# Exploring Retrieval Augmented Generation For Real-world Claim Verification

**Omar Adjali**
Université Paris-Saclay
Gif-sur-Yvette, France

## Abstract

Automated Fact-Checking (AFC) has recently gained considerable attention to address the increasing misinformation spreading in the web and social media. The recently introduced AVeriTeC dataset alleviates some limitations of existing AFC benchmarks. In this paper, we propose to explore Retrieval Augmented Generation (RAG) and describe the system (UPS participant) we implemented to solve the AVeriTeC shared task. Our end-to-end system integrates retrieval and generation in a joint training setup to enhance evidence retrieval and question generation. Our system operates as follows: First, we conduct dense retrieval of evidence by encoding candidate evidence sentences from the provided knowledge store documents. Next, we perform a secondary retrieval of question-answer pairs from the training set, encoding these into dense vectors to support question generation with relevant in-context examples. During training, the question generator is optimized to generate questions based on retrieved or gold evidence. In preliminary automatic evaluation, our system achieved respectively 0.198 and 0.210 AVeriTeC scores on the dev and test sets.

## 1 Introduction

With the unprecedented growing of fake news in the web and on social media, several research efforts have been supported in the recent years to combat online misinformation. While manual fact-checking is the most reliable method for verifying information, the large-scale amount of daily published and shared content has made the development of automated fact-checking solutions crucial to assist in the manual fact checking process. Following such initiatives, the recently introduced AVeriTeC (Automated VERIfication of TExtual Claims) dataset (Schlichtkrull et al., 2024) contributes to address the aforementioned challenges, and serves as a benchmark for the AVeriTeC shared task. In this paper, we report our findings in addressing the AVeriTeC shared task and describe the proposed system which is evaluated on its ability of verifying real-world claims with evidence from the Web. In contrast to other fact-checking datasets such as FEVER (Thorne et al., 2018), VITAMINC (Schuster et al., 2021) and FEVEROUS (Aly et al., 2021), AVeriTeC focuses on realistic scenarios where real-world claims are derived from the web rather than Wikipedia. In this context, systems are required to retrieve evidence that either supports or refutes a given claim, using sources from either the Web or a document collection scrapped from the web and provided by the organizers. Based on this evidence, systems must categorize the claim as *Supported*, *Refuted*, *Not Enough Evidence* (when there is insufficient evidence to make a determination), or *Conflicting Evidence/Cherry-picking* (when both supporting and refuting evidence are present). A response is considered correct only if it includes both the accurate label and sufficient supporting evidence. Due to the complexity of evaluating evidence retrieval automatically, a manual evaluation process will be conducted to ensure a fair assessment of the participant systems.

## 2 AVeriTeC baseline

The AVeriTeC shared task organizers proposed a pipeline system which comprises the following steps: 1) Given a claim c, it is used as a query input of a search engine (Google API) to obtain relevant URLs which are parsed into sentences. The collection of sentences serves for evidence retrieval. 2) For each claim c, only the top 100 sentences $\{s_i\}_{i=1}^{100}$ are kept based on the BM25 similarity between each $s_i$ and c. 3) For each of the top 100 sentence $s_i$, BLOOM (Le Scao et al., 2023) allows to generate QA pairs which are used as evidence for veracity prediction. To allow more in-context examples for QA pairs generation, the 10 closest



*Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 113–117

claim-QA pairs are retrieved from the training set using the BM25 similarity between $s_i$ and each answer included in a claim-QA pair of the training set. 4) The top 3 generated QA pairs are kept as evidence using a pre-trained BERT-based re-ranker (Devlin et al., 2019). 5) Finally, a claim c and its 3 generated QA evidence pairs are input in another pre-trained BERT model to predict the veracity label.

## 3 Proposed system

Following the baseline pipeline, we propose a simpler end-to-end integrated system ( see Figure 1) which relies on the Retrieval Augmented Generation (RAG) framework to solve the AVeriTeC challenge where retrieval and generation complement each other using joint training. At the first stage, we perform evidence dense retrieval after encoding all potential evidence sentences retrieved from the provided knowledge store documents. Then, we perform a second retrieval of question-answer pairs from the training set (encoder into dense vectors) to support question generation with in-context examples. During training, the question generator learns to generate question given retrieved/gold evidence by jointly updating both generator and evidence/answer encoder using the RAG loss (Lewis et al., 2020). Finally, a veracity prediction model is employed to label the retrieved evidence.

### 3.1 Evidence retrieval

Using the searched documents provided by the search engine, we similarly keep the top 100 sentences as potential evidence using BM25. We then encode each sentence $s_i$ into dense vector representations using a Bert-base encoder $\mathbf{E_s}(\cdot)$. We represent each sentence using the 768-d pooled vector of the `[CLS]` special token. Given a dataset $\mathcal{D}$ of $N$ claims, instead of encoding all sentences into a $(N \times 100 \times 768)$ matrix, we rather encode the top 100 potential sentence evidence of each claim $c_i \in \mathcal{D}$ into one $(N \times 100 \times 768)$ matrix. This allows to reduce the search space during evidence retrieval since the relevant evidence sentences of claim $c_i$ are likely to be found in its corresponding top-100 retrieved sentence set. Thus, we build $N$ Faiss indexes (Johnson et al., 2019) for each $c_i \in \mathcal{D}$ where each of them, maps evidence sentences to dense vectors. These enable us to perform fast exact maximum inner product search (MIPS). Formally, given a claim $c_i$, and its top-100 evidence sentence set $S_i = \{s_j\}_{j=1}^{100}$, we compute the inner product between its dense vectors and all $s_j \in S_i$ as follows :

$$s(c_i, s_j) = \mathbf{E_s}(c_i)\mathbf{E_s}(s_j) \quad (1)$$

In this way, given an input claim $c_i$, we retrieve the top-K most relevant sentence using the highest relevance scores $s(\cdot)$.

### 3.2 In-context QA pairs retrieval

Similar to (Schlichtkrull et al., 2024), in order to provide the generator in-context examples for question-answer pair generation, we retrieve the top $L$ QA-pairs from the training set which serve for building the final prompt. Given a retrieved sentence $s_i$ obtained after the first step, we encode it using the same pre-train BERT-base encoder $\mathbf{E_s}(\cdot)$. Similar to the baseline system, the top $L$ QA-pairs are selected according to the semantic similarity between answers in the QA pair training set and the retrieved evidence sentences. We therefore perform maximum inner product search for each sentence $s_i$ after encoding and indexing all the answers in the training set as follows:

$$s(s_i, a_j) = \mathbf{E_s}(s_i)\mathbf{E_s}(a_j) \quad (2)$$

Similar to the sentence retrieval stage, we select the top-L QA pairs whose answers achieve the highest retrieval scores.

### 3.3 Question generation

In this step, given a claim $c_i$, we generate a question for each sentence retrieved in the first stage. Note that the top-L retrieved QA pairs (in-context examples) are used in the same way as in (Schlichtkrull et al., 2024) to build the prompt. Given a generated question $q_i$ and a retrieved sentence $s_i$, we consider ($q_i$, $s_i$) as a QA evidence pair for $c_i$.

### 3.4 Veracity prediction

Given a claim $c_i$, its top-K QA generated pairs as evidence, we followed the baseline system to predict the veracity label which relies on a pre-trained BERT sequence classification model.

### 3.5 Training and inference

During training, given a claim $c_i$, we use its ground-truth QA evidence pairs provided in the training set to build the question generation prompt as well as generation labels. More precisely, given a set of ground truth QA pairs, we use the question of the

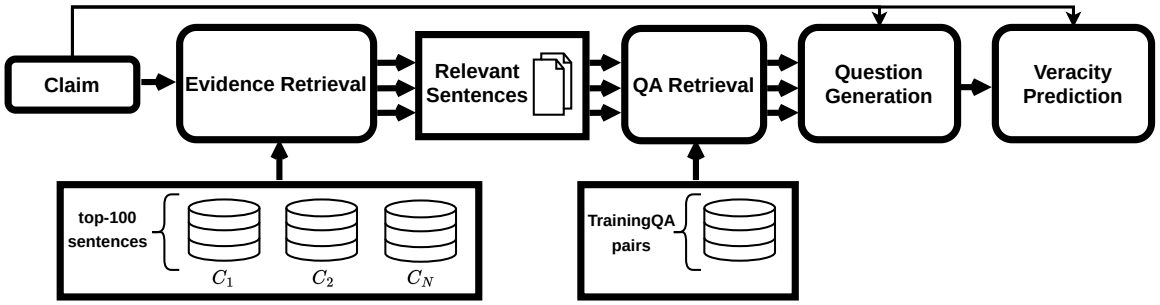


Figure 1: Our proposed pipeline system overview.

first QA pair as the generation target while the remaining pairs are used as in-context QA examples to build the final prompt. Experiments showed that using ground-truth QA pairs to build the prompt during training showed better performance than using retrieved ones. Thus, evidence retrieval and in-context QA pairs retrieval are only performed at inference time. In this setting, the sentence encoder and the question generator are jointly trained on the following RAG loss (Lewis et al., 2020):

$$\mathcal{L}_{\text{RAG}} = -\sum_{i=1}^{N} (\log(s(c_i, a^*) \cdot p_\Phi(q^*|pt(c_i), a^*))) \quad (3)$$

where $N$ is the number of claims in the dataset, $q^*$ is the ground truth question, $a^*$ is the ground truth answer and $p_\Phi(q^*|pt(c_i), a^*)$ is the probability distribution of generating the question $q^*$ given the built prompt $pt(c_i)$ and $a^*$, and $\Phi$ is the generator's parameters. $s(c_i, a^*)$ is the similarity score between the claim $c_i$ and the ground truth answer. This learning objective allows to condition the generated questions on the retrieved evidence since the gradients are propagated through both the sentence encoder and the generator. At inference time, more relevant evidence sentences are expected to be retrieved thanks to the generator feedback signals during training while improved retrieval will contribute to generate more accurate questions.

## 4 Experiments

### 4.1 Evaluation

Systems are evaluated on their ability to retrieve evidence and to predict veracity labels. Note that veracity predictions are considered correct only when correct evidence has been found. The Hungarian METEOR metric (Schlichtkrull et al., 2024) is used to score retrieved questions and retrieved questions + answers. Furthermore, systems are ranked according to the Averitec score (METEOR) conditioned on correct evidence retrieved at a cut-off value of 0.25.

### 4.2 Implementation details

We initialized the pre-trained BERT-base model used for evidence retrieval and in-context QA retrieval with an answer encoder trained on TriviaQA (Joshi et al., 2017). For question generation, we experiment with the T5-large (738M parameters) (Raffel et al., 2020) pre-trained generator. The batch size is set to 2 due to GPU memory limit. We trained our system using a 2e-5 learning rate for 20 epochs. At inference time, we decode using beam-search with 2 beams. We selected the model checkpoints based on the validation performance. All experiments needed only one Nvidia A100 (80G) GPU. Our implementation is based on PyTorch (Paszke et al., 2017). Pretrained models are obtained using Hugginface and Transformers (Wolf et al., 2020). The Faiss library (Johnson et al., 2019) is used for MIPS search and vector indexing.

## 5 Results

Table 1 reports the performance results of our approach and baseline systems evaluated on the AVeriTeC shared task for the dev and test splits. Models are evaluated based on their ability to: 1) retrieve evidence in two settings: Question only (Q only), Question and Answer (Q+A). 2) Verifying veracity of claims using the AVeriTeC score for different cutoff values. Overall, our system with 955M parameters (BERT encoder + T5-large) significantly outperforms the AVeriTeC-BLOOM-7b baseline on both evidence retrieval and veracity checking across all the metrics suggesting that LLM's parametric memory is not sufficient to solve knowledge-intensive tasks such as fact-checking.

| Model | split | Q only | Q+A | Veracity@0.2 | Veracity@0.25 | Veracity@0.3 |
|---|---|---|---|---|---|---|
| AVeriTeC-BLOOM-7b | dev | 0.240 | 0.185 | 0.186 | 0.092 | 0.050 |
| AVeriTeC-BLOOM-7b | test | 0.248 | 0.185 | 0.176 | 0.109 | 0.059 |
| Ours (UPS) | dev | 0.280 | 0.250 | 0.280 | 0.198 | 0.092 |
| Ours (UPS) | test | 0.310 | 0.270 | - | 0.210 | - |

Table 1: Averitec shared task results

| Claim Type | Veracity score |
|---|---|
| Event/Property Claim | 0.131 |
| Position Statement | 0.168 |
| Causal Claim | 0.118 |
| Numerical Claim | 0.144 |
| Quote Verification | 0.123 |

Table 2: Averitec scores by type @0.25 of our best performing system for dev set.

| Veracity Label | F1 |
|---|---|
| Supported | 0.292 |
| Refuted | 0.653 |
| Not Enough Evidence | 0.160 |
| Conflicting Evidence/Cherrypicking | 0.166 |

Table 3: Veracity prediction dev set F1 results for each veracity label.

At inference time, we achieved the best performance with the number of retrieved evidence K=10, while higher values decreases both evidence retrieval and veracity checking. Regarding the number of retrieved in-context examples L, we found that building the prompt using only L=3 is sufficient for the question generation model to reach our best performing system. We assume that our BERT-base retrieval provides more useful in-context examples in the top retrieved QA pairs and does not need to re-rank evidence compared to the baseline model which relies on BM25 to retrieve evidence. Indeed, while we do not perform evidence retrieval during training, we still update the BERT retrieval encoder parameters using the claim-evidence similarity scores with the RAG loss. This latter allows to learn retrieving more relevant evidence for the target question using the feedback from the question generator.

We reports in Table 2 the veracity scores of our best performing system for each claim type. We note that there is no substantial performance gap between claim types, even if our system struggles more with *causal* and *Quote Verification* claims. Analysing these results need more investigations in future work.

Table 3 shows the F1 scores for each veracity label. We employed the provided checkpoint for veracity prediction which failed to predict the *Conflicting Evidence/Cherrypicking* label even with gold evidence (Schlichtkrull et al., 2024). Veracity prediction performs better on this label using our system however predictions are worse for the *Supported* label which suggests that improving evidence retrieval plays an important role to achieve the best fact-checking performance.

## 6 Conclusion

We presented in this paper our participant system (UPS) at the AVeriTeC shared task on verifying real-world claims with evidence from the Web. In preliminary automatic evaluation, our system achieved respectively 0.198 and 0.210 AVeriTeC scores on the dev and test sets, and was ranked 13 out of 23 participant teams. In terms of limitations, our proposed system relies solely on the AVeriTeC training set which is relatively small size. We believe that our RAG approach would benefit from more training data. Moreover, experimenting with larger generator models may improve the quality of generated questions and thus the overall fact-checking performance.

## 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 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, Long Beach, CA, USA. MIT Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin c! robust fact verification with contrastive evidence. *arXiv preprint arXiv:2103.08541*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.