LIS at CheckThat! 2025: Multi-Stage Open-Source Large Language Models for Fact-Checking Numerical Claims

Notebook for the CheckThat! Lab at CLEF 2025

Quy Thanh Le[1,*], Ismail Badache[1], Aznam Yacoub[2] and Maamar El Amine Hamri[1]

[1]Aix Marseille Université, CNRS, LIS, Marseille, France
[2]University of Windsor, Canada

Abstract
The fast and large-scale spread of information through social networks and digital platforms has become an important global issue for today's societies, making automated fact-checking necessary. This paper presents the contribution of the R2I[1] and MoFED[2] teams from the LIS Laboratory[3] to the task of automated claim relevance estimation, in the context of the 2025 CheckThat! Lab, specifically Task 3: Fact-Checking Numerical Claims[4]. This task focuses on the verification of claims expressed in Arabic, English, and Spanish, particularly those involving numerical data or temporal references. In this study, we explore the effectiveness of recent open-source large language models (LLMs), such as Mistral[5] and Qwen[6], for automating the fact-checking of numerical claims. We propose a two-stage pipeline that incorporates these LLMs into the fact-checking process: evidence retrieval and veracity prediction. First, we employ the QwQ-32B model to automatically generate questions from each claim, guiding the retrieval of relevant evidence from the corpus provided for Task 3. Second, we fine-tune the Mistral-Small-24B-Instruct-2501 model using the LoRA (Low-Rank Adaptation) technique to predict the veracity of each claim. This hybrid approach is designed to enhance both the performance and efficiency of the fact-checking pipeline. Despite variations in performance across languages, our method achieved outstanding results, ranking first in all 3 languages : Spanish, English and Arabic. The multilingual nature of the datasets played a crucial role in improving the generalizability of claim validation across linguistic contexts. Our approach obtained macro-F1 scores of 0.503 for Spanish, 0.595 for English and an exceptional 0.960 for Arabic, significantly outperforming the second-best Arabic score of 0.635. These results not only underscore the efficacy of leveraging open-source LLMs for fact-checking, but also contribute to ongoing research in claim detection. They further highlight the importance of language-specific adaptations and the potential of multilingual strategies in the development of robust, automated fact-checking systems.

Keywords
NLP, Automated Fact-checking, Numerical claims, Large Language Models, Multilingual datasets

1. Introduction

The issue of Big Data often refers to a situation where the speed at which data is generated and disseminated far exceeds current computational and processing capacities. This results in a gap between the massive production of data and the ability to analyze or exploit it effectively in real

[1]Recherche d'Information et Interactions
[2]Modèles et Formalismes à Evénements Discrets
[3]Laboratoire d'Informatique et des Systèmes
[4]https://checkthat.gitlab.io/clef2025/task3/
[5]https://chat.mistral.ai/
[6]https://chat.qwen.ai/

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

*Corresponding author.
quy-thanh.le@lis-lab.fr (Q. T. Le); ismail.badache@lis-lab.fr (I. Badache); aznam.yacoub@uwindsor.ca (A. Yacoub); amine.hamri@lis-lab.fr (M. E. A. Hamri)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

time. A similar phenomenon occurs in the field of information verification: the rapid proliferation of digital content, particularly on social media, makes the swift and systematic validation of circulating claims extremely difficult [1]. In this context, the implementation of robust mechanisms to assess the veracity of information has become an increasingly critical challenge. Moreover, the speed at which information circulates, often with viral dynamics, surpasses the capacity of verification processes, allowing misinformation to take hold before corrections can be made. This situation highlights the urgent need to develop effective tools and strategies to address this major informational challenge, and among the possible solutions is fact checking [1].

Fact-checking techniques, initially developed within the field of journalism, aim to evaluate the veracity of information by systematically comparing claims with verified and trustworthy sources. In recent years, these techniques have been increasingly adapted and enhanced through computational approaches. Automated fact-checking generally follows a standardized three-step pipeline: claim detection, evidence retrieval, and verdict prediction [2]. Fact-checking can be performed either manually or automatically. Manual fact-checking, typically carried out by trained experts or journalists, is known for its high reliability and contextual sensitivity. However, it is inherently limited in terms of scalability and response time. For instance, platforms like Snopes[1] manually verify viral claims, often requiring several hours to a full day to process a single claim [3]. These temporal constraints are especially problematic given that false information tends to spread significantly faster and more broadly than truthful content, particularly on social media platforms [4]. This growing asymmetry between the pace of misinformation and the capacity for human-led verification highlights the pressing need for automated fact-checking systems. Such systems are designed to identify claims requiring validation and to assess their credibility by retrieving relevant evidence and determining whether the claims are supported, contradicted, or unverifiable due to insufficient information [5]. The development of robust, scalable, and accurate automated fact-checking methods is therefore crucial to addressing the challenges posed by the rapid and widespread diffusion of false information in contemporary digital ecosystems.

However, automating fact-checking presents substantial challenges, especially when dealing with complex claims that require nuanced reasoning and contextual understanding. This complexity is further exacerbated when claims include numerical information, which are often perceived as more credible due to the numeric-truth effect—a cognitive bias where numbers enhance perceived accuracy [6]. Recent research has shown that verifying numerical claims is more difficult than verifying non-numerical ones [7, 8]. For example, the social media claim that "Pfizer admitted its COVID-19 vaccine is only 12% effective, not 95% as previously stated" is a misinterpretation of Pfizer's briefing document submitted to the FDA in December 2020, prior to the issuance of the Emergency Use Authorization (EUA)[2], and has caused unnecessary public concern. As a result, developing methods to automatically verify numerical claims has become an essential area of investigation in fact-checking research.

In this paper, we present our participation in CheckThat! 2025 Task 3: Fact-Checking Numerical Claims[3], a challenge that focuses on verifying claims involving numerical quantities and temporal expressions. The main objective of this task is to assess the veracity of claims containing explicit or implicit quantitative or temporal information by classifying them as *True*, *False*, or *Conflicting*, based on evidence retrieved from a predefined corpus. To address this task, we propose a two-stage fact-checking framework specifically tailored to the challenges posed by numerical and temporal claims. The first stage

[1] https://www.snopes.com/
[2] https://www.factcheck.org/2022/05/scicheck-pfizer-documents-show-vaccine-is-highly-effective
[3] https://checkthat.gitlab.io/clef2025/task3/

consists of an evidence retrieval module, where we automatically generate natural language questions from each claim using the multilingual large language model (LLM) QWQ-32B. These questions guide the retrieval of the most relevant supporting or contradicting documents from the corpus. In the second stage, we implement a veracity prediction module, in which the model Mistral-Small-24B-Instruct-2501 is fine-tuned on the task's training data to classify the claims based on the retrieved evidence. Our study builds on a key hypothesis shaped by open-source LLM democratization: **Are LLMs capable of performing accurate fact-checking on the dataset provided in CheckThat! 2025 Task 3?**

The remainder of this paper is organized as follows. Section 2 provides a short review of some related work, situating our contribution within the current research on fact-checking. Section 3 presents the methodology, detailing the LLMs experimented with and the prompts employed. Section 4 discusses the experimental results, highlighting key findings and observed limitations. Finally, Section 5 concludes the paper by summarizing the main insights and outlining directions for future research.

2. Some Related Works

Although an increasing number of studies have been conducted to enhance fact-checking methods [1, 9], the majority of existing work primarily focuses on verifying textual claims using either structured or unstructured data sources [10, 11, 12]. In contrast, research that specifically addresses the verification of numerical claims remains relatively scarce.

To the best of our knowledge, only a few prior works have directly addressed the fact-checking of numerical claims. For example, Wallat et al. [13] present a focused evaluation of LLMs ability to verify temporal factual claims through a dedicated Temporal Fact Checking task. They use a dataset of 4196 manually verified claims sourced from fact-checking websites [7] to assess whether models can classify statements as *True*, *False*, or contradictory. This task differs from traditional question-answering by requiring critical judgment rather than mere fact retrieval. The results reveal limited performance across all models, with classification accuracy ranging from 29% (Llama 3.1) to 74.7% (Qwen 2.5). Surprisingly, even highly capable models like GPT-4 underperform, often refusing to answer when uncertain—a behavior interpreted as cautious calibration rather than lack of knowledge. This part of the study demonstrates that while LLMs may store temporal facts, they often lack the reasoning ability or confidence to verify them reliably, especially when temporal specificity is required.

Building on the state of the art and existing literature on Fact-Checking Claims, while adopting a somewhat different perspective regarding the formulation of our research question, We investigate the capability of open-source LLMs to accurately perform fact-checking on the specific dataset provided in CheckThat! 2025 Task 3.

3. Open source LLMs in Fact-Checking process

Our goal is to apply large open-source language models to two key stages of the fact-checking process, evidence retrieval and veracity prediction, for predicting the label of numerical claims. This section introduces the dataset, the Open source LLMs Used in Experiments, and the pipeline of the proposed method followed by our models to fact-check claims. Since we use a shared pipeline across all three languages, we focus on the English setup here; the setups for Arabic and Spanish are analogous, with prompts adapted to each language while maintaining the same format.

3.1. Dataset

The CheckThat! Lab 2025 task 3 [14] provided participants with datasets in English, Spanish and Arabic. Each dataset includes training (*Train*), development (*Dev*), and test (*Test*) splits, with the test set reserved for final submissions. Additionally, the organizers provided a collection of evidence used to verify all claims for each language (see Table 1).The distribution of claims shows that the Spanish and Arabic datasets are significantly smaller than the English dataset (see Table 2). The training (*Train*) and development (*Dev*) splits show a highly imbalanced label distribution across the 3 labels, with the *False* label accounting for the majority in both English and Spanish, whereas the Arabic dataset exhibits a more balanced class distribution. However, the Arabic dataset includes only 2 labels, *True* and *False*, compared to 3 labels in English and Spanish (see Table 3).

Table 1
Number of evidences in the corpus by language.

Language	Evidences collection
English	426741
Spanish	10101
Arabic	5022

Table 2
Distribution of claims by language and dataset split.

Language	Train	Dev	Test	Total
English	9935	3084	3656	16675
Spanish	1506	377	1806	3689
Arabic	2191	587	482	3260

Table 3
Class percentage of claims by language and dataset split.

Language	Train			Dev		
	True	False	Conflicting	True	False	Conflicting
English	18%	58%	24%	20%	58%	22%
Spanish	8%	79%	13%	8%	79%	13%
Arabic	45%	55%	-	46%	54%	-

3.2. Open source LLMs Used in Experiments

Open-source LLMs provide significant advantages in terms of cost-effectiveness, transparency and community collaboration. Therefore, we utilize two open-source LLMs, along with an embedding model developed upon LLM foundations, as outlined in Table 4.

Table 4
Utilized Open-Source Models.

Model	Number of Parameters	Release Date
QwQ-32B	32 billions	March 2025
Linq-Embed-Mistral	7 billions	Jun 2024
Mistral-Small-24B-Instruct-2501	24 billions	Jan 2025

QwQ-32B[4] is the reasoning-focused model from the Qwen series. Unlike conventional instruction-tuned models, QwQ-32B demonstrates enhanced capabilities in reasoning and complex problem-solving tasks. It performs competitively with state-of-the-art models such as DeepSeek-R1 and o1-mini, making

[4]https://huggingface.co/Qwen/QwQ-32B

it well-suited for fact-checking tasks that require logical inference. This model is specifically employed to perform the question generation task as part of our inference pipeline.

Linq-Embed-Mistral[5] builds upon the foundations of E5-mistral-7b-instruct and Mistral-7B-v0.1. It demonstrates strong retrieval performance, ranking 2nd on the MTEB benchmark leaderboard [15] with a score of 60.2. This model is specifically used for embedding and retrieval tasks within our pipeline.

Mistral Small 3[6] (also referred to as 2501) is a compact model with 24B parameters, making it one of the most capable models in the sub-70B category. Developed by Mistral AI, it optimizes transformer-based architectures for language tasks, combining high performance with computational efficiency. Within our inference pipeline, this model is specifically responsible for assessing the veracity of the claim based on the retrieved evidences, claim and generated questions.

3.3. Methodology

Figure 1 presents our proposed pipeline for fact-checking, which follows a two-stage architecture: evidence retrieval and veracity prediction. The input to the system consists of a claim to be verified and a corpus of evidence provided by the task organizers, which serves as a knowledge base containing relevant information for evaluating claims. In the first stage, the system retrieves the top-k most relevant pieces of evidence from the corpus by applying a decomposition method to the input claims. In the second stage, veracity prediction, the input includes the retrieved evidence, the original claim, and the generated questions. These are fed into a fine-tuned LLM that has been pre-trained for the veracity prediction task using the training dataset to produce a final verdict, reflecting the claim's degree of validity. To ensure consistency, the same pipeline described above is used to retrieve evidences for the training, development, and test sets. We do not use the generated questions, retrieved evidences, or golden labels provided by the task organizers at any point in this process.

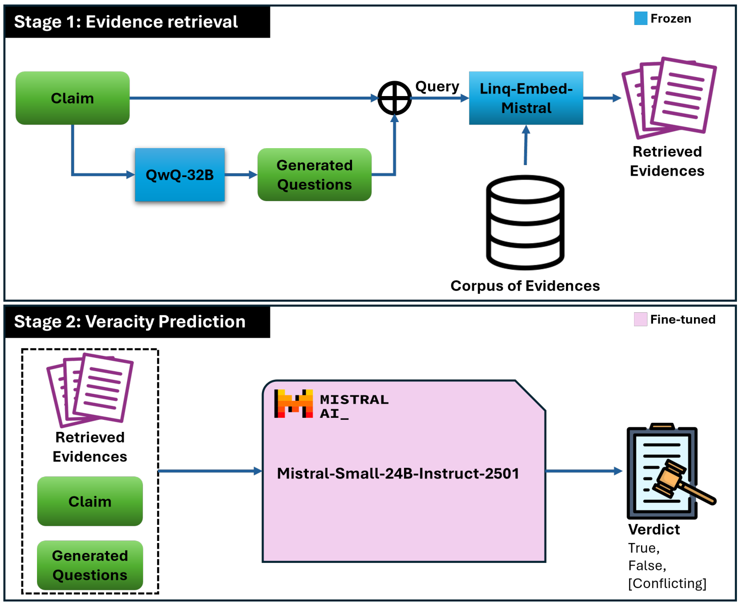

Figure 1: Two-stage inference pipeline for our claim verification system.

[5]https://huggingface.co/Linq-AI-Research/Linq-Embed-Mistral
[6]https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501

Although the Arabic dataset contains only two labels (*True* and *False*), a three-label prompt structure, as shown in Figure 4, is maintained to ensure consistency across different languages. While the prompt suggests the possibility of three output labels, the fine-tuned model, trained solely on two-label data, produces only two labels during evaluation on the development set. This behavior does not result in any errors or require additional normalization, and the model still achieves high performance.

Table 5 presents our model configurations (LIS system) in comparison to the baseline system provided by the organizers. Key differences lie in the evidence retrieval stage, where we employ claim-augmented retrieval queries (combining the original claim with generated questions), as opposed to the baseline system which uses question-only queries. For retrieval, we use Linq-Embed-Mistral, which selects the most relevant evidence based on cosine similarity, rather than relying on BM25 followed by reranking with paraphrase-MiniLM-L6-v2 as in the baseline. For question generation, our system utilizes the QwQ-32B model, while the baseline employs GPT-3.5. In the veracity prediction stage, our system leverages the powerful Mistral-Small-24B-Instruct-2501, whereas the baseline is based on FinQA-Roberta-Large.

Table 5
Model configurations

System	Evidence Retrieval		Question Generation	Reranking	Veracity Prediction
	Query	Model			
Baseline	generated questions	BM25	GPT-3.5	paraphrase-MiniLM-L6-v2	FinQA-Roberta-Large
LIS	claim + generated questions	Linq-Embed-Mistral	QwQ-32B	-	Mistral-Small-24B-Instruct-2501

3.3.1. Evidence Retrieval

As presented in Figure 1, the claim verification system is based on a two-stage inference pipeline. The first stage consists of retrieving relevant evidence from the corpus of evidences to either support or refute a given claim. Inspired by recent advances in generative retrieval methods [16, 17], we employ an instruction-following language model to generate questions based on the input claims. The prompt used during this generation process is illustrated in Figure 2.

Once the questions are generated, each question is concatenated with its corresponding claim to form a query. This query is then embedded using our embedding model (Linq-Embed-Mistral) and compared against the corpus to retrieve the most relevant piece of evidence (top-1 retrieval) based on cosine similarity. This approach ensures that the selected evidence is contextually aligned with both the claim and the generated question. An example of this retrieval process is illustrated in Figure 3.

3.3.2. Veracity Prediction

In the same vein, and as shown in Figure 1, the second stage handles the veracity prediction task based on the generated question, the retrieved evidence, and the original claim as inputs to predict the veracity of the claim. We employ an instruction-following LLM, specifically the fine-tuned Mistral-Small-24B-Instruct-2501 model, to perform this task. This model predicts the final veracity label (*True,*, *False* or *Conflicting*) based on the combined context. The prompt structure used for veracity prediction is illustrated in Figure 4.

Figure 2 illustrates a concrete example of an instruction prompt alongside its generated output for the task of question generation. In this figure, the instruction is highlighted in bold text, providing clear guidance to the model. The italicized text presents the claim or statement on which the question is based. Finally, the question produced by the model is shown in blue, demonstrating how the model interprets and transforms the input prompt into a relevant query. This visual representation helps to clarify the interaction between the instruction, the claim, and the model's generated response.

 # Instructions
You are a fact-checker. Your general motivation is to verify a given claim. You are at the beginning of the fact-checking process, meaning you have just received the claim, optionally with some additional metadata (such as the date of the claim or the author), if available. Your task now is to prepare the fact-check. That means:
1. Begin with an interpretation of the claim. As part of the interpretation, list the key points of the claim as a list of reformulated sub-claims.
2. Then, analyze what information is missing.
3. Finally, present a complete, numbered list of 5 questions: These are questions that explore the truthfulness of the claim and that we need to answer in order to factually verify the claim.

IMPORTANT:
Follow these rules:
*** Phrase each question so that it can be understood independently and without additional context. Be explicit and do not use pronouns or generic terms in place of names or objects.**
*** Enclose each question in backticks like this.**
 # Examples
Claim: "The new Food Bill in New Zealand bans gardening"
Good Question: "Did the New Zealand government pass a food bill that restricted gardening activities for its citizens?"
Bad Question: "Did the government pass a bill?"
Bad Question: "Did the bill restrict activities?"
 # The Claim
french president emmanuel macron is around 25 years younger than his wife, brigitte, and they started dating when he was a teenager and she was one of his teachers. her daughter, laurence, was once one of his classmates, and later became his stepdaughter.

Output | Generated questions:
"What is the exact age difference between French President Emmanuel Macron and his wife Brigitte Macron?"
"Did Laurence, Brigitte Macron's daughter, legally become Emmanuel Macron's stepdaughter through the marriage of Brigitte Macron and Emmanuel Macron?"
"Did Emmanuel Macron and Brigitte Macron begin their romantic relationship while Macron was still a teenager?"
"Was Brigitte Macron serving as a teacher at the same educational institution where Emmanuel Macron was a student when their relationship began?"
"Was Laurence, Brigitte Macron's daughter from a previous relationship, enrolled as a student in the same school class as Emmanuel Macron before he married Brigitte?"

Figure 2: An example of an instruction prompt and its output for question generation. The bold text represents the instruction, the italic text is the claim, and the blue text indicates the model's generated question

Figure 3 illustrates the process of query formation by integrating a claim with a generated question aimed at retrieving relevant evidence. In this example, the bold text represents the instruction guiding the task, while the italic text shows the claim under verification. The gray text displays the most pertinent evidence retrieved from the data source, and the blue text highlights the questions generated by the model to support the evidence search. This structured approach facilitates targeted evidence gathering for fact-checking purposes.

Query:*french president emmanuel macron is around 25 years younger than his wife, brigitte, and they started dating when he was a teenager and she was one of his teachers. her daughter, laurence, was once one of his classmates, and later became his stepdaughter.* What is the exact age difference between French President Emmanuel Macron and his wife Brigitte Macron?

Output | Relevant evidence 1:
Emmanuel, who became president in 2017, and Brigitte first met when the now-president was in high school. In a November 2023 interview with Paris Match — translated by The Independent — Brigitte shared how she never thought of entertaining a relationship with her student, who was 15 at the time. Brigitte and Emmanuel met when he was a student at Le Providence, a Catholic school in northern France. Brigitte worked as a drama teacher and was 39 years old at the time, while Emmanuel was 15. The president's parents originally thought he was seeing Brigitte's daughter, but when they found out he had feelings for the teacher, they sent him to boarding school. French President Emmanuel Macron and his wife, Brigitte, have been married for nearly two decades, but their relationship dates back even further.

Query:*french president emmanuel macron is around 25 years younger than his wife, brigitte, and they started dating when he was a teenager and she was one of his teachers. her daughter, laurence, was once one of his classmates, and later became his stepdaughter.* Was Brigitte Macron serving as a teacher at the same educational institution where Emmanuel Macron was a student when their relationship began?

Output | Relevant evidence 2:
Emmanuel Macron's wife and former teacher thought he would "fall in love with someone his [own] age" after he was moved from the school where she taught him. Mrs Macron said her older siblings used to joke about the gossip. She divorced from her estranged husband, André-Louis Auzière, a banker, in 2006 and married Mr Macron a year later, when he was a young civil servant. The French president was 15 when he fell for his married drama teacher, Brigitte Auzière, then 40, at the Catholic Providence school in Amiens in the early 1990s. Her daughter Laurence was the future French president's classmate.

...

Query:*french president emmanuel macron is around 25 years younger than his wife, brigitte, and they started dating when he was a teenager and she was one of his teachers. her daughter, laurence, was once one of his classmates, and later became his stepdaughter.* Question k and $k \in \{3, 5\}$

Output | Relevant evidence k:
...

Figure 3: Example of query formation by combining a claim with a generated question to search for relevant evidences. The bold text is the instruction, the italic text is the claim, the gray text is the most relevant evidence retrieved, and the blue text denotes the generated questions.

Figure 4 presents an example of an instruction prompt and output for veracity prediction. The bold text corresponds to the instruction given to the model, the italic text is the claim to be verified, while the gray text represents the extracted evidence deemed relevant. Finally, the blue text includes the questions generated by the model as well as its final prediction regarding the claim's veracity. This figure clearly illustrates the complete process of automatic claim verification based on evidence and generated questions.

 # Instructions
You are a fact-checker. You have done a fact-check to verify a Claim based on the provided question-answer pair evidence.
Your task is to predict the verdict of a claim based on the provided question-answer pair evidence whether the Claim is one of the labels: 'True', 'False', 'Conflicting'. Do this by following:
- Respond "True" only if the relevant evidence fully or almost fully supports and verifies the claim as correct.
- Respond "False" if:
- The relevant evidence contradicts or disproves the claim.
- The claim is misleading based on the relevant evidence.
- The evidence is too weak or insufficient to support the claim.
- Respond "Conflicting" if the evidence is ambiguous, incomplete, or inconclusive, making it impossible to determine if the claim is fully true or false.
Always adhere to the following rules:
- Use information only from the recorded evidence: Avoid inserting information that is not implied by the evidence. You may use commonsense knowledge, though.
- Avoid repeating yourself.
Claim: *french president emmanuel macron is around 25 years younger than his wife, brigitte, and they started dating when he was a teenager and she was one of his teachers. her daughter, laurence, was once one of his classmates, and later became his stepdaughter.*

Q1: "What is the exact age difference between French President Emmanuel Macron and his wife Brigitte Macron?"
A1: Emmanuel, who became president in 2017, and Brigitte first met when the now-president was in high school. In a November 2023 interview with Paris Match — translated by The Independent — Brigitte shared how she never thought of entertaining a relationship with her student, who was 15 at the time. Brigitte and Emmanuel met when he was a student at Le Providence, a Catholic school in northern France. Brigitte worked as a drama teacher and was 39 years old at the time, while Emmanuel was 15. The president's parents originally thought he was seeing Brigitte's daughter, but when they found out he had feelings for the teacher, they sent him to boarding school. French President Emmanuel Macron and his wife, Brigitte, have been married for nearly two decades, but their relationship dates back even further.

Q2: "Was Brigitte Macron serving as a teacher at the same educational institution where Emmanuel Macron was a student when their relationship began?"
A2: Emmanuel Macron's wife and former teacher thought he would "fall in love with someone his [own] age" after he was moved from the school where she taught him. Mrs Macron said her older siblings used to joke about the gossip. She divorced from her estranged husband, André-Louis Auzière, a banker, in 2006 and married Mr Macron a year later, when he was a young civil servant. The French president was 15 when he fell for his married drama teacher, Brigitte Auzière, then 40, at the Catholic Providence school in Amiens in the early 1990s. Her daughter Laurence was the future French president's classmate.

...

Output | Verdict: True

Figure 4: Example of instruction prompt and output for veracity prediction.The bold text indicates the instruction, the italic text is the claim, the gray text shows the retrieved evidence, and the blue text includes both the generated question and the model's veracity prediction.

4. Experimental Results

This section presents the results of our implementation, including the hyperparameter configurations used to fine-tune Mistral-Small-24B-Instruct-2501 for veracity prediction and to generate questions with QwQ-32B, as well as the environment setup and our experiment results and discussion.

4.1. Implementation

The hyperparameters used for fine-tuning the LLMs in our experiments are summarized in Table 6. To enable efficient fine-tuning, we employed the Low-Rank Adaptation (LoRA) technique [18]. The hyperparameters specific to the question generation process are provided in Table 7, while those for veracity prediction using the fine-tuned model are detailed in Table 8. We also updated the evidence corpus with the language-specific documents provided by the task organizers during the test phase to ensure the retrieval of the most relevant evidence for each claim. All experiments were conducted on the LIS cluster, equipped with NVIDIA A100 GPUs (80GB). The evidence retrieval phase took approximately 8 hours, while the question generation phase required around 2 days. Veracity prediction across all three languages was completed in roughly 3 hours. To address potential cases where the language model produced abnormal outputs lacking a final label in the required format, we implemented a fallback mechanism whereby the generation process was repeated up to five times. At each iteration, the temperature was incrementally increased by 0.1—starting from an initial value of 0.1—to promote more diverse and correctly formatted outputs, as illustrated in Table 8. Nevertheless, during evaluation on the test set, no such abnormal outputs were observed. For the comparison experiments, we trained our models on the training set and evaluated them on the development set. These models were subsequently used for the final submission to the shared task.

Table 6
Hyperparameters used for Fine-tuning in veracity prediction

Parameter	Value
Epochs	2
Training batch size	2
Gradient accumulation steps	4
Optimizer	AdamW 8-bit
Learning rate	2e-4
Weight decay	0.01
Warmup step	5
Temperature	0.1
Lora Alpha	16
Lora dropout	0.1
Lora rank	64

Table 6 summarizes the hyperparameters used for fine-tuning a model for veracity prediction. The training was conducted over 2 epochs with a batch size of 2 and gradient accumulation steps of 4, using the AdamW 8-bit optimizer. Key settings include a learning rate of 2e-4, weight decay of 0.01, and a warmup of 5 steps. A temperature of 0.1 was applied for output control. LoRA-based fine-tuning was used with alpha set to 16, dropout to 0.1, and rank to 64, enabling efficient and scalable parameter adaptation.

Table 7
Hyperparameters used for generating questions

Parameter	Value
Max token length	6000
Temperature	0.6
Top p	0.9
Top k	30
Min p	0.1

Table 8
Hyperparameters used for veracity prediction

Parameter	Value
Max token length	500
Temperature	0.3
Top p	0.9
Top k	10

Tables 6 and 7 present the decoding hyperparameters used for two distinct tasks: question generation and veracity prediction, respectively. For question generation (Table 6), the model allows a maximum token length of 6000, with a temperature of 0.6, top-p (nucleus sampling) set to 0.9, top-k set to 30, and a minimum probability threshold (min p) of 0.1, favoring diverse yet coherent outputs. In contrast, veracity prediction (Table 7) uses a more constrained setting with a maximum token length of 500, temperature of 0.3, top-p of 0.9, and a smaller top-k of 10, reflecting a more focused and deterministic generation process suitable for classification or decision-making tasks.

4.2. Results and Discussions

4.2.1. English datasets

Table 9 presents the results obtained from various configurations evaluated on the English dataset of CheckThat! 2025 Task 3, including both the development and test sets. The evaluation is based on two scenarios regarding the number of generated questions (column # questions) per claim—3 questions vs. 5 questions—using the model Mistral-Small-24B-Instruct-2501, which was fine-tuned on the training set. These results are also compared against a baseline using the FinQA-RoBERTa-Large NLI model on the development set. The evaluation metrics include macro-F1 and class-wise F1 scores for each class.

Table 9
Performance of the proposed model scenarios compared to the baseline on the Dev and Test set partitions of the English datasets

Partition	Method	# questions	Macro F1	True F1	False F1	Conflicting F1
Dev	[Baseline] FinQA-Roberta-Large	3	0.5815	0.5058	0.7914	0.4472
	Mistral-Small-24B-Instruct-2501	3	0.6130	0.5550	0.8470	0.4380
	Mistral-Small-24B-Instruct-2501	5	0.6110	0.5560	0.8390	0.4380
Test	Mistral-Small-24B-Instruct-2501	3	0.5954	0.6332	0.8280	0.3250

Results on the development set show that our proposed model achieved nearly equivalent performance in both the 3-question and 5-question scenarios, with the 3-question setup performing slightly better by about 0.2%. Notably, both scenarios outperformed the baseline by approximately 3%.

Due to the task requirement of selecting only one best-performing model for the final evaluation on the test set, we submitted the proposed 3-question model. This model achieved a macro-F1 score of 59.54%, ranking first, ahead of all other participating teams. The full leaderboard is published on the official CLEF website[7]. Compared to the second-ranked team, our system outperformed by approximately 3%, and exceeded the performance of the lowest-ranked team by about 24%. Notably, this performance was approximately 1.5% lower than its result on the development set. However, we were unable to compare this result against the baseline on the test set, as the official baseline results for Task 3 have not yet been released.

4.2.2. Spanish and Arabic datasets

Table 10 presents the results obtained from various configurations evaluated on the Spanish and Arabic datasets of the CheckThat! 2025 Task 3, covering both the development and test sets. The evaluation was conducted based on two scenarios regarding the number of generated questions per claim—3 questions versus 5 questions—using the Mistral-Small-24B-Instruct-2501 model, which was fine-tuned on the training set. The evaluation metrics include macro-F1 and class-wise F1 scores for each label. However, unlike the English dataset, the authors did not provide baseline model results for comparison.

Table 10
Performance of the proposed model scenarios compared to the baseline on the Dev and Test set partitions of the Spanish and Arabic datasets

Partition	Language	# questions	Macro F1	True F1	False F1	Conflicting F1
Dev	Spanish	3	0.5740	0.4090	0.9240	0.3900
	Spanish	5	0.4400	0.2920	0.9020	0.1400
	Arabic	3	0.9600	0.9560	0.9640	-
	Arabic	5	0.9500	0.9450	0.9540	-
Test	Spanish	3	0.5034	0.3086	0.9309	0.2707
	Arabic	3	0.9615	0.9552	0.9679	-

On the development set, the macro-F1 scores indicate that, similar to the English results, Arabic achieved nearly equivalent performance in both the 3-question and 5-question scenarios, with the 3-question configuration slightly outperforming by about 1%. Notably, the 3-question setup yielded almost perfect performance, reaching approximately 96%. For Spanish, using 3 questions resulted in a performance improvement of over 10% compared to the 5-question scenario, suggesting that a smaller number of questions may help retrieve more accurate and relevant evidence.

Due to the task requirement that only one best-performing model per language could be submitted for test set evaluation—as was the case with the English dataset—we submitted the 3-question configuration.

In the Spanish, the model attained a macro-F1 score of 59.54%, reflecting a decrease of roughly 7% compared to its performance on the development set. Despite this, it secured first place, outperforming all other competing teams. The complete leaderboard is available on the official CLEF website[8]. Our system surpassed the second-place team by approximately 13% and outperformed the lowest-ranked team by around 25%.

For Arabic, although a three-label prompt was used as mentioned earlier in Section 3.3, the results on the test set consistently produced only two labels without any errors or the need for normalization, the model achieved a macro-F1 score of 96.15%, showing a slight improvement of about 0.15% relative

[7]https://codalab.lisn.upsaclay.fr/competitions/22699#results
[8]https://codalab.lisn.upsaclay.fr/competitions/22823#results

to its development set results. It also ranked first among all participants. The full rankings can be found on the official CLEF website[9]. Compared to the runner-up, our model exceeded their performance by roughly 33%, and outperformed the lowest-ranked team by approximately 60%.

We chose to generate three questions per claim based on consistent empirical results across Arabic, English, and Spanish, showing this configuration to be the most effective. This finding aligns with [7], who tested multiple settings (1, 3, 5, and 7 questions) and also reported optimal performance with three. This convergence reinforces the robustness of our approach.

5. Conclusion

This paper presents the experiments conducted by the LIS team for CheckThat! 2025 Task 3, which focuses on verifying numerical claims in English, Spanish, and Arabic. We investigated the application of large open-source language models in the two key stages of the fact-checking pipeline: evidence retrieval and veracity prediction. Our proposed pipeline, which integrates instruction-following LLMs and effective fine-tuning strategies such as LoRA. Our model consistently applies the same methodology across all three languages and is the only team to fully participate in all three. The model achieved strong performance; notably, the fine-tuned model on the training data secured first place on the official leaderboard, outperforming submissions from 19 participating teams across all languages. These results highlight the potential of open-source LLMs in multilingual fact-checking tasks. Furthermore, our findings emphasize that fine-tuning Mistral-Small-24B-Instruct-2501 model, yields significantly better performance compared to NLI models like FinQA-Roberta-Large, which was used as the baseline on the English development set. This demonstrates the benefit of leveraging both model scale and task-specific adaptation in complex fact-checking scenarios involving numerical claims.

However, many potentially promising open-source LLMs remain unexplored in this experiment. The number of test questions is still limited to only one configuration: 3 questions. Additionally, the current fine-tuning approach has been limited to monolingual language models. In future work, we plan to conduct experiments on a broader range of open-source LLMs as well as with more diverse question configurations, and to explore fine-tuning multilingual models, which may yield better results compared to monolingual ones. Furthermore, although the Arabic dataset includes only two labels (True and False), the current prompt structure uses three possible verdicts to ensure consistency across languages. This design choice may lead the model to generate non-applicable verdicts for certain Arabic claims, potentially affecting system behavior. The impact of this discrepancy will be examined to determine whether normalization techniques—such as mapping the model's output to the available binary labels—can enhance reliability and alignment with the ground truth.

Acknowledgments

This work was supported by the Eiffel Excellence Scholarship awarded to Quy Thanh Le by the French government, as well as by the R2I and MoFED teams of the LIS Laboratory.

[9]https://codalab.lisn.upsaclay.fr/competitions/22699#results

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4 for grammar and spelling checks, as well as for paraphrasing and rewording. After using these tools, the author(s) reviewed and edited the content as needed, and take full responsibility for the publication's content.

References

[1] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. URL: https://aclanthology.org/2022.tacl-1.11/. doi:10.1162/tacl_a_00454.

[2] X. Zeng, A. S. Abumansour, A. Zubiaga, Automated fact-checking: A survey, Language and Linguistics Compass 15 (2021) e12438.

[3] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, M. Tremayne, Claimbuster: the first-ever end-to-end fact-checking system, Proc. VLDB Endow. 10 (2017) 1945–1948. URL: https://doi.org/10.14778/3137765.3137815. doi:10.14778/3137765.3137815.

[4] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (2018) 1146–1151. URL: https://www.science.org/doi/abs/10.1126/science.aap9559. doi:10.1126/science.aap9559. arXiv:https://www.science.org/doi/pdf/10.1126/science.aap9559.

[5] R. Panchendrarajan, A. Zubiaga, Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research, Natural Language Processing Journal 7 (2024) 100066. URL: http://dx.doi.org/10.1016/j.nlp.2024.100066. doi:10.1016/j.nlp.2024.100066.

[6] N. Sagara, E. Peters, Consumer understanding and use of numeric information in product claims, in: D. R. Deeter-Schmelz (Ed.), Proceedings of the 2010 Academy of Marketing Science (AMS) Annual Conference, Springer International Publishing, Cham, 2015, pp. 245–245. doi:10.1007/978-3-319-11797-3_140.

[7] 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. URL: https://doi.org/10.1145/3626772.3657874. doi:10.1145/3626772.3657874.

[8] R. Aly, Z. Guo, M. S. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, A. Mittal, The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task, in: R. Aly, C. Christodoulopoulos, O. Cocarascu, Z. Guo, A. Mittal, M. Schlichtkrull, J. Thorne, A. Vlachos (Eds.), Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Dominican Republic, 2021, pp. 1–13. URL: https://aclanthology.org/2021.fever-1.1/. doi:10.18653/v1/2021.fever-1.1.

[9] V. Setty, Factcheck editor: Multilingual text editor with end-to-end fact-checking, 2024. URL: https://arxiv.org/abs/2404.19482. arXiv:2404.19482.

[10] J. Thorne, A. Vlachos, Automated fact checking: Task formulations, methods and future directions, in: E. M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference

on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 3346–3359. URL: https://aclanthology.org/C18-1283/.

[11] R. Aly, Z. Guo, M. Schlichtkrull, J. Thorne, A. Vlachos, C. Christodoulopoulos, O. Cocarascu, A. Mittal, Feverous: Fact extraction and verification over unstructured and structured information, 2021. URL: https://arxiv.org/abs/2106.05707. arXiv:2106.05707.

[12] J.-F. Bussotti, L. Ragazzi, G. Frisoni, G. Moro, P. Papotti, Unknown claims: Generation of fact-checking training examples from unstructured and structured data, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 12105–12122. URL: https://aclanthology.org/2024.emnlp-main.675/. doi:10.18653/v1/2024.emnlp-main.675.

[13] J. Wallat, A. Abdallah, A. Jatowt, A. Anand, A study into investigating temporal robustness of llms, 2025. arXiv:2503.17073.

[14] V. Venktesh, V. Setty, A. Anand, M. Hasanain, B. Bendou, H. Bouamor, F. Alam, G. Iturra-Bocaz, P. Galuščáková, 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.

[15] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB: Massive text embedding benchmark, in: A. Vlachos, I. Augenstein (Eds.), Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 2014–2037. URL: https://aclanthology.org/2023.eacl-main.148/. doi:10.18653/v1/2023.eacl-main.148.

[16] J. Chen, A. Sriram, E. Choi, G. Durrett, Generating literal and implied subquestions to fact-check complex claims, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 3495–3516. URL: https://aclanthology.org/2022.emnlp-main.229/. doi:10.18653/v1/2022.emnlp-main.229.

[17] L. Pan, X. Wu, X. Lu, A. T. Luu, W. Y. Wang, M.-Y. Kan, P. Nakov, Fact-checking complex claims with program-guided reasoning, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 6981–7004. URL: https://aclanthology.org/2023.acl-long.386/. doi:10.18653/v1/2023.acl-long.386.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, W. Chen, Lora: Low-rank adaptation of large language models, CoRR abs/2106.09685 (2021). arXiv:2106.09685.