance of evidence-level is attributed to the deterministic function. For instance, for a "Supported" claim "*Amy Coney Barrett was confirmed as US Supreme Court Justice on October 26, 2020.*", our evidence retrieval retrieves 7 evidence and the evidence-level verifier predicts 6 out of the 7 evidence as "Supported". The last evidence stated that "*The summarized information does not provide the exact date of Amy Coney Barrett's confirmation to the US Supreme Court. It only states that she has been confirmed.*", which the verifier predicts as Not Enough Evidence. Finally, the final claim label is Not Enough Evidence due to the deterministic function. Nevertheless, evidence-level verifier is superior in identifying Not Enough Evidence label, achieving 0.28 F1 score compared to 0.16 F1 score for claim-level verifier.

Impact of using different LLMs: Experimental results indicate that different models exhibit varying strength. In claim-level verifier, GPT3.5 and GPT4 are superior on Supported and Refuted labels, whereas Mistral and Mixtral excel on Not Enough Evidence and Conflicting labels. Conversely, in the evidence-level verifier, GPT3.5 and GPT4 are the most effective on Not Enough Evidence and Conflicting labels, meanwhile Mixtral excels on Refuted and BART on Supported. This suggest that each LLM possesses it's own strength depending on the verifier variant. Consequently, combining the strength of these models across different variants can enhance the robustness of the verifier.

3.3 Full Pipeline

For our final pipeline, we use the best performance for the evidence retrieval, which is a combination Single+Multi+FC-style based QG. For the claim verification, we ensemble GPT4 on evidence-level verifier and Mistral, GPT3.5, and GPT4 on claim-level verifier to gain benefit the strength of different variants. Table 4 indicates that our final pipeline significantly outperforms the baseline on every metrics, by 0.17 on Q, 0.10 in Q+A, and 0.31 in AVeriTeC score.

Model	AVeriTeC	F1				
		Sup	Ref	Nee	Conf	Macro
baseline (BART_large)	0.09	**0.43**	0.71	0.00	0.09	0.32
T5	0.33	0.28	0.78	0.27	**0.14**	0.36
Mistral	0.32	0.19	0.78	0.24	0.10	0.33
Mixtral	0.36	0.33	**0.81**	0.16	0.09	0.35
GPT3.5	0.35	0.24	0.79	**0.28**	0.13	0.36
GPT4	**0.37**	0.40	0.80	0.22	**0.14**	**0.39**

Table 2: Evidence-level verifier results on the development set. "Sup" denotes "Supported," "Ref" stands for "Refuted," "Nee" represents "Not Enough Information," and "Conf" corresponds to "Conflicting" or "Cherrypicking" label types.

Model	AVeriTeC	F1				
		Sup	Ref	Nee	Conf	Macro
T5	0.39	0.42	0.79	0.11	0.14	0.37
Mistral	0.44	**0.61**	0.82	0.09	**0.20**	**0.43**
Mixtral	0.38	0.46	0.82	**0.16**	0.16	0.40
GPT3.5	**0.46**	**0.61**	**0.84**	0.12	0.16	**0.43**
GPT4	0.44	0.59	**0.84**	0.08	0.18	0.42

Table 3: Claim-level Verifier Result on Development Set, where "Sup" - Supported, "Ref" - Refuted, "Nee" - Not Enough Information, "Conf" - Conflicting/Cherrypicking type of labels.

	Development Set			Test Set		
Model	Q	Q+A	AVeriTeC	Q	Q+A	AVeriTeC
baseline	0.24	0.19	0.09	0.24	0.20	0.11
ours	**0.44**	**0.31**	**0.46**	**0.41**	**0.30**	**0.42**

Table 4: Result on Full Pipeline compare with baseline results, where "Q" - question-based retrieval performance, "Q+A" - question + answer retrieval performance

4 Conclusion

In this paper, we introduced the QECV, a pipeline for verifying real-world claims. Improving the evidence retrieval through question enrichment enable the framework to cover more evidence for verifying the claim, thus achieves 0.41 and 0.30 for the Q and Q+A performance on the test set. Additionally, our pipeline combines across various claim verifier variants and LLMs to leverage their unique strengths, resulting in more robust verification process and an 0.42 AVeriTeC score on the test set.

5 Limitations

We believe one of the major limitations of this pipeline is relevance of documents we retrieve for each question. We have tried to address this by introducing multi-hop QG, claim-as-question module, and emphasising fact-checking styled documents. However, there is definitely scope of further improvement here.

Despite the ability of our question enrichment methods on the evidence retrieval, the hallucination remains, particularly in the question answering stage. Moreover, our claim verification models rely solely on the ground truth data for training. Given that the previous works demonstrate the effectiveness of adding noise for claim verification on synthethic claim, it is worthwhile to investigate whether a similar approach can be applied to the real-world claims.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information.

Anab Maulana Barik, Wynne Hsu, and Mong Li Lee. 2022. Incorporating external knowledge for evidence-based fact verification. In *Companion Pro-*

ceedings of the Web Conference 2022, pages 429–437.

Mostafa Bouziane, Hugo Perrin, Amine Sadeq, Thanh Nguyen, Aurélien Cluzeau, and Julien Mardas. 2021. Fabulous: fact-checking based on understanding of language over unstructured and structured information. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 31–39.

Jiangjie Chen, Qiaoben Bao, Changzhi Sun, Xinbo Zhang, Jiaze Chen, Hao Zhou, Yanghua Xiao, and Lei Li. 2022. Loren: Logic-regularized reasoning for interpretable fact verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10482–10491.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

In-Zu Gi, Ting-Yu Fang, and Richard Tzong-Han Tsai. 2021. Verdict inference with claim and retrieved elements using roberta. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 60–65.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

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.

Liangming Pan, Xinyuan Lu, Min-Yen Kan, and Preslav Nakov. 2023. Qacheck: A demonstration system for question-guided multi-hop fact-checking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 264–273.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2024. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36.

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.

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.

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.