UCOM_UNAM_PLN at CheckThat! 2025: Evaluating LLMs in a two-Step Architecture for Numerical Fact Checking*

Notebook for the CheckThat! Lab at CLEF 2025

Guido Acosta[1,†], Eduardo Morales[1,†] and Helena Gómez-Adorno[2]

[1] *Maestría en Ciencia de Datos, Universidad Comunera, Monseñor Bogarín 284, Asunción, Paraguay*

[2] *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Av. Universidad 3000, C.U., Coyoacán, Ciudad de México, México*

Abstract

This paper presents the approach developed by the UCOM-UNAM-PLN team for Task 3 of the CLEF 2025 CheckThat! Lab, focused on verifying numerical and temporal claims in Spanish. We propose a two-step fact-checking architecture leveraging large language models (LLMs) in both stages: (1) evidence retrieval through BM25-preselected documents, and (2) veracity classification using zero-shot and few-shot prompting strategies. We conduct a comprehensive evaluation of retrieval techniques—including statistical, embedding-based, and LLM-based models—on the CLEF 2024 development set, identifying GPT-4o as the top-performing retriever. Based on these findings, we deploy GPT-4o for both evidence extraction and classification. Our best-performing pipeline combines few-shot prompting with curated evidence, achieving a macro-averaged F1 score of 0.3595 on the official CLEF 2025 test set. Results highlight the effectiveness of hybrid LLM-based architectures in identifying false claims, while also revealing challenges in handling ambiguous or true statements. This study underscores the potential of combining semantic retrieval and task-specific prompting for robust fact verification in real-world contexts.

Keywords

Fact checking, LLMS, Classification, Information Retrieval

1. Introduction

This paper presents our approach to the task 3 - Fact-Checking Numerical Claims of the CLEF 2025[1] – CheckThat! Lab [1] The task consists of verifying claims with numerical quantities and temporal expressions [2]. Claims must be classified as True, False, or Conflicting based on a short list of evidence. We only report our participation in the Spanish challenge.

Organizers provided documents retrieved by top-k BM25 ranking approach. This documents can be used to determine a label for each claim. The evidence documents were provided in the order retrieved by a BM25-based retriever, but participants can employ some re-ranking approaches to improve fact verification performance. Participants were also able to use other approaches to obtain evidence from the corpus provided by the organizers.

Our approach involves using LLMs in different ways for the classification of claims. We use LLM models in two ways, first to classify the claim based on the complete document and then to retrieve relevant sentences within the document. These sentences were used later as fine-grained evidence to classify the claims.

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 describes the dataset utilized in this study. Section 4 details our approach. Section 5 presents previous experiments conducted for CheckThat 2024. Section 6 shows our results in CheckThat 2025. Section 7 outlines some limitations of our approach. Finally, Section 8 concludes the paper.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[†] These authors contributed equally.

guidoacosta@s.ucom.edu.py (G. Acosta); eduardo.morales@t.ucom.edu.py (E. Morales); helena.gomez@iimas.unam.mx (H. Gómez-Adorno)

0009-0005-9589-6456 (G. Acosta); 0009-0003-4615-7214 (E. Morales); 0000-0002-6966-9912 (H. Gómez-Adorno)

[1] https://checkthat.gitlab.io/clef2025/

2. Related Work

In their state-of-the-art review, [3] organize automated fact-checking methods according to four-stage architecture proposed by [4]: (1) claim detection, where statements that can be verified are extracted; (2) evidence retrieval, which involves locating relevant information from external sources such as databases or textual corpora; (3) claim verification, which entails determining whether the evidence supports, refutes, or does not allow a conclusion about the claim; and (4) justification generation, where a textual explanation is produced to accompany the system's decision. In [5], the fact-checking process mainly consists of two stages. First, an evidence retrieval model based on transformers extracts a relevant text snippet from a trusted source, which is crucial for assessing the validity of a claim. Then, a claim verification model, also based on transformers, uses the retrieved evidence to classify the claim as true, false, or "not enough information" (NEI) if the evidence is inconclusive. In this paper, we will focus on the two-step architecture.

In their review on automated fact-checking, [3] identify three main approaches for evidence retrieval: traditional lexical-based methods like BM25, dense embedding-based methods, and hybrid approaches that combine initial retrieval and re-ranking with more complex models. For information retrieval, [5] employs a transformer-based model inspired by extractive question-answering systems. This model, typically BERT-based, takes a claim and a reliable source (such as a Wikipedia article) and extracts a relevant text snippet for claim verification. The training is done by fine-tuning pre-trained BERT models (such as BERT and DistilBERT) using the FEVER dataset. In the experiments conducted, DistilBERT achieved an exact match score of 90.19% and an F1-score of 93.98%, slightly outperforming BERT, which obtained an exact match score of 89.89% and an F1-score of 93.93%.

In [5] the claim verification model is based on BERT. This model takes as input the original textual claim and the retrieved textual evidence. Sentence embeddings for the claim and the evidence are obtained through a pre-trained BERT embedding layer. These embeddings are then processed and passed to a fully connected layer. Finally, the output of this layer is used to classify the claim into one of three categories: true, false, or "not enough information" (NEI), determining whether the evidence is sufficient to verify the claim. In the verification stage in [6], focuses primarily on the use of Large Language Models (LLMs). These LLMs, such as `GPT-4` and `Llama3-70B-Instruct`, are guided by system prompts to analyze a claim and the provided textual evidence. Their goal is to predict a label (SUPPORTS, REFUTES, or NOT ENOUGH INFO) along with a confidence score, leveraging their deep understanding of natural language for complex reasoning tasks. In [7], the authors conducted a comprehensive analysis on the identification of critical statements using various strategies, including lexical representations, embeddings, and LLMs. Specifically, they evaluated the use of LLMs as direct classifiers by applying zero-shot and few-shot prompting strategies, without requiring additional training.

Large Language Models (LLMs) are emerging as a promising component in fact-checking systems. In this work, we explore the use of Large Language Models (LLMs) in both stages of the fact-checking architecture.

3. Dataset

To address this challenge we only used the *development dataset* of the Checkthat 2025 Lab [2] which contains elements in JSON format with the following structure:

```
{
    "claim": "San Luis no tiene empleados públicos en exceso; hoy son 17 mil",
    "crawled_date": "2011-09-26",
    "country_of_origin": "argentina",
    "doc": "Las cifras del Presupuesto provincial son consistentes con esa afirmación. No así los resultados de la Encuesta Permanente de Hogares. Otros indicadores muestran que la provincia tiene una alta tasa de empleo en negro y de planes sociales. San Luis es [la provincia] mejor administrada de la República porque no tiene empleados públicos en exceso; hoy son 17 mil e
```

```
        inclusive creemos que son excesivos, subrayó Alberto Rodríguez Saá en una
        realizada en la escuela de periodismo TEA.",
    "fact_source": "chequeado",
    "factchecker": [],
    "label": "False",
    "url": "https://chequeado.com/ultimas-noticias/rodriguez-saa-san-luis-no-tiene-
        empleados-publicos-en-exceso-hoy-son-17-mil/",
    "lang": "es",
    "numerical": true,
    "taxonomy_label": "statistical"
}
```

The *test dataset* with 1.806 claims has a different structure than the *development dataset* with 377 claims. This dataset contained only the *claims* without associated documents. For this reason, it was necessary to perform an additional task to collect the corresponding evidence documents for each claim. To obtain the evidence, we used the corpus provided by the challenge organizers. In this context, we considered two alternatives: (1) using the pre-selected evidence provided by the organizers, which was retrieved using the BM25 algorithm, and (2) developing a custom function to perform automatic evidence retrieval from the corpus. The procedures implemented in both strategies are detailed in the following sections.

Table 1 presents the statistic of the development. It shows class distribution of the development dataset, documents length, and the average evidence length measured in characters.

Table 1
Development set statistics

Claim Class	Quantity	Avg. Evidence Length
True claims	30	4756
False claims	299	3495
Conflicting claims	48	5161
Total	**377**	

4. Method

From each entry of the development dataset, we extracted: the *claim*, the *doc* (the document containing the associated evidence) and the *label*.

Our proposed approach is based on the use of large language models (LLM), which were employed in two different ways: (1) as a classifier, assigning a label to each claim based on the available evidence, and (2) as an evidence retriever, for identifying relevant fragments within the document that are later used to support, refute, or indicate conflict regarding the *claim*.

Both approaches were configured under two different inference schemes: *zero-shot*, in which the model performs the task without having received prior examples during the interaction, and *few-shot*, where representative examples are provided within the prompt to guide its behavior and attempt to improve accuracy in classification or evidence extraction.

The following section provides a detailed description of both stages of the approach.

4.1. Claim classifier

In this first approach, the LLM was instructed to act as an automatic claim classifier. Its goal was to analyze each claim along with its associated evidence and assign a veracity label based on the provided content. The instructions used are shown below:

```
You are an assistant focused on fact verification involving numerical claims and temporal expressions.
Your role is to classify the truthfulness of each claim based on the evidence provided.
Numerical claims are defined as those requiring validation of explicit or implicit quantitative or temporal details.
```

```
You will receive data in JSON format. Each instance will include:

- a claim (the statement to verify),
- and an evidence (the supporting or contradicting information),

You must analyze the evidence and assign one of the following labels to the claim:

- True
- False
- Conflicting

Only respond with the label. No additional explanation is required.
```

In the case of the *few-shot* approach, the same base instructions as in the *zero-shot* scenario were used, but labeled examples were added to the *prompt* in order to guide the model in the task. To do this, the following instruction was included:

```
You will first receive a few labeled examples. Use these examples as guidance to determine the correct label for the
      following unlabeled instances. Examples:
```

Following this instruction, an example of a claim with its corresponding evidence and label was included for each of the possible classes (*True, False, and Conflicting*). These examples were intended to illustrate both the expected format and the necessary criteria for the model to perform a coherent classification aligned with the task. It is worth noting that, in order to identify the examples to be used, a manual curation was carried out to obtain representative examples.

4.2. Relevant evidence retriever

In a second approach, the LLM was instructed as an evidence retriever, with the goal of identifying relevant fragments within the document that could later be used in the classification of the claim. Specifically, the model was asked to extract three sentences from the *doc* field, assigning each one a label indicating its relation to the claim (*True, False, or Conflicting*), depending on whether the sentence supports, contradicts, or shows ambiguity with respect to the *claim.* The instructions used for the *zero-shot* case are provided below:

```
You are an assistant focused on fact verification involving numerical claims and temporal expressions. Your role is to
      analyze the provided evidence based on the given claim and return three sentences from the evidence that are related to
      the claim. You will receive data in JSON format. Each instance will include: a claim (the statement to verify),  and an
      evidence (contains information related to the claim). You must analyze the evidence in relation to the claim and select
      up to three sentences. For each selected sentence, you must determine if it supports ("True"),  if it contradicts ("
      False") or if there is suspicion about the truthfulness of the claim ("Conflicting"). Your response must be in JSON
      format, with a list of dictionaries. Each dictionary should have the following keys: "sentence": the extracted sentence
      from the evidence, "verdict": "True" if the sentence supports the claim, "False" if it contradicts the claim or "
      Conflicting" if there is suspicion about the truthfulness of the claim. The JSON should have the following structure:
      [{"sentence": "a sentence", "verdict": "True"}, {"sentence": "another sentence", "verdict": "False"}, {"sentence": "yet
      another sentence", "verdict": "True"}] You should select at most three sentences. If fewer than three relevant sentences
       are found, return only the sentences found. Only respond with the JSON output. No additional explanation is required.
```

In the case of the *few-shot* approach, the same base instructions as in the *zero-shot* scenario were used, but labeled examples were added to the *prompt* in order to guide the model in the task. To do this, the following instruction was included:

```
Here are some examples. Use these examples as a guide to get the correct sentences.
```

Once the relevant sentences were obtained, the following logic was applied to determine the final label of the claim:

- If at least one of the sentences was labeled as *Conflicting*, the claim was classified as *Conflicting*.
- In the absence of *Conflicting* sentences, if at least one sentence was labeled as *False*, the claim was classified as *False*.
- If all the sentences were labeled as *True*, the claim was classified as *True*.
- In cases where the retriever failed to identify relevant fragments and therefore returned no sentences, the claim was classified as *False* by default.

In this work, we adopted this heuristic under the assumption that the absence of relevant evidence suggests that the claim is false. If a claim were true or conflicting, it is likely that verifiable information associated with it would exist in the document corpus. Therefore, a complete lack of retrieved evidence could lead us to conclude that the claim is false in this context. It is possible that there are cases in which no evidence exists—either because the evidence retriever failed to find the relevant evidence or because such evidence simply does not exist. In either case, a default class must be chosen at the time of classification. We acknowledge that an alternative could be to classify these cases as “not enough information for a verdict,” however, this class is not used in the current system. It is clear that this heuristic may introduce bias toward the False class. Nevertheless, introducing a new class like NOT ENOUGH INFO could also introduce bias toward that class, since the classification stage strongly depends on the evidence retrieval stage.

4.2.1. Use of evidence provided by the organizers

The organizers provided a file containing the *claims* along with the 100 most relevant pieces of evidence retrieved from the *corpus* using the BM25 algorithm. Each entry in the file had the following structure:

```
{
    "doc_id": [list of document identifiers],
    "scores": [list of scores associated with each selected evidence],
    "query_id": "claim identifier number",
    "claim": "claim",
    "docs": [list of corresponding evidence]
}
```

For the experiments, the five pieces of evidence with the highest scores were selected for each *claim*. These pieces of evidence were concatenated into a single text, which served as the input document (*doc*). This document, along with the corresponding claim, was sent to the assistant to assign the veracity label or to extract the most relevant evidence.

5. Previous Experiments on Fact Checking

Before addressing Task 3 of the CheckThat! 2025 challenge — Fact-Checking of Numerical Claims — , we ran preliminary experiments using the dataset from the CheckThat! 2024 Rumor Verification task [8]. The goal of this task was to explore information retrieval techniques and select the most suitable one as the relevant evidence retriever.

We conducted a comprehensive evaluation of a wide range of retrieval techniques, grouped into three main categories:

- **Traditional statistical techniques** [9, 10, 11, 12, 13, 14, 15, 16]:
  BM25, BM25PLUS, BM25-OKAPI, BM25+PL2, BM25LARGE, PL2, TF-IDF, DPH, DPH+PL2, DLH, LEMUR, HIEMSTRA, DFIZ, DFIC, DFR-BM25.
- **Embedding-based models** [17, 18, 19] :
  OpenAI: text-embedding-3-large, text-embedding-3-small.
  Gemini: text-embedding-005-FACT-VERIFICATION, text-embedding-005-SEMANTIC-SIMILARITY, text-embedding-005-RETRIEVAL-DOCUMENT.
  SBERT: SBERT-all-MiniLM-L6-v3.
- **Large Language Models (LLMs) used as semantic retrievers** [20, 21, 22, 23]:
  GPT-4o: gpt-4o-2024-08-06, gpt-4o-mini-2024-07-18.
  Qwen: qwen2.5-72b.
  LLAMA: LLAMA3.2LARGE.
  Deepseek: deepseek-llm-67b.

These techniques were evaluated on the CheckThat! 2024 development set, which contains 32 rumors, using Recall@5 (R@5) [24] which measures the model’s ability to retrieve at least one relevant document among the top 5 results and Mean Average Precision (MAP) which measures the model’s ability to

Table 2
Performance comparison of retrievers on Recall@5 and MAP. Results ordered by MAP

Retriever Label	R@5	MAP	Type
GPT-4o-2024-08-06	0.940	0.918	LLM
QWEN2.5-72b	0.791	0.787	LLM
GPT-4o-mini-2024-07-18	0.747	0.747	LLM
OPENAI-text-embedding-3-large	0.800	0.731	Embedding
GEMINI-text-embedding-005-fact-verification	0.769	0.695	Embedding
GEMINI-text-embedding-005-semantic-similarity	0.736	0.685	Embedding
DFIZ	0.717	0.680	Stat
DPH	0.715	0.676	Stat
OPENAI-text-embedding-3-small	0.747	0.674	Embedding
GEMINI-text-embedding-005-retrieval-document	0.750	0.670	Embedding
DLH	0.717	0.662	Stat
BM25+PL2	0.708	0.658	Stat
DPH+PL2	0.708	0.658	Stat
PL2	0.708	0.658	Stat
LEMUR	0.706	0.658	Stat
HIEMSTRA	0.708	0.657	Stat
DFIC	0.708	0.647	Stat
SBERT-all-MiniLM-L6-v3	0.686	0.646	Embedding
DFR_BM25	0.653	0.644	Stat
BM25	0.653	0.644	Stat
BM25PLUS	0.732	0.599	Stat
TFIDF	0.669	0.593	Stat
BM25-OKAPI	0.697	0.578	Stat
BM25LARGE	0.671	0.545	Stat
LLAMA3.2LARGE	0.344	0.282	LLM
DEEPSEEK-llm-67b	0.465	0.273	LLM

rank relevant documents higher across the entire result list as evaluation metrics. The key results are summarized in Table 2. Bellow we draw some conclusions from these results.

LLMs as Retrievers. Large Language Models (LLMs) demonstrated a remarkable improvement in semantic retrieval capabilities. Notably, GPT-4o-2024-08-06 achieved the best overall performance, with R@5=0.940 and MAP=0.918, significantly outperforming all other techniques. This performance indicates a deep semantic understanding of claims and an outstanding ability to locate relevant evidence, even when it is expressed implicitly or through paraphrasing. Other LLMs, such as Qwen 2.5-72B and GPT-4o-mini, also delivered strong results (MAP > 0.74), albeit with slightly lower recall. In contrast, models like DeepSeek-67B and LLAMA3.2 Large showed significantly lower performance, possibly due to limitations in their training or in how they represent truthfulness or evidential relations in retrieval tasks.

Embedding-based Models. Pretrained embedding models offered an interesting balance between efficiency and quality. The `OpenAI text-embedding-3-large` model achieved notable results (R@5=0.800, MAP=0.731), approaching the performance of some LLMs while maintaining a lower computational cost. Within this category, `Gemini-FACT_VERIFICATION` remained competitive (MAP = 0.695), confirming its suitability for contextual verification tasks. SBERT also yielded acceptable results considering its lightweight nature.

Traditional Statistical Techniques. While often considered baselines, traditional statistical retrieval methods such as BM25, PL2, TF-IDF, and DFR variants demonstrated surprisingly competitive performance, particularly when evaluating MAP. Notably, DFIZ achieved a MAP of 0.680 and DPH reached 0.676, both comparable to modern embedding models such as GEMINI-fact-verification (MAP=0.695), GEMINI-semantic-similarity (MAP=0.685), and OpenAI-embedding-3-small (MAP=0.674). In fact, DFIZ and DPH outperformed other embedding-based retrievers like GEMINI-retrieval-document

(MAP=0.670), showing that statistical methods can still offer strong baseline performance under the right conditions. These results suggest that, despite lacking semantic understanding, term-frequency-based approaches remain viable for certain fact-checking tasks—particularly when the evidence is lexically similar to the claim. Their lower computational cost and strong MAP scores make them attractive for large-scale or resource-constrained settings.

6. Checkthat 2025 Results

To address Task 3 of the CheckThat! 2025 Lab, which consists in verifying numerical and temporal claims in Spanish, we implemented and evaluated two distinct pipelines using the `gpt-4o-2024-08-06` model. This model was selected based on its outstanding retrieval performance in the preliminary experiments (MAP = 0.918, R@5 = 0.940), as shown in Section 5.

We explored two main strategies:

- **Direct classifier:** a single-stage prompt that directly predicts the veracity label (True, False, or Conflicting) given the claim and evidence.
- **Evidence collection + classification:** a two-stage pipeline where evidence is first retrieved using a semantic retriever (top-5 from BM25 preselection), and then the claim is classified based on this curated evidence.

Each strategy was evaluated using two prompting approaches:

- **Zero-shot:** the LLM receives only the task instruction.
- **Few-shot:** the LLM is provided with a small number of labeled examples before prediction.

The experiments were carried out using the OpenAI assistant configuration, with a top p value set to 1 and a temperature of 0.01. This setup minimizes the randomness of the generated responses, making the model highly deterministic by prioritizing the most probable tokens at each step, while still considering the full probability distribution due to the top-p value of 1.

We report **macro-averaged F1** [25] score on the official Spanish development set. The results are summarized in Table 3. This metric was selected as it aligns with the official evaluation criteria used by the competition organizers, ensuring consistency and comparability with the leaderboard results.

Table 3
Macro F1 scores for each classification approach and prompting configuration.

Strategy	Prompt Type	Model	Macro-F1
Direct classifier	Zero-shot	gpt-4o	0.349
Direct classifier	Few-shot	gpt-4o	0.621
Evidence + classifier	Zero-shot	gpt-4o	0.509
Evidence + classifier	Few-shot	gpt-4o	**0.672**

The few-shot evidence+classifier pipeline yielded the best result (F1 = 0.672), indicating that combining relevant evidence selection with prompting examples significantly improves performance. In contrast, the zero-shot direct classification approach underperformed, achieving only F1 = 0.349, highlighting the importance of both evidence quality and task-specific guidance.

6.1. Checkthat Submission Results

In the Checkthat 2025 Lab, we participated in Task 3 (Fact-Checking Numerical Claims) for the Spanish dataset. According to the competition rules, teams were allowed to submit only **one final run** for evaluation on the official test set.

Based on our validation results (see Table 3), we selected the **Evidence + classifier** strategy with **few-shot prompting** using the `gpt-4o-2024-08-06` model, as it achieved the best macro F1 (0.672) on the development set.

The final submission was evaluated by the organizers using macro-averaged F1, as well as class-specific F1 scores.

Table 4
ChechThat! 2025 Task 3 official test set results (Spanish).

Team	Macro F1	True F1	Conflicting F1	False F1
tsdlovehta	0.5034	0.3086	0.2707	0.9309
helenpy	0.3668	0.1729	0.0478	0.8797
UCOM_UNAM_PLN (ours)	**0.3595**	**0.1853**	**0.1490**	**0.7443**
Mohamed_Abdallah	0.2441	0.1694	0.1418	0.4211

Our approach achieved the **third-best macro F1 score** (0.3595) in the official CheckThat! 2025 Task 3 evaluation for Spanish (see Table 4). While not leading in overall ranking, our model demonstrated particularly strong performance in identifying `False` claims, with an **F1 score of 0.7443**, indicating a reliable ability to detect and reject incorrect numerical or temporal assertions. However, the F1 scores for the `True` (0.1853) and `Conflicting` (0.1490) labels were significantly lower on the test dataset. This disparity in class-wise performance suggests a particular challenge in distinguishing and handling claims that are not clearly "False", an aspect we address in more detail in the limitations and future perspectives of this work.

7. Limitations of the Approach

Despite the promising results obtained by the proposed pipeline, several limitations must be acknowledged. The computational cost associated with using large language models like `gpt-4o` could be a practical limitation for large-scale deployments or resource-constrained environments, despite the observed performance gains. This limitation highlights the need to explore smaller models in future work.

Furthermore, the effectiveness of the LLM-based components, particularly in the few-shot configuration, relies heavily on the quality and representativeness of the examples provided; poorly chosen examples can lead to suboptimal classifications or evidence extraction. The classification is highly dependent on the evidence retrieved; in cases where relevant evidence is not retrieved by BM25, the system may fail to properly assess the claim's veracity.

A crucial limitation that explains the lower F1 scores for the "True" and "Conflicting" labels lies in the classification heuristic applied. When the evidence retriever identifies relevant sentences, the current logic assigns specific priority to the verdicts of the extracted sentences. If at least one sentence is "Conflicting", the claim is classified as "Conflicting". In the absence of "Conflicting" sentences, if at least one sentence is "False", the claim is classified as "False". Only if all sentences are "True" is the claim classified as "True". This precedence implies that if sentences with both "True" and "False" verdicts are retrieved simultaneously (i.e., contradictions exist among the relevant sentences) and no sentence is explicitly labeled as "Conflicting" by the LLM, the system tends to classify the claim as "False" due to the priority given to the "False" verdict. This may result in the failure of identification of "Conflicting" or even "True" claims when the retrieved evidence is not uniformly positive.

Likewise, in cases where the retriever failed to identify relevant fragments and therefore returned no sentences, the claim was classified as False by default. This heuristic, adopted under the assumption that the absence of verifiable evidence suggests the claim is false in this context, further biases the system toward the "False" classification, contributing to the poor performance in the "True" and "Conflicting" categories and potentially reducing the overall robustness of the classification.

8. Conclusion and Perspectives for Future Work

This paper presented an effective two-step architecture for numerical fact-checking, leveraging the power of large language models within a hybrid pipeline. Our approach, combining semantic evidence retrieval with a few-shot classification strategy using the `gpt-4o-2024-08-06` model, achieved a competitive macro F1 score (0.3595) and ranked third in the CheckThat! 2025 Task 3 official test set for the Spanish dataset [1]. The results underscore the significant impact of evidence quality and task-specific guidance through few-shot prompting on fact verification performance.

For future work, several promising avenues remain open for exploration. One potential direction is to investigate alternative evidence retrieval strategies, including the full implementation and evaluation of embedding-based retrieval methods using dedicated performance metrics. Such approaches could enrich the initial evidence pool and potentially lead to improved overall system performance.

In addition, to more effectively address the identified limitations in the classification of "True" and "Conflicting" claims, there is a need to improve the classification heuristic. Specifically, we will explore a more sophisticated logic to handle situations where the retrieved sentences present contradictions (e.g., a mix of "True" and "False" verdicts without an explicitly "Conflicting" sentence). The goal is to ensure that such cases are more accurately classified as "Conflicting", better reflecting the inherent ambiguity of contradictory evidence.

Finally, given the computational cost and scalability limitations of large models like GPT-4o, exploring the potential of smaller language models is identified as a priority for future work. This line of research will include both a comparative evaluation and the application of fine-tuning techniques specifically tailored to fact-checking tasks.

Fine-tuning lighter models will significantly reduce deployment costs and also open up the possibility of training models to capture complex semantic nuances. For example, in situations where all retrieved evidence appears to be "True" or "False", but the claim contains a subtlety that requires a "Conflicting" classification, a properly fine-tuned model could learn to recognize these patterns beyond traditional heuristic rules. By incorporating such subtleties into supervised training, we expect to enhance the system's ability to produce more accurate and robust judgments.

These future work directions are considered fundamental steps to increase both the technical viability and the practical applicability of the proposed architecture, especially in contexts where efficiency is required without sacrificing accuracy.

Acknowledgments

This work was partially supported by UNAM PAPIIT project IN104424 and by the Mexican Government through SECIHTI Proyect FC-2023-G-64.

Declaration on Generative AI

During the preparation of this work, the authors used chatGPT in order to: Grammar and spelling check. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] V. Venktesh, V. Setty, A. Anand, M. Hasanain, B. Bendou, H. Bouamor, F. Alam, G. Iturra-Bocaz, P. Galuscakova, Overview of the CLEF-2025 CheckThat! lab task 3 on fact-checking numerical claims, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, CLEF 2025, Madrid, Spain, 2025.

[2] V. V, A. Anand, A. Anand, V. Setty, Quantemp: A real-world open-domain benchmark for fact-checking numerical claims, in: Proceedings of the 47th International ACM SIGIR Conference

on Research and Development in Information Retrieval, SIGIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 650–660. doi:10.1145/3626772.3657874.

[3] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. doi:10.1162/tacl_a_00454.

[4] A. Vlachos, S. Riedel, Fact checking: Task definition and dataset construction, in: C. Danescu-Niculescu-Mizil, J. Eisenstein, K. McKeown, N. A. Smith (Eds.), Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, Association for Computational Linguistics, Baltimore, MD, USA, 2014, pp. 18–22. URL: https://aclanthology.org/W14-2508/. doi:10.3115/v1/W14-2508.

[5] A. Adesokan, S. Elbassuoni, Factify: An automated fact-checker for web information, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 1546–1551. doi:10.1109/BigData62323.2024.10825147.

[6] L. Kolb, A. Hanbury, Authev-lkolb at checkthat! 2024: a two-stage approach to evidence-based social media claim verification, Faggioli et al.[22] (2024).

[7] G. E. Pianciola Bartol, A. Tommasel, Towards automated fact-checking: An exploratory study on identifying check-worthy phrases for verification, in: 2024 L Latin American Computer Conference (CLEI), 2024, pp. 1–10. doi:10.1109/CLEI64178.2024.10700241.

[8] A. Barrón-Cedeño, F. Alam, T. Chakraborty, T. Elsayed, P. Nakov, P. Przybyła, J. M. Struß, F. Haouari, M. Hasanain, F. Ruggeri, X. Song, R. Suwaileh, The clef-2024 checkthat! lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 449–458.

[9] A. Trotman, A. Puurula, B. Burgess, Improvements to bm25 and language models examined, in: Proceedings of the 19th Australasian Document Computing Symposium, ADCS '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 58–65. doi:10.1145/2682862.2682863.

[10] Kocabaş, B. T. Dinçer, B. Karaoğlan, A nonparametric term weighting method for information retrieval based on measuring the divergence from independence, Information Retrieval 17 (2013) 153–176. doi:10.1007/s10791-013-9225-4.

[11] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, G. Gambosi, Fub, iasi-cnr and university of tor vergata at trec 2007 blog track, 2007.

[12] G. Amati, Frequentist and bayesian approach to information retrieval, 1970, pp. 13–24. doi:10.1007/11735106_3.

[13] B. He, I. Ounis, Term frequency normalisation tuning for bm25 and dfr models, in: D. E. Losada, J. M. Fernández-Luna (Eds.), Advances in Information Retrieval, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 200–214.

[14] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[15] J. Perea-Ortega, M. García-Cumbreras, M. García-Vega, L. López, Comparing several textual information retrieval systems for the geographical information retrieval task, volume 5039, 2008, pp. 142–147. doi:10.1007/978-3-540-69858-6_15.

[16] D. Hiemstra, Using language models for information retrieval, 2001.

[17] OpenAI, text-embedding-3: Openai embedding models, 2024. URL: https://platform.openai.com/docs/guides/embeddings, accessed: 2025-05-28.

[18] G. Research, Generalizable embeddings from gemini, arXiv preprint arXiv:2503.07891 (2025). URL: https://arxiv.org/abs/2503.07891.

[19] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.

[20] OpenAI, GPT-4 technical report, CoRR abs/2303.08774 (2023). doi:10.48550/ARXIV.2303.08774. arXiv:2303.08774.

[21] Qwen, et al., Qwen2.5 technical report, 2025. URL: https://arxiv.org/abs/2412.15115. arXiv:2412.15115.

[22] A. G. et al., The llama 3 herd of models, 2024. URL: https://arxiv.org/abs/2407.21783. arXiv:2407.21783.

[23] D.-A. et al., Deepseek llm: Scaling open-source language models with longtermism, 2024. URL: https://arxiv.org/abs/2401.02954. arXiv:2401.02954.

[24] A. Yates, R. Nogueira, J. Lin, Pretrained transformers for text ranking: Bert and beyond, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 2666–2668. doi:10.1145/3404835.3462812.

[25] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (2002) 1–47. doi:10.1145/505282.505283.