SK_DU Team: Cross-Encoder based Evidence Retrieval and Question Generation with Improved Prompt for the AVeriTeC Shared Task

Shrikant Malviya and **Stamos Katsigiannis**
Department of Computer Science, Durham University, UK
{shrikant.malviya,stamos.katsigiannis}@durham.ac.uk

Abstract

As part of the AVeriTeC shared task, we developed a pipelined system comprising robust and finely tuned models. Our system integrates advanced techniques for evidence retrieval and question generation, leveraging cross-encoders and large language models (LLMs) for optimal performance. With multi-stage processing, the pipeline demonstrates improvements over baseline models, particularly in handling complex claims that require nuanced reasoning, by improved evidence extraction, question generation and veracity prediction. Through detailed experiments and ablation studies, we provide insights into the strengths and weaknesses of our approach, highlighting the critical role of evidence sufficiency and context dependency in automated fact-checking systems. Our system secured a competitive rank, 7th on the development and 12th on the test data, in the shared task, underscoring the effectiveness of our methods in addressing the challenges of real-world claim verification.

1 Introduction

Fact-checking has become an essential tool in the fight against misinformation, which can have far-reaching impacts on public opinion and policy. Manual fact-checking is a resource-intensive process, requiring skilled analysts to meticulously scrutinise claims and verify their authenticity. This necessity has driven the development of automated fact-checking (AFC) systems designed to assist human fact-checkers by efficiently processing large volumes of information and detecting false claims. (Nakov et al., 2021; Guo et al., 2022). The effectiveness of AFC systems depends significantly on the quality of the datasets used to train and evaluate them. Common datasets, such as FEVER (Thorne et al., 2018), FEVEROUS (Aly et al., 2021) and MultiFC (Augenstein et al., 2019), have been instrumental in advancing AFC research, but come with limitations, including the reliance on artificially constructed claims and inadequate evidence annotations (Schlichtkrull et al., 2023).

In response to these limitations, the 2024 AVeriTeC (Automated VERIfication of TExtual Claims) task was specifically designed to address the challenges of real-world claim verification (Schlichtkrull et al., 2023). AVeriTeC comprises 5,783 claims sourced from 50 fact-checking organisations, collected via the Google FactCheck Claim Search API. Each claim in the dataset is meticulously annotated with question-answer pairs, supported by online evidence, and accompanied by textual justifications explaining how the evidence leads to a verdict. This structured annotation approach ensures that the dataset supports robust AFC model training and evaluation (Schlichtkrull et al., 2023). This advancement aligns the dataset more closely with real-world scenarios, potentially enhancing the generalisation ability of the developed models and facilitating the creation of more robust approaches. The diversity of the data presents unique challenges, necessitating a deeper understanding of the data and the development of effective reasoning strategies. Our method (SK_DU) achieved the 12th Rank in the AVeriTeC shared task during the testing phase[1], providing valuable insights into the strengths and weaknesses of our pipeline and highlighting areas for further improvement.

In this paper, we aim to describe the design of our proposed fact verification pipeline and to share the insights we gained on the AVeriTeC dataset (Schlichtkrull et al., 2023) during the workshop competition. The paper introduces a comprehensive approach to real-world claim verification, leveraging the AVeriTeC dataset to develop and evaluate a sophisticated pipeline for automated fact-checking. The proposed system incorporates cutting-edge models and techniques,

[1] https://eval.ai/web/challenges/challenge-page/2285/leaderboard/5655

Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), pages 99–107

including cross-encoders for precise evidence retrieval/reranking (Humeau et al., 2019) and large language models (LLMs) for effective question generation (Schlichtkrull et al., 2023), and Cross-Encoder based natural language inference (NLI) for veracity prediction (Li et al., 2022). By focusing on multi-stage processing—ranging from the selection of evidence to nuanced reasoning for claim validation, the work addresses the complexities of real-world data, emphasising the importance of context and evidence sufficiency in fact-checking processes. Our code is released to the public for further exploration[2].

In short, the contributions of this paper are the following:

- The paper presents a detailed pipeline that integrates cross-encoders for evidence retrieval and LLMs for question generation, improving the overall accuracy of claim verification.
- Showing a pretrained Cross-Encoder model performs better than a fine-tuned BERT model on evidence extraction and reranking tasks.
- The paper provides in-depth ablation studies and performance analysis, offering insights into the strengths and weaknesses of the proposed approach.
- The model's competitive performance in the AVeriTeC shared task highlights its practical applicability and potential for real-world deployment in automated fact-checking systems.

2 Dataset Insights

AVeriTeC consists of 5,783 claims sourced from 50 reputable fact-checking organisations, where 4,568 claims' data were released earlier, while 1,215 were released during the testing phase of the AVeriTeC Shared Task[3]. Each claim is annotated with detailed question-answer (QA) pairs as evidence, a veracity label, and a textual justification, ensuring a robust foundation for training and evaluating AFC systems (Schlichtkrull et al., 2023). Additionally, the meta-data information, e.g., speaker, date, URL, location, etc., provides contextual details to the claim to support questions, answers, and justifications. This structured and meticulous approach aims to bridge the gap between academic research and practical application in building systems for misinformation detection.

Property	Stats
Avg questions per claim	2.60
Avg answers per question	1.07
Questions with extractive answer	53%
Questions with abstractive answer	26%
Questions with boolean answer	17%
Questions with no answer	4%

Table 1: Dataset statistics.

As the claims in AVeriTeC are also annotated with date, the dataset is split temporally (ordered by date) into training, validation, and test sets, having 500, 3,068, and 2,215 claims data, respectively. Table 1 illustrates some properties of the AVeriTeC dataset. Claims contain an average of 2.60 questions each, with questions averaging 1.07 answers each. Most answers are extractive (53%), followed by abstractive (26%), and boolean (17%), with 4% being unanswerable. The dataset is somewhat unbalanced, with the majority of claims being refuted, reflecting the focus of journalists on false or misleading claims.

Reasoning about evidence is structured through a question-and-answer format, allowing for multiple answers to reflect potential disagreements. Multi-hop reasoning is also allowed by referring to previous questions, and all answers must be backed by source URLs. In the AVeriTeC dataset, the veracity of claims is predicted into typical classes: Supported, Refuted, and Not Enough Evidence. AVeriTeC also introduces a fourth class: Conflicting evidence/Cherry-picking, which includes conflicting evidence and technically true claims that mislead by omitting crucial context. This addition addresses real-world scenarios where sources may legitimately disagree on interpretations.

One of the primary challenges is *context dependence*. Many claims cannot be accurately verified without additional context that is not always available in the fact-checking articles. This lack of context can lead to incorrect or incomplete verification outcomes. Another major challenge is *evidence sufficiency*. Ensuring that the evidence provided is comprehensive enough to support or refute claims is crucial, as incomplete evidence can skew the verification results. *Temporal leakage* is another critical challenge, where evidence published af-

[2] https://github.com/skmalviya/AVeriTeC_SKDU
[3] https://fever.ai/task.html

ter the claim date may inadvertently influence the verification process. This can result in biased or inaccurate conclusions, undermining the integrity of the dataset. Additionally, the diverse nature of the data from various sources and the wide range of claim types introduce complexity in data annotation and processing, making it difficult to maintain consistency and accuracy across the dataset.

3 System Description

3.1 AVeriTeC Baseline

The baseline model for AVeriTeC employs a sophisticated approach to automate the fact-checking process, leveraging state-of-the-art natural language processing (NLP) techniques. Specifically, it utilises transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) and its variants, which have proven highly effective in understanding and processing natural language. These models are fine-tuned on the AVeriTeC dataset to optimise their performance in various stages of the fact-checking pipeline, including claim representation, evidence retrieval, and veracity prediction.

The *evidence retrieval* component of the baseline model is designed to efficiently retrieve relevant evidence from a vast pool of sentences scrapped from Google Search API. The baseline applies BM25 (Robertson and Zaragoza, 2009) as a coarse filter to select the top 100 sentences to keep relevant evidence pinpointed and presented for evaluation in further stages in the pipeline.

Further, during the *question generation* stage, each evidence is paired with a question generated by an LLM based on few-shot prompting, where the QA pairs as few-shot examples are extracted from the training data using BM25. Baseline utilises BLOOM (Workshop et al., 2023) for this task. It is empirically shown that a 10-shot setting consistently outperforms other configurations, such as 1, 3, or 5-shot prompting, in generating accurate and contextually appropriate questions. To further refine the generated QA pairs, a fine-tuned BERT-large model (Devlin et al., 2019) is employed to *rerank* the outputs, ultimately selecting the top $N = 3$ evidence sets that best support or refute the claim.

The final stage of the baseline model is *veracity prediction*, where the selected evidence as QA pairs are used to determine the truthfulness of the claim. This step involves integrating the claim-evidence pairs into a coherent representation and feeding it into a classification model that assigns a veracity label. The labels typically include categories such as "supported" or "refuted", "not enough evidence" or "conflicting evidence/cherry-picking". The baseline uses a fine-tuned BERT-large model, fine-tuned on annotated examples from the AVeriTeC dataset, learning to weigh the evidence and make informed decisions about the claim's veracity (Schlichtkrull et al., 2023).

3.2 Our Pipeline

Similar to AVeriTeC, our pipeline consists of several models integrated into a multi-stage process, offering a comprehensive solution framework for real-world claim verification. Figure 1 depicts our pipeline, showing various components for a specific task. Each pipeline stage is crucial for accurate claim verification, from retrieving relevant evidence to predicting the claim's veracity. Below, we outline the models utilised in our pipeline. We make use of the evidence collection (knowledge store) retrieved through the Google Search API, as provided in the AVeriTeC shared task.

3.2.1 Evidence Selection

For *evidence retrieval*, we employ a Cross-Encoder to extract evidence sentences from the knowledge store. (Humeau et al., 2019) has shown that cross-encoders typically outperform bi-encoders on sentence-scoring tasks by enabling rich interactions between the claim and candidate evidence. We also compared the retrieval results with those of BM25, TF-IDF, and Bi-Encoder to evaluate their effectiveness. Similar to the baseline, we keep only the top 100 sentences based on the score predicted by the Cross-Encoder. The Cross-Encoder takes the pair of claim $\mathbf{c}$ and evidence $\mathbf{e}$ and processes it through a transformer model, e.g. RoBERTa (Liu et al., 2019):

$$\mathbf{h}_{[\text{CLS}]} = \text{RoBERTa}([\mathbf{c}; \mathbf{e}]) \quad (1)$$

where $\mathbf{h}_{[\text{CLS}]}$ is the final hidden state corresponding to the special [CLS] token. The score $s(\mathbf{c}, \mathbf{e})$ for the (claim, evidence) pair is then computed by applying a linear layer followed by a sigmoid activation function as:

$$s(\mathbf{c}, \mathbf{e}) = \sigma(\mathbf{W} \cdot \mathbf{h}_{[\text{CLS}]} + b) \quad (2)$$

where $\mathbf{W}$ and $\mathbf{b}$ are the linear layer's weight matrix and bias term, and σ is the sigmoid function.

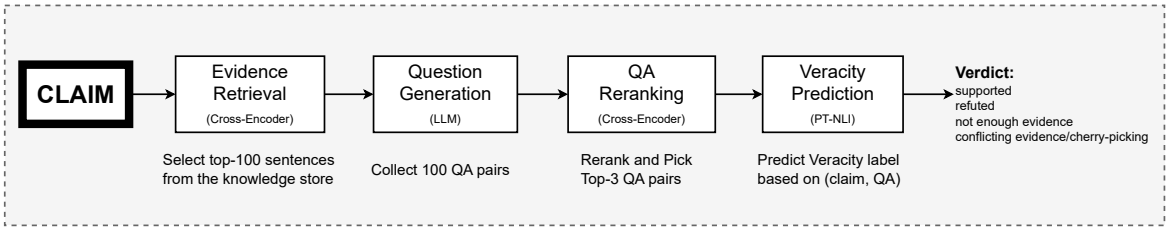

Figure 1: Overview of the pipelined Evidence-Retrieval and Verdict Prediction for a given claim.

Evidence:...
Question answered:...

Evidence:...
Question answered:...

...

Evidence:...
Question answered:

(a) Prompt1

Outrageously, example['speaker'] Claim:...
with Evidence:...
as an answer to the Question:...

Outrageously, example['speaker'] Claim:...
with Evidence:...
as an answer to the Question:...

...

Outrageously, example['speaker'] Claim:...
with Evidence:...
as an answer to the Question:...

(b) Prompt2

Figure 2: Prompts used by an LLM for question generation task.

This strategy ensures that the most pertinent evidence is identified (relevance) and made computationally feasible (top-100) for further stages in the verification pipeline.

3.2.2 Question Generation

To generate questions for the extracted evidence sentences from the previous step, we conducted experiments on two fronts: 1) Prompt Engineering, and 2) Utilisation of Various Large Language Models (LLMs).

Prompt Engineering We experimented with two prompt configurations for few-shot learning:

Prompt1: A straightforward pair of evidence and questions.

Prompt2: A more descriptive prompt that includes a triplet of claim, answer, and question.

Figure 2 illustrates the prompt configurations employed in our study. In "Prompt2", if a sample lacks a 'speaker' field or is set to NULL, we substitute it with "Speaker" to maintain consistency across the prompts.

In line with baseline criteria for question generation, we adopt a 10-shot approach for prompt construction. Additionally, we explored using the Bi-Encoder model to identify the 10 most relevant examples from the training set for prompting. The Bi-Encoder, based on a transformer architecture, is effective in retrieving in-context examples, enhancing the quality of few-shot prompting. An ablation study in the results section compares the effectiveness of these approaches.

Utilisation of Various Large Language Models (LLMs) With the GPU resources at our disposal, we conducted question-generation experiments using LLMs with up to 8 billion parameters. We evaluated leading open-source models such as BLOOM (Workshop et al., 2023) and Meta-Llama-3-8B (Dubey et al., 2024). Additionally, we tested the recently released Meta-Llama-3.1-8B for the generation task. For comparison, we also utilised the ChatGPT API[4] with the 'OpenAI-GPT-4o' model.

3.2.3 Question-Answer Reranking

After retrieving the initial set of evidence, we apply a reranking process to ensure that the most relevant pieces are selected for the claim verification task. This reranking is essential for identifying specific question-answer (QA) pairs that directly support or refute the claim, thereby sharpening the focus

[4] https://platform.openai.com/docs/api-reference/introduction

on the most pertinent information. To achieve this, we again utilise a Cross-Encoder model, which is particularly effective in capturing nuanced relationships between the claim and the evidence. At this stage, the input format changes to (claim, QA), allowing the model to evaluate the alignment between the claim and the concatenated question-answer (QA) pairs as:

$$\mathbf{h}_{[\text{CLS}]} = \text{RoBERTa}([\mathbf{c}; \mathbf{q} \cdot \mathbf{a}]) \tag{3}$$

the final hidden state $\mathbf{h}_{[\text{CLS}]}$ is then processed through a linear layer followed by a sigmoid activation function (as in Equation 2) to obtain a score $s(\mathbf{c}, \mathbf{qa})$ for the (claim, QA) pair.

By carefully selecting the most relevant evidence, the system significantly reduces noise and enhances the precision of the information used in the final verification step. This meticulous approach ensures that the verification process is not only accurate but also efficient, ultimately leading to more reliable outcomes in automated fact-checking.

3.2.4 Veracity Prediction

Veracity prediction is the final and most critical stage in the automated fact-checking pipeline. In this stage, the model classifies a claim based on the evidence retrieved (e.g., Top 3 QA pairs) and selected in previous stages to predict its veracity into four classes. Unlike the baseline approach using a BERT-Large model, we fine-tune a Cross-Encoder—a smaller, transformer-based model—through supervised natural language inference (NLI) training. This approach is computationally less expensive and well-suited for entailment tasks, where it infers the relationship between pairs of sentences (premise and hypothesis) (Li et al., 2022)

We use the Cross-Encoder with a text classification head for the task. Similar to Equation 3, the claim $\mathbf{c}$ and evidence pair $\mathbf{q} \cdot \mathbf{a}$ are inputted to the model to obtain an encoded input representation $\mathbf{h}_{[\text{CLS}]} = \text{RoBERTa}([\mathbf{c}; \mathbf{q} \cdot \mathbf{a}])$. The hidden state $\mathbf{h}_{[\text{CLS}]}$ is then passed through a linear layer (classification head) followed by a softmax activation function to produce a probability distribution $\mathbf{p}$ over the possible veracity labels (e.g., supported, refuted, insufficient evidence, conflicting/cherry-picking) as:

$$\mathbf{p} = \text{softmax}(\mathbf{W} \cdot \mathbf{h}_{[\text{CLS}]} + b) \tag{4}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias term of the linear layer. The output $\mathbf{p}$ is a vector of probabilities corresponding to each veracity class.

The model is trained using a cross-entropy loss function, which measures the difference between the predicted probability distribution and the true distribution. If $\mathbf{y}$ is the true label (encoded as a one-hot vector) and $\mathbf{p}$ is the predicted probability distribution, the loss function $\mathcal{L}$ is given by:

$$\mathcal{L} = -\sum_{k=1}^{K} y_k \log(p_k) \tag{5}$$

where K is the number of veracity classes, y_k is the true label for class k, and p_k is the predicted probability for class k. The model parameters are optimised to minimise this loss, thereby improving the accuracy of veracity prediction.

4 Experiments

4.1 Evaluation Metrics

In the evaluation of the AVeriTeC dataset and the associated automated fact-checking (AFC) systems, several metrics are employed to assess the performance at various stages of the pipeline. These stages consist of evidence retrieval, evidence selection, and veracity prediction. The metrics are designed to comprehensively measure the effectiveness and accuracy of each component, ensuring robust evaluation and comparison.

Unlike the FEVER dataset and others that use a closed source of evidence like Wikipedia, AVeriTeC is designed to retrieve evidence from the open web. This approach can result in finding the same evidence across multiple sources, making exact matching impractical for scoring purposes. Therefore, a Hungarian algorithm-based pairwise scoring function $f : S \times S \rightarrow \mathbb{R}$ is utilised to evaluate how well a set of generated sequences, such as questions or answers, aligns with the reference sequences of tokens. The Hungarian algorithm provides the solution as a boolean function $X : \hat{Y} \times Y \rightarrow \{0, 1\}$, maximising the assignment problem between the generated sequences $\hat{Y}$ and the reference sequences Y (Crouse, 2016). This metric, referred to as the Hungarian METEOR (HuMETEOR) score s_f and is then calculated between $\hat{Y}$ and Y as:

$$s_f(\hat{Y}, Y) = \frac{1}{|Y|} \max \sum_{\hat{y} \in \hat{Y}} \sum_{y \in Y} f(\hat{y}, y) X(\hat{y}, y) \tag{6}$$

where f denotes METEOR, a pointwise scoring function, and X is a boolean function optimised as a linear sum assignment problem. The Final Hu-METEOR score is estimated as the mean of scores between all pairs of generated and reference sequences. The Hu-METEOR is used twice to evaluate questions-only sequences and concatenated question-answer (QA) pairs.

AVeriTeC Score is an accuracy metric utilised to compare the overall performance of the system. The metric considers veracity prediction True for a given claim if the Hu-METEOR score between generated and reference evidence is above a certain threshold ($\lambda > 0.25$):

$$\text{AVeriTeC\_Score} = \frac{1}{|C|} \sum_{c \in C} \big(c_{\text{pred\_label}} == c_{\text{true\_label}}, f(c_{\hat{y}}, c_y) > (\lambda = 0.25)\big) \quad (7)$$

where, $c_{\text{pred\_label}}$, $c_{\text{true\_label}}$ denotes predicted and true labels, respectively, and $c_{\hat{y}}$ and c_y are the generated and reference evidence sets of the claim.

4.2 Implementation Details

Table 2 provides a comprehensive overview of the models used within the various components of our pipeline, including specific details and the corresponding checkpoints.

In the evidence retrieval step, we extracted sentences from the provided knowledge store using three models: 1) BM25 (AVeriTeC baseline), 2) Bi-Encoder, and 3) Cross-Encoder, for comparison. For the Bi-Encoder, we employed the standard BERT model with a hidden size of 768. For the Cross-Encoder, we utilised a smaller transformer model with a hidden size of 384, fine-tuned specifically for reranking tasks such as MS-Marco Passage reranking (Nguyen et al., 2016). We set the batch size to 32 for both Bi-Encoder and Cross-Encoder. The average time in scoring 1,000 sentences by BM25, Cross-Encoder, and Bi-Encoder are 10.9, 31.9, and 80.3 milliseconds, respectively.

For the *question generation* task, we leverage several large language models (LLMs), including BLOOM, Meta-Llama-3-8B, and Meta-Llama-3.1-8B. For comparison, ChatGPT's GPT-4o model is accessed through its API. Due to financial restrictions, the questions are generated only for the top 25 evidence with ChatGPT. The average time to generate a single question varies across the models, with BLOOM taking 8.9 seconds, Meta-Llama-3-8B taking 3.1 seconds, and Meta-Llama-3.1-8B taking 3.6 seconds. This performance data highlights the efficiency of the Meta-Llama models, particularly in resource-constrained environments. For prompting, BM25 and Bi-Encoder are considered for selecting the 10 most relevant examples from the training set for prompting.

For the *Question-Answer reranking*, Cross-Encoder with 'ms-marco-MiniLM-L-12-v2' checkpoint is utilised instead of the baseline's BERT-large model. It requires no training and is computationally less expensive due to its smaller size, leading to 5 times faster performance. For each claim, it takes approx 40 ms to reorder the QA pairs.

The final stage *verdict prediction* involves training a supervised NLI model as an entailment task. The model takes a pair of a claim and concatenated QA as input and predicts a veracity label. With a cross-encoder setting, we fine-tune a DeBERTa-NLI model on examples from train/development data using Adam (Kingma and Ba, 2017) with a learning rate of 2e-5 and a batch size of 16 for four epochs.

All the experiments were conducted on an NVIDIA RTX 6000 Ada 48GB type GPUs.

5 Results

The proposed pipeline's evaluation involved a comprehensive analysis of performance across various stages, including evidence retrieval, evidence selection, and veracity prediction. The results highlight the effectiveness of the proposed approach in handling the complexities of real-world claim verification and the challenges encountered during the process.

5.1 Evidence Selection

In the evidence retrieval step, we extract the top-100 evidence sentences for each claim from a vast pool of a knowledge store. Table 3 shows the Hu-METEOR based retrieval score by various methods, i.e. BM25, TF-IDF, Bi-Encoder and Cross-Encoder. The Cross-Encoder model demonstrated strong performance in identifying pieces of evidence that were most relevant to the claims. The model's ability to consider both the claim and the evidence sentence jointly allowed it to capture nuanced relationships, leading to improved evidence selection effectively. Additionally, its lightweight architecture makes it comparable to Bi-Encoder.

Models	Checkpoint	Hidden Size	#Parameters	Task
Cross-Encoder	ms-marco-MiniLM-L-12-v2[5]	384	22.7M	Evidence-Retr, QA Reranking
Bi-Encoder	bert-base-uncased[6]	768	109.5M	Evidence-Retr, 10-Shot Prompt
BLOOM	bloom-7b1[7]	4096	7B	Q-Generation
Meta-3	Meta-Llama-3-8B[8]	4096	8B	Q-Generation
Meta-3.1	Meta-Llama-3.1-8B[9]	4096	8B	Q-Generation
ChatGPT	Openai-GPT-4o[10]	–	–	Q-Generation
DeBERTa-NLI	deberta-v3-base[11]	768	82M	Veracity Prediction

Table 2: The details for models used for various tasks in the pipeline.

Models	A only @ (3 / 5 / 10 / 50 / 100)				
BM25 (baseline)	0.1027	0.1207	0.1452	0.2049	0.2338
TF-IDF	0.1062	0.1237	0.1474	0.2077	0.2382
Bi-Encoder	0.1311	0.1521	0.1787	0.2474	0.2753
Cross-Encoder	**0.1413**	**0.1624**	**0.1913**	**0.2614**	**0.2907**

Table 3: Results of evidence selection in terms of Hu-METEOR on the development set.

Prompt Setting	Few-Shot Selection	Q only @ (3 / 5 / 10 / 100)				QA only @ (3 / 5 / 10 / 100)			
Prompt1	Bi-Encoder	0.21	0.25	0.30	0.43	0.22	0.25	0.28	0.36
Prompt1	BM25	0.23	0.27	0.33	0.46	0.22	0.25	0.28	0.36
Prompt2	Bi-Encoder	0.24	0.29	0.34	0.48	**0.23**	**0.26**	**0.29**	0.37
Prompt2	**BM25**	**0.26**	**0.30**	**0.36**	**0.49**	**0.23**	**0.26**	**0.29**	**0.38**

Table 4: Influence of Prompt setting on question generation. `bigscience/bloom-7b1` is used as LLM for generation.

5.2 Question Generation

We consider various LLMs for the question generation task based on the extracted evidence, i.e. bloom-7b1, Meta-Llama-3-8B, Meta-Llama-3.1-8B, and Openai-GPT-4o. We also experimented with sparse, e.g. BM25, and dense, e.g. Bi-Encoder, methods for selecting few-shot examples during prompt construction. The result on prompt construction is shown in Table 4 with both few-shot selection methods under prompt-setting Prompt1 and Prompt2. We found that a descriptive prompt can generate relevant questions in the context of given claims and evidence pairs. This shows BM25's superiority to Bi-Encoders for few-shot example selection in prompting due to its emphasis on exact term matching and robustness in low data scenarios.

[5]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[6]https://huggingface.co/google-bert/bert-base-uncased

[7]https://huggingface.co/bigscience/bloom-7b1/tree/main

[8]https://huggingface.co/meta-llama/Meta-Llama-3-8B

[9]https://huggingface.co/meta-llama/Meta-Llama-3.1-8B

[10]https://platform.openai.com/docs/models/gpt-4o

[11]https://huggingface.co/microsoft/deberta-v3-base

LLM	Q only @ (3 / 5 / 10 / 100)				QA only @ (3 / 5 / 10 / 100)			
bloom-7b1	0.26	0.30	0.36	0.49	0.23	0.26	0.29	0.38
Meta-Llama-3-8B	0.28	0.32	0.37	0.49	0.23	0.26	0.29	0.38
Meta-Llama-3.1-8B	0.28	0.32	0.37	0.49	0.23	0.26	0.30	0.38
Openai-GPT-4o	**0.41**	**0.45**	**0.49**	–	**0.25**	**0.29**	**0.32**	–

Table 5: Influence of using various LLMs on question generation task. Few-shot selection is done by BM25. Openai-GPT-4o has been used to generate questions for only the first 25 sentences.

Reranking Models	LLM	Q only @3	A only @3	QA @3
BERT-Dual Encoder (baseline)	Meta-Llama-3-8B	0.2799	0.1173	0.2032
	Meta-Llama-3.1-8B	0.2832	0.1199	0.2069
	Openai-GPT-4o	0.4023	0.1392	0.2464
Cross-Encoder	Meta-Llama-3-8B	0.2991	0.1360	0.2341
	Meta-Llama-3.1-8B	0.3018	0.1323	0.2334
	Openai-GPT-4o	**0.4122**	**0.1374**	**0.2584**

Table 6: Results of post-QA reranking Hu-METEOR score @3 through BERT-Dual Encoder (baseline) and Cross-Encoder.

Table 5 depicts the influence of using various LLMs for question generation. It shows Meta models are better than BLOOM due to their bigger architecture and being trained on more diverse and high-quality data (Dubey et al., 2024). ChatGPT-based Openai-GPT-4o model has shown a 0.13 jump in Hu-METEOR score on Q only @3, achieving an overall high performance on AVeriTeC task.

5.3 QA Reranking

In the *question-answer reranking* stage, a pre-trained Cross-Encoder is utilised to select top QA pairs achieving higher Hu-METEOR scores than the baseline's BERT-large, which requires explicit fine-tuning on the training data. Table 6 presents the Hu-METEOR scores for questions only (Q), answers only (A), and combined question-answer (QA) across various LLMs, including Meta-Llama-3-8B, Meta-Llama-3.1-8B, and OpenAI-GPT-4o. The Cross-Encoder based reranking consistently outperforms the baseline in question generation.

LLM	Development set				Test set			
	Q Only	A Only	QA	A.S	Q Only	A Only	QA	A.S
Official Baseline	0.24	–	0.19	0.09	0.24	–	0.20	0.11
Meta-Llama-3-8B	0.2992	0.1360	0.2342	0.1780	0.2976	–	0.2409	0.1986
Meta-Llama-3.1-8B	0.3018	0.1323	0.2334	0.1900	0.2978	–	0.2405	0.1937
Openai-GPT-4o	**0.4122**	**0.1374**	**0.2584**	**0.2240**	**0.3961**	–	**0.2613**	**0.2239**

Table 7: Performance on the development set and test set. A.S is the AVeriTeC score, and Q Only, A Only, and QA are the Hu-METEOR scores of question, answer and question-answer, respectively.

5.4 Overall results: Veracity Prediction

The veracity prediction stage was crucial for determining the final classification of the claims. We fine-tuned a transformer-based classification model, DeBERTa-NLI, on the AVeriTeC dataset, achieving strong results in classifying claims into the predefined categories: supported, refuted, insufficient evidence, and conflicting/cherry-picking. The model's performance was evaluated using metrics Q Only, A Only, QA, and A.S (AVeriTeC Score), where the Q Only, A Only, QA scores are Hu-METEOR scores of the retrieved evidence and A.S is a special metric that considers veracity prediction true for a given claim if the Hu-METEOR is above a certain threshold ($\lambda = 0.25$) as shown in Table 7. We observe that under the same pipeline models, Meta LLMs outperform the baseline by 0.9 to 0.10 AVeriTeC score through obtaining improved QA evidence. Openai-GPT-4o shows a remarkable improvement in question generation, which leads to achieving a higher overall AVeriTec score on both development and test data.

6 Conclusion

In this paper, we presented a comprehensive pipeline for real-world claim verification tailored to the AVeriTeC dataset. Our approach, which integrates cross-encoders for evidence retrieval and LLMs for question generation, has shown to be effective in improving the accuracy of automated fact-checking systems. We show that the cross-encoder performs better than the baseline on both evidence extraction and reranking. The results of our experiments highlight the importance of multi-stage processing and the careful selection of evidence to support or refute claims. Our model's performance in the AVeriTeC shared task demonstrated its potential in real-world applications, particularly in scenarios requiring detailed reasoning and context understanding. Although our system has made significant strides in addressing the complexities of real-world claim verification, further improvements are necessary, particularly in handling ambiguous claims and ensuring the completeness of evidence.

7 Limitations

Despite the promising results, our approach has several limitations. First, we rely on the knowledge store provided by the shared task; therefore, retrieving evidence from scratch from Google with better scrapping and parsing methods may provide a better knowledge space. Secondly, the reliance on cross-encoders, while effective, is computationally expensive, which may hinder scalability in real-time applications. Additionally, advanced reranking models, such as HLATR (Zhang et al., 2023), HybRank (Zhang et al., 2022), and M-ReRank (Malviya and Katsigiannis, 2024) can further enhance evidence retrieval. Thirdly, "the performance of our question generation model, though robust, can be affected by the quality and diversity of few-shot examples used for prompting.

Additionally, our system's ability to handle claims with insufficient or conflicting evidence remains a challenge, often leading to less accurate veracity predictions. Finally, the dataset's temporal dependency introduces potential biases, as evidence published after the claim date could influence the verification process. Addressing these limitations will be crucial for enhancing our system's robustness and generalisability in future work.

Acknowledgements

The authors in this project have been funded by UK EPSRC grant "AGENCY: Assuring Citizen Agency in a World with Complex Online Harms" under grant EP/W032481/2.

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. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1. 1

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. 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 4685–4697, Hong Kong, China. Association for Computational Linguistics. 1

David F. Crouse. 2016. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696. 5

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. 3

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and Others. 2024. The Llama 3 Herd of Models. 4, 7

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206. 1

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *International Conference on Learning Representations*. 2, 3

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. 6

Shuang Li, Xuming Hu, Li Lin, and Lijie Wen. 2022. Pair-Level Supervised Contrastive Learning for Natural Language Inference. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8237–8241. 2, 5

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. 3

Shrikant Malviya and Stamos Katsigiannis. 2024. Evidence Retrieval for Fact Verification using Multi-stage Reranking. In *ACL Rolling Review - June 2024*. 8

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated Fact-Checking for Assisting Human Fact-Checkers. In *Twenty-Ninth International Joint Conference on Artificial Intelligence*, volume 5, pages 4551–4558. 1

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human-Generated MAchine Reading COmprehension Dataset. 6

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389. 3

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A Dataset for Real-world Claim Verification with Evidence from the Web. *Advances in Neural Information Processing Systems*, 36:65128–65167. 1, 2, 3

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. 1

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, and Others. 2023. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. 3, 4

Yanzhao Zhang, Dingkun Long, Guangwei Xu, and Pengjun Xie. 2022. HLATR: Enhance Multi-stage Text Retrieval with Hybrid List Aware Transformer Reranking. 8

Zongmeng Zhang, Wengang Zhou, Jiaxin Shi, and Houqiang Li. 2023. Hybrid and Collaborative Passage Reranking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14003–14021, Toronto, Canada. Association for Computational Linguistics. 8