Dunamu-ml's Submissions on AVERITEC Shared Task

Heesoo Park, Dongjun Lee, Jaehyuk Kim, ChoongWon Park, Changhwa Park
Dunamu Inc.
{belle, tonny, loki, elvie, dexter}@dunamu.com

Abstract

This paper presents the Dunamu-ml's submission to the AVERITEC shared task of the 7th the Fact Extraction and VERification (FEVER) workshop. The task focused on discriminating whether each claim is a fact or not. Our method is powered by the combination of an LLM and a non-parametric lexicon-based method (i.e. BM25). Essentially, we augmented the list of evidences containing the query and the corresponding answers using an powerful LLM, then, retrieved the relative documents using the generated evidences. As such, our method made a great improvement over the baseline results, achieving 0.33 performance gain over the baseline in AveriTec score.

1 Introduction

The rise in misinformation has led to a greater need for fact-checking, which involves determining the accuracy of a claim through evidence. Consequently, research on methods that automatically detect whether specific claims are true or false is being conducted actively. (Vlachos and Riedel, 2014; Thorne et al., 2018a) As part of this effort, the shared task Fact Extraction and VERification (FEVER) [1] is held regularly (Thorne et al., 2018b, 2019; Wang et al., 2021; Aly et al., 2021).

Fact-checking requires large-scale retrieval. Large-scale retrieval involves retrieving the most relevant documents from a vast collection containing millions to billions of entries in response to a text query. Over the past ten years, deep representation learning techniques have become essential for large-scale retrieval, transitioning from traditional Bag-of-Words (BoW) (Mikolov et al., 2013) methods to Pre-trained Language Models (PLMs) (Devlin et al., 2019). The latest advancements in LLMs offer a quicker path to achieve zero-shot retrieval by enhancing a query with potential answers obtained from the LLMs (Gao et al., 2023).

In this paper, we introduce our approach to the FEVER 2024 Share Task named AveriTeC shared tasks (Schlichtkrull et al., 2023). We aim to build our model powered by the generation and retrieval ability of recent LLMs (Achiam et al., 2023). Our method is inspired by (Shen et al., 2023) which utilize a non-parametric lexicon-based method (such as BM25 (Robertson et al., 2009)) as the retrieval component to directly measure the similarity between the query and document and boost the query using powerful LLM.

First, we generated initial question and answer pairs without any documents retrieved. Then, we retrieved relevant documents and fix the initial answers using it. Finally, we infer the final answer using the given evidences. Our approach significantly enhanced the baseline outcomes, securing a 0.33 increase in performance compared to the baseline according to the AveriTec score. For evaluation, we used the given system[2].

2 Task Description

The AVeriTeC challenge (Schlichtkrull et al., 2023) aims to evaluate the ability of systems to verify real-world claims with evidence from the Web.

- The systems need to find evidence that either supports or contradicts a claim, based on the claim itself and its accompanying metadata. This evidence can be sourced from the Web or from the collection of documents provided by the organizers.

- Based on the evidence gathered, classify the claim as either `Supported` or `Refuted`, or categorize it as `Not Enough Evidence` if there is insufficient evidence to make a determination. If the evidence presents conflicting view-

[1] https://fever.ai/index.html

[2] https://eval.ai/web/challenges/challenge-page/2285/overview

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 71–76

points or appears selective, label the claim as `Conflicting Evidence/Cherry-picking`.

- For a response to be deemed accurate, both the label assigned and the quality of evidence provided must be correct. Since evaluating evidence retrieval can be challenging to automate, participants will be requested to assist in manually evaluating it to ensure a fair assessment of the systems.

The output format of each claim should be:

- *claim_id*: The ID of the sample.
- *claim*: The claim text itself.
- *pred_label*: The predicted label of the claim.
- *evidence*: A list of QA pairs. Each set consists of dictionaries with four fields.
  - *question*: The text of the generated question.
  - *answer*: The text of the answer of the generated question.
  - *url*: The source url for the answer.
  - *scraped_text*: The text scraped from the url.

2.1 AVERITEC Corpus

The AVeriTeC dataset, as described in the study by (Schlichtkrull et al., 2023), comprises 4,568 examples sourced from 50 fact-checking organizations using the Google FactCheck Claim Search API[3], which is built on ClaimReview[4]. AVeriTeC is distinguished as the initial AFC dataset to offer question-answer decomposition along with justifications, while also addressing challenges related to context dependence, evidence insufficiency, and temporal leaks. Additional details about AVeriTeC can be found on the project's GitHub repository: https://github.com/MichSchli/AVeriTeC.

2.2 Evaluation metric

The AVeriTeC score is based on adjustments made to the FEVER scorer (Thorne et al., 2018a). While FEVER relies on a closed evidence source such as Wikipedia, AVERITEC is tailored to handle evidence sourced from the open web. Since identical evidence may be found across multiple sources, precise matching for scoring retrieved evidence is impractical. Hence, AVERITEC utilizes approximate matching and utilizes the Hungarian Algorithm to determine the most suitable match between the provided evidence and the annotated evidence.

$$u_f(\widehat{Y}, Y) = \frac{1}{|Y|} \max \sum_{\hat{y} \in \hat{Y}} \sum_{y \in Y} f(\hat{y}, y) X(\hat{y}, y) \quad (1)$$

During the evaluation process, the system employed the METEOR (Banerjee and Lavie, 2005) implementation from NLTK (Bird et al., 2009) as the scoring function f, known for its strong correlation with human assessments of similarity (Fomicheva and Specia, 2019). They do not utilize a precision metric to prevent penalizing systems for posing extra relevant information-seeking questions. Nevertheless, all systems are constrained to a maximum of $k = 10$ question-answer pairs. We assess the accuracy of truthfulness predictions and supporting evidence by applying a threshold of $f(\hat{y}, y) >= \lambda$ to ascertain the retrieval of accurate evidence (using combined questions and answers). Claims with lower evidence scores are assigned veracity and justification scores of 0.

3 System Overview

In this section, we firstly provide a brief description of how we pre-processed the given knowledge store and present our approach to the task.

3.1 Data crawling and preprocessing

As we mentioned in Section 2.1, the pre-googled knowledge store, which includes web urls and their scraped text for each claim, is provided by the organizers. However, in the case that the url corresponds to either a YouTube video or a PDF document, the scraped text field is left blank, even if it includes crucial evidence for verifying the claim. To address this, we extract the transcripts from YouTube videos and parse the text from PDF documents, subsequently saving them in the data store. In addition, we segment all the documents into segments comprising 10 sentences each, not containing an excessive amount of information.

3.2 Model configuration

Our approach to the task consists of three steps, as depicted in Figure 1.

Step 1: Generate initial question and answer pairs without any documents retrieved. In order to verify the veracity of claims, it is essential to

[3] https://toolbox.google.com/factcheck/apis
[4] https://www.claimreviewproject.com/

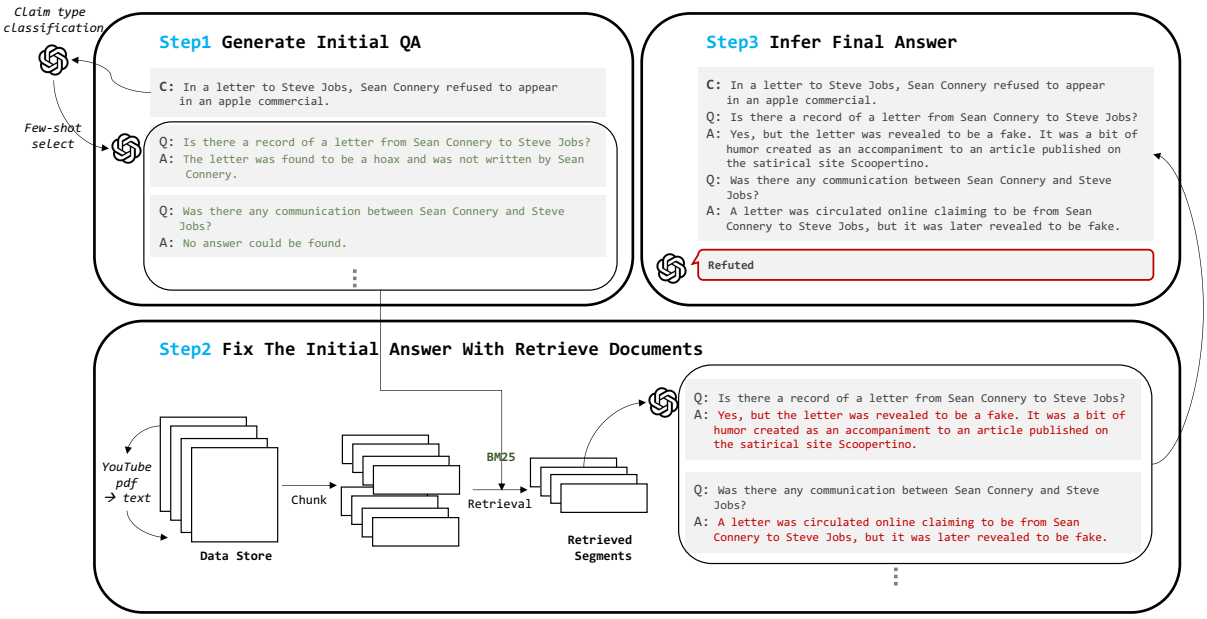

Figure 1: A diagram illustrating the three steps of our method for AVERITEC shared task. The text generated by GPT-4 is in green in Step 1 and in red in Step2. In Step 3, the predicted answer by GPT-4 is enclosed in a red box.

formulate questions that can be answered based on reliable documents retrieved from the knowledge store. Research has shown that utilizing artificially generated answers in the search, as opposed to using the questions alone, can enhance document retrieval performance. (Gao et al., 2023) As a result, a decision has been made to concurrently generate both questions and answers for use in the search process. This approach aims to improve the efficiency and effectiveness of information retrieval for fact-checking purposes.

Initially, we categorize each claim using GPT-4 with few shots which consist of pairs of (1) each claim and (2) its corresponding claim type. We classify each claim using the following prompt:

```
 Every claim belongs to at least one of the
categories below.
It may also belong to multiple categories.
Return one or more categories to which the
claim belongs. The majority of claims belong
to only one category.
{'Numerical Claim', 'Causal Claim', 'Quote
Verification', 'Event/Property Claim',
'Position Statement'}

<few shots>
<claim>
```

Next, in training dataset, we extract samples corresponding to the predicted claim category. We then create total 20 few-shot samples by randomly selecting four samples labeled as "supported" or "refuted," respectively and six samples from the other two labels, respectively. Each few-shot sample consists of (1) claim, (2) claim label and (3) its evidence list. Finally, we have gpt-4 to generate initial evidence list, question and answer pairs, using these few shots with following prompt:

```
 The given claim falls into one of the following
four categories.
1. Supported
2. Refuted
3. Not Enough Evidence (if there isn't
sufficient evidence to either support or refute
it)
4. Conflicting Evidence/Cherry-picking (if the
claim has both supporting and refuting evidence)

Classify each claim into four categories
and provide evidence for the classification.
If there are not enough evidences, you should
list the evidence that needs to be supported
or refuted.

<few shots>
<claim>
```

Step 2: Retrieve relevant documents and fix the initial answers using it. In the second step, we revise the initial answers for each question we generate in Step 1. Initially, for each generated question answer pair, we retrieved reliable document segments. Then, we also retrieved similar questions with each generated question for few shots. We construct each few-shot sample with (1) the retrieved questions, (2) their corresponding answers and (3) gold documents segments. For both retrieval, we leveraged ranked bm25 package which built on the

algorithm taken from (Trotman et al., 2014). Using those few shots and retrieved document segments, we fix the initial answer with following prompt:

```
 Given the context, you should find the answer
for each question.
When answering, try to use as many words from
the passage as possible.
But if you cannot find the answer, say "No
answer could be found." without extra words.

<few shots>
<claim>
<retrieved document segments>
<generated question>
```

Step 3: Infer the final answer using the given evidences. In the last step, we infer the final answer. We re-used the same samples as a few-shot in Step 1 (used in the second prompt). While in Step 1 we utilized a sequence the claim, evidence list, and label for one few-shot, in this step, we employed a sequence including (1) the claim, (2) evidence list, (3) justification, and (4) label. The justification text describes the reason why the claim is supported and refuted (Wei et al., 2022). Using gpt-4, we predict final answer with the following prompt:

```
 The given claim falls into one of the following
four categories.
1. Supported
2. Refuted
3.    Not  Enough  Evidence  (if  there  isn't
sufficient evidence to either support or refute
it)
4. Conflicting Evidence/Cherry-picking (if the
claim has both supporting and refuting evidence)

Classify  each  claim  into  four  categories
and provide evidence for the classification.
If there are not enough evidences, you should
list the evidence that needs to be supported
or refuted.

<few shots>
<claim>
<generated evidence>
```

4 Experiment

In this section, we present our experimental setup, the tools we used and the final task results.

Implementation Details The library used to obtain Youtube transcripts is youtube-transcript-api [5], and the library used for PDF parsing is PyMuPDF [6]. We used GPT-4 as an LLM and the LLM model used GPT-4, and BM25 was implemented through the langchain library [7]. For GPT-4 we use $T = 0.7$ without top-k truncation and $N = 5$, then select the last answer by majority voting (Wang et al., 2022).

Model	Q only	Q+A	AveriTeC
TUDA_MAI_0	0.45	0.34	**0.63**
HerO	0.48	**0.35**	0.57
AIC System	0.46	0.32	0.50
papelo-ten-r773	0.44	0.30	0.48
dun-factchecker	**0.49**	**0.35**	0.50

Table 1: The systems ranked in the top 5 in the AVERITEC leaderboard during the test phase. The system "dun-factchecker" is ours.

Baseline The baseline model that has been fine-tuned on BLOOM (Schlichtkrull et al., 2023) can be referred to in (Le Scao et al., 2023).

Main Results Table 1 presents the evaluation results in test phase. We have the following observations:

- Our method achieved SOTA in Q and Q+A humeteor scores, indicating that the few-shot sampling method following classification in Step 1 was effective.
- We observed that although our scores in evidence generation were higher or equal to those of the TUDA_MAI_0 and HerO systems, there was a slight drop in the performance when it comes to the final label prediction.
- It appears that utilizing generated questions and answers for retrieval was quite effective, but there are some limitations of the final prediction in the Step 3 that need to be addressed in the future.

5 Conclusion

In this work, we described the Dunamu-ml's submission to the AVERITEC shared tasks of the FEVER 2024. By integrating a language model (LLM) with a non-parametric lexicon-based method (BM25), our approach bolstered the evidence list by integrating the query and associated answers using a robust LLM. This strategy allowed us to pinpoint pertinent documents based on the

[5] https://pypi.org/project/youtube-transcript-api/

[6] https://github.com/pymupdf/PyMuPDF

[7] https://github.com/langchain-ai/langchain

generated evidence, resulting in a notable improvement over the baseline outcomes with a 0.33 performance gain in the AVeriTeC score.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

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

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.

Marina Fomicheva and Lucia Specia. 2019. Taking MT evaluation metrics to extremes: Beyond correlation with human judgments. *Computational Linguistics*, 45(3):515–558.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, 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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. 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.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, ADCS '14, page 58–65, New York, NY, USA. Association for Computing Machinery.

Andreas Vlachos and Sebastian Riedel. 2014. Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.

Nancy X. R. 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). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 317–326, Online. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain

of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.