NGU_Research at CheckThat2025: An LLM-Based Hybrid Fact Checking Pipeline for Numerical Claims

Notebook for the CheckThat! Lab at CLEF 2025

Mohamed A. Abdallah[1,*,†], Rokayah M. Fekry[2,†] and Samhaa R. El-Beltagy[3,†]

[1]NewGiza University, First 6th of October, Giza Governorate 3294701, Cairo, Egypt.

Abstract

In this work, we present a four-stage, retrieval-augmented LLM pipeline for fact-checking numerical claims. The pipeline rewrites each numerical claim into a focused question, fuses OpenAI dense vectors with BM25 to fetch evidence, answers in context with DeepSeek-Chat, and issues language-aware verdicts via GPT-4.1-mini. The system ranked second in Arabic (macro F1 = 0.635) and fourth in Spanish (0.244) on the CLEF-2025 leaderboard, showing that balanced hybrid retrieval and encoding proper insights into prompts can deliver competitive accuracy on limited hardware.

Keywords

BM25, MacroF1 score, Hallucinations, Hybrid Retrieval Models, Instruction Tuning, Large Language Models (LLMs), Prompt Engineering

1. Introduction

Numerical claims such as 'GDP rose by 7 %', 'the budget allocates 35000 crore rupees' or '300 peaceful students were detained' dominate public debate and spread rapidly throughout the world, regardless of the language in which it is aimed. Verifying such statements is hard according to the trustfulness of the news resources, also the figures may be rounded, translated, or nested inside long documents, and the editorial yardsticks for "mostly true" or "misleading" may easily differ from Arabic to Spanish to English.In CLEF-2025, specifically in the CheckThatLab task 3 (Fact-Checking Numerical Claims) [1] therefore sets a three-way classification task (True, False, Conflicting) is set over a multilingual corpus whose training material is richly annotated with gold evidences and numeric normalizations. The challenge is to build a checker that can read a claim, locate the most relevant passages in a shared evidence pool, and decide its veracity while honoring each language's editorial style. Our approach treats fact-checking as a retrieval-augmented generation (RAG) problem. First, we ground the claim in external knowledge by retrieving passages from a Qdrant index that combines OpenAI dense embeddings with classic BM25 term matches. This hybrid retriever anchors both numeric surface forms ("3.5 M" vs. "three-and-a-half million") and semantic paraphrases. The retrieved snippets are then injected into a large language model that produces a concise, language-matched answer. Finally, a lightweight LLM judge compares the original claim to the reference answer, issuing the verdict according to language-aware label priors (e.g., Arabic collapses any ambiguity into False). By separating where information is found from how it is reasoned over, the pipeline keeps hallucination in check and makes each decision traceable to specific evidence. The remainder of the paper details the Related Work (2), Methodology (3), Experiments (4), Results (5), and Conclusions (6).

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

*Corresponding author.

†These authors contributed equally.

mohamedabdallah9850@gmail.com (M. A. Abdallah); rokayah.mohamed@ngu.edu.eg (R. M. Fekry); samhaa@computer.org (S. R. El-Beltagy)

https://www.linkedin.com/in/mohamed-abdallah-579014202/ (M. A. Abdallah); https://www.linkedin.com/in/rokayah-mahmoud-b153b510b/ (R. M. Fekry); Phttps://www.linkedin.com/in/samhaa-el-beltagy-b410a530/ (S. R. El-Beltagy)

2. Related Work

Hallucinations of AI, which are defined as correct-sounding but incorrect data, are a phenomenon that has been observed in generative AI tools based on large language models such as GPT, T5, and BART. In her work [2], Kamel showed the distribution of fear of hallucinations in various fields, specifically when AI-generated output is used for decision making.

Hallucinations are not just a linguistic issue. It extends to factual correctness, which is dangerous when numerical data is involved. Numerical hallucinations involve the generation of incorrect or fabricated numerical data (e.g., statistics, dates, values) in the AI-generated outputs. Bera et al. addressed this issue by introducing a framework for validating numerical values in AI-generated financial summaries. Their system uses a T5-based model to predict masked numbers and compares them with ground-truth values extracted from source reports. [3]

With the advancement of summarization of technical reports, the need of developing such models to ensure the factuality of the numerical data in these reports increases day by day. Bera et al. in [3] has provided an automated framework that focuses on validating the numerical data in generated financial report summaries. Their approach addressed the prediction of masked numerical values in the summaries generated by using a T5-based model. By taking only the most relevant sections of the original report of interest, they cross-checked between the predicted and actual numerical data to verify the numerical factuality of a specific sentence in the report.

V. Venktesh et al. has introduced QuanTemp to address the gap between the available models and the numerical claims available in real life. With a multi-domain dataset that focused on temporal, statistical, and other diverse aspects. Their dataset was also supported by detailed and precise metadata as well a strong basis evidence collection to avoid data leakage. They were able to achieve a model with a macro-F1 score of 58.32 that was able to address real-life data [4].

The previous researches highlight that numerical hallucination is a growing subfield requiring domain-specific fact-checking approaches. Therefore, recent work has emphasized the value of multistage or hybrid fact-checking pipelines.

3. Methodology

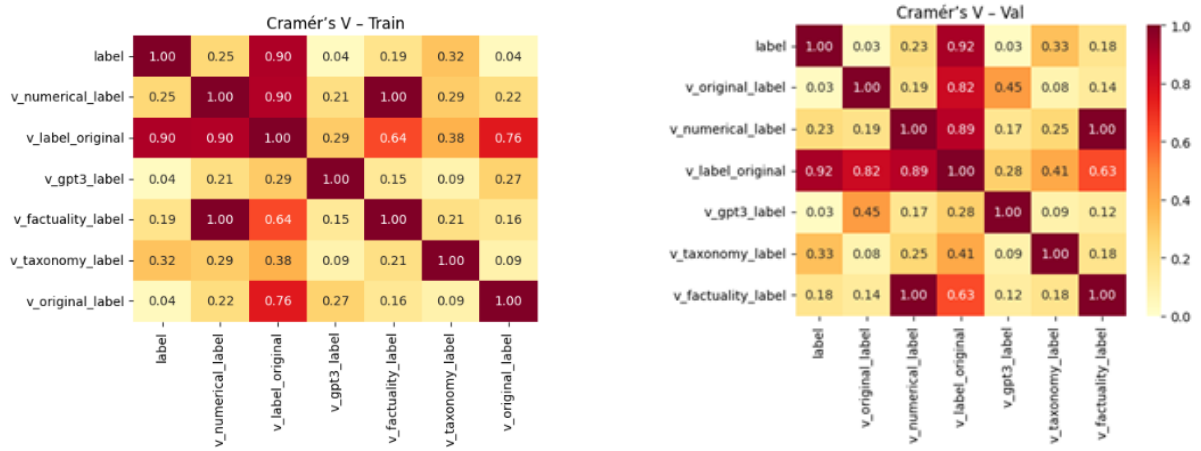
3.1. Dataset Exploration, Parsing, and feature extraction

The corpus we work with mixes three languages and three data splits. After merging the official training and validation partitions we have 13632 training claims and 4048 validation claims: English dominates ($\approx$ 73 %), followed by Arabic and Spanish. The class balance is skewed; half the claims are labelled "False", one-fifth "Conflicting", and the rest "True," with Arabic containing no Conflicting examples at all. A separate test set (3656 English claims, 1806 Spanish claims, and 482 Arabic claims) is held back for leaderboard scoring.

The dataset is organized as JSON files per language (English, Spanish, and Arabic) containing each claim's text, taxonomy tag, original verdict, normalized quantities, source URLs, and oracle document metadata for training and validation. For every claim, a pool of top-100 BM25-retrieved evidence is pulled from a shared, language-specific corpus. Each piece of evidence provides a gold context to guide the three-way (true/false/conflicting) classification task.

Our insights about data and the annotation process show that nearly every claim carries at least one explicit quantity: in the merged data, numeric-bearing statements outnumber text-only statements roughly a seven-to-one ratio. Correlation analysis with Cramér's V represented in fig.1 highlights which auxiliary signals help with the decision of the final label.

English and Spanish fact-checking teams often talk in shades of truth, from the wild "pants-fire" label all the way through "barely-true," "half-true," and "mostly-true." To fit our three-way task, we simply treat "barely-true" as False, "half-true" as Conflicting, and "mostly-true" as True as shown in fig.2a and fig.2b.



(a) Cramér's V Train

(b) Cramér's V Val.

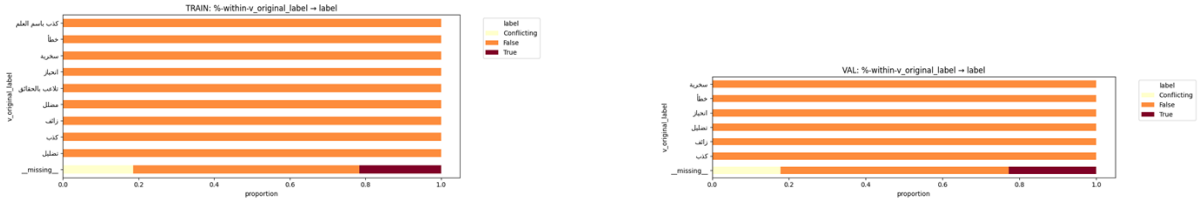
Figure 1: Correlation analysis with Cramér's V between different features in Training and Validation data

(a) Train Labels

(b) Validation Labels

Figure 2: Mapping V_label_original data into True, False, and conflicting classes.

In Arabic, the vocabulary looks different, terms like سخرية ، تضليل و زائف carry any hint of doubt straight into "False," reserving "True" only for rock-solid claims fig.3a and fig.3b. That sharper boundary gives Arabic a very different feel, and our model needs to work within that editorial style. By baking these language-specific tendencies into our prompts as default priors, we let the LLM start its reasoning with the same instincts that human fact-checkers bring. Because every claim in a language taps the same pool, the overlap between evidences is huge.



(a) Arabic Train Labels

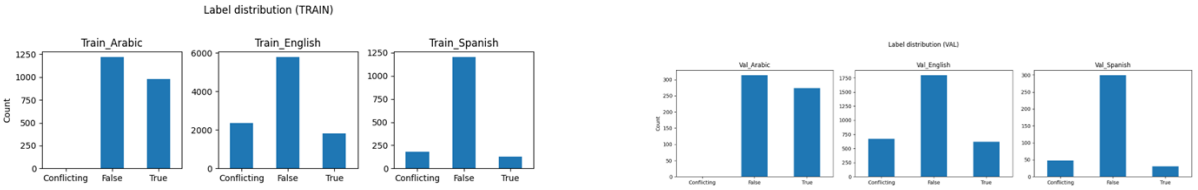
(b) Arabic Validation Labels

Figure 3: Arabic language focused Mapping V_label_original data into true, false, and conflicting classes.

In general, data shows serious bias towards the false label in Fig. 1reffig:disttrain and Fig.4b, suggesting that it would confirm the ability of a fact-checking pipeline to detect which claims are false.

English training claims point to 2.17 million document mentions, yet only 383k IDs are unique, and more than 320k of those are cited by multiple claims (nearly 84 % re-use). Spanish and Arabic exhibit even tighter reuse. For Spanish, training data shows nearly 117k mentions vs. 6.9k unique IDs as a 98% re-use, and for Arabic training data, we have nearly 221k mentions and 5k unique, indicating greater than 99% re-use. For validation and test sets, evidences are also shared across multiple claims. This structure is more than curiosity, it has two direct consequences for our work. First, the dense overlap simplifies the retrieval mission. Second, by encoding the language-specific label priors in our prompts,

we have the intuition of boosting system performance, as later results show.



(a) Training Label Distribution

(b) Validation Label Distribution

Figure 4: Distribution of Training and Validation labels

3.2. Large Language Models (LLMs)

3.2.1. Introduction to LLMs

GPT-3, GPT-4, Gemini, and many other well-known models that have been part of our daily life are examples of large language models (LLMs). These models are transformer-based models that are very well trained on a huge and vast corpus of mainly textual data by following self-supervised learning. State-of-the-art performance on a variety of natural language understanding and generation tasks has been made possible by their ability to encode intricate linguistic, semantic, and factual patterns. Language Models are Few-Shot Learners[5]. To train an LLM, the model passes by two main stages: pretraining and Fine-tuning. In the pretraining stage, the model learns the general language patterns by providing large text datasets like Common Crawl, Wikipedia, etc. [5]. Aiming to improve the alignment with human expectations, the models are fine tuned in the second stage by being adjusted on specific datasets. In this stage, the models follow supervised learning or reinforcement learning with human feedback (RLHF). Training Language Models to Follow Instructions with Human Feedback [6].

3.2.2. Prompt engineering as a paradigm

Prompt engineering is a term that refers to formulating input texts (prompts) to steer the model to generate specified responses. This method has emerged as a lightweight replacement yet powerful method for full retraining models. There are multiple prompting strategies like zero-shot, Few-shot, and chain-of-thought prompting. All of these strategies involve providing the model with the task description, however, the amount of examples provided to the model vary from one strategy to the other. In zero-shot prompting, only the task description is provided to the model without providing any examples, while in few-shot a few examples are also provided to the model [5]. Chain-of-thought prompting is a bit different were the model is provided by a step-by-step reasoning of the whole task. This method is generally used in tasks that involve arithmetic or factual reasoning [7]. Adapting general-purpose LLMs to domain-specific tasks, such as fact-checking, requires prompt engineering. This is because it points the LLMs in the direction of pertinent information and promotes lines of reasoning that strengthen factual consistency [8].

3.2.3. Automating LLM for factuality checking

With the improvement of LLMs in text generation tasks, it became crucial to develop models that are also capable of evaluating the quality, relevance, and factual correctness of the content. The role of LLMs in fact-checking tasks involves judging the truthfulness of a claim of interest with respect to given evidence that serves as ground truth. This method expands on the idea that LLMs can internally model world knowledge and reasoning pathways adequate for factual verification when trained on a variety of fact-rich corpora [9].

Zheng et al in [10], and Bai et al in [11] have shown in their studies that LLMs were able to successfully judge textual generation that surpassed human performance in either correctness, helpfulness, or even

harmfulness. The studies not only included judging text generations but also expanded to include evaluating summaries or answers in QA systems, as Bubeck et al have shown in their work [12]. Moreover, Jiang et al in [13] have assessed the factuality of statements by comparing them with structured or unstructured statements. FEVER, a framework introduced by Thorne et al in [14], has inspired some other researches to follow the same approach by prompting a claim to the model along with a piece of an evidence that is supporting or contradicting with the claim. A verdict is then returned with one of the following labels: ("SUPPORTED", "REFUTED", "INSUFFICIENT INFO").

3.2.4. Instruction Tuning of LLMs for Fact-Checking of Numerical Claims

In order to help LLMs generalize to new or unseen instructions without any additional training, the instruction tuning process is followed. The process involves exposing the model to a set of pairs of instructions and responses, which helps it to follow these natural language instructions. The instruction tuning process differs from general fine-tuning, which typically focuses on a specific task, in that it is based on teaching the model to follow task descriptions and expectations. Such a process improves the versatility of a model across various downstream tasks [6], [15]. T0 [16], FLAN-T5 [17], and InstructGPT [6] are examples of notable instruction-tuned models that showed a strong performance in zero-shot and few-shot in summarization, reasoning, and QA tasks.

Instruction tuning plays a significant role in the context of fact-checking, specifically for numerical claims. The instruction tuning would be helpful from many aspects, such as understanding what constitutes a factual claim. They are also able to reason over numerical and symbolic information. When provided with evidence, structured reasoning, or multi-hop deduction, the models are able to justify the judgments. Models can be trained using prompts such as:

> *"Given the claim and the evidence, decide whether the claim is supported, refuted, or Not Enough Information. Explain your reasoning."*

Several datasets, like the previously mentioned FEVER, AQuA, or even some corpora that contain numerical statement associated with evidences are adapted to serve for this purpose. However, transparency and consistency of the outputs of the model should be considered. Some techniques are used to address these aspects, such as chain-of-thought prompting and rationale generation [18].

In the context of numerical data, it is quite challenging to relay on LLMs for evaluating and checking the factuality and correctness of a given numerical claim. When it comes to numerical and arithmetic operations, LLMs can fail to generate precise numbers since they are prone to numerical hallucinations. The key challenge is that not only does context understanding matter, but also the mathematical verification of a claim is needed to evaluate its factuality. In such cases, the typical instruction tuning can be insufficient if they are not explicitly trained on numerical datasets or if external tools like calculator modules or symbolic reasoning chains are not integrated with it. Recently, hybrid setups, as have been presented by Chen et al. and jiang et al. in [19], and [13] respectively tend to combine LLMs with other retrieval modules, symbolic calculators, or even rule-based verification pipelines. Their results have shown a significant reliability in quantitative claims verification.

3.3. System Design

Our fact-checking system in Fig. 5 moves each claim through four modular stages: it first calls a multilingual LLM that creates a concise investigative question from the provided claim. This created a question, and the original claim drives a hybrid retriever that blends dense semantic similarity from OpenAI embeddings with exact-term BM25 scores, retrieving the top 15 passages most likely to mention the claim. These passages are provided as a context to a model that will use in answering the created question for a short reference answer in the same language. Finally, a verdict classifier compares that answer against the original claim.

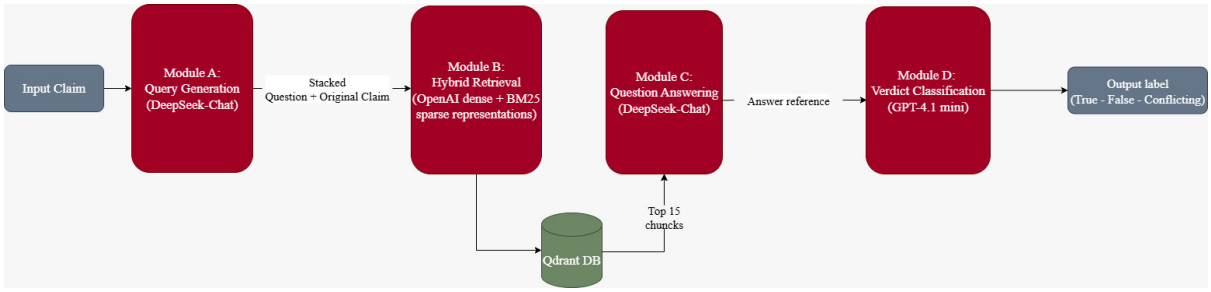


Figure 5: Fact-Checking Pipeline Block Diagram.

3.3.1. Query Generation

The first functional block rewrites each raw claim into a single-sentence question in the very same language. We feed the claim to DeepSeek-Chat under a few-shot prompt that instructs the model to hunt explicitly for all numbers, dates, quantities, and any stated causal links or consequences while forbidding any judgment of trustworthiness. Because the training pool spans Arabic, Spanish, and English, the prompt illustrates all three languages and enforces language preservation in the output. To avoid redundant calls, the module keeps a pickled cache keyed by the original claim; once a question has been generated, it can be reused throughout later experiments.

3.3.2. Hybrid Evidence Retrieval

In the retrieval stage, we concatenate the DeepSeek-generated question directly with the original claim, creating a single "stacked" query that anchors both the numeric surface forms and their possible paraphrases. This composite query is submitted to a Qdrant index where every passage is stored twice: once as a dense vector produced by OpenAI's text-embedding-3-large model and again as a sparse term-frequency map for BM25 scoring. Dense vectors help us match phrased or translated numbers ("three-and-a-half million", "3.5 M"), while the sparse map guards against LLM hallucinations by insisting on literal term overlap. To rank candidates we apply a simple linear fusion: each passage receives a min-max-normalized cosine score from the dense space and a normalized BM25 score; the two are blended with equal weight $alpha$ = 0.5. Extensive ablations showed this midpoint consistently lifts both early precision and deep-rank recall versus either signal alone, and using a cut-off of fifteen passages gives us a comfortable recall buffer within token limits. The top-15 passages, ordered by this hybrid score, advance as the evidence context for the question-answering module.

3.3.3. Question Answering

The fifteen ranked passages are concatenated—truncated if they exceed an 8k character safety cap and supplied, along with the generated question, to DeepSeek-Chat under a "document-first" prompt, i.e., the model must harvest its answer from the provided context and may consult prior knowledge only when the evidence is genuinely silent. This constraint keeps explanations grounded and reduces hallucination as in ref. The prompt also enforces brevity and language fidelity, so the reply is a compact, fact-centered sentence in Arabic or Spanish, mirroring the question's tongue. The result is a crisp reference answer that distils the numeric gist of the evidence and is ready for verdict comparison.

3.3.4. Verdict Classification

In the final step, we present the original claim and the reference answer as a tuple to a lightweight GPT-4.1-mini judge. The prompt spells out a quantitative rubric that is guided by the insights we extracted from the original labels and annotation tendencies. If the answer confirms at least 75% of the claim's numeric and causal content the label is True; if it contradicts or diverges on most key facts the label is False; anything in between is Conflicting. Because Arabic training data never uses the

Conflicting tag, we deploy a parallel prompt for Arabic that collapses all ambiguous or mixed cases into False, mirroring native editorial practice. The judge runs at zero temperature and is instructed to output the single word label—no rationale, no extra tokens—so its decision can be consumed directly by the evaluation stage. To stay within LLM rate-limits and OpenAI embedding quotas, we batch embedding calls and maintain a pickled question cache that eliminates redundant DeepSeek prompts, keeping per-claim latency near one second on an A100 GPU Colab environment. With the design fully laid out, we now turn to the empirical studies that shaped these choices.

Table 1
English, claim-only retrieval results on a sample of 300 examples from the training set (100 per language).

Retriever	MRR	R@1	nDCG@1	R@3	nDCG@3	R@5	nDCG@5	R@10	nDCG@10
BGE-m3	**0.404**	**0.287**	**0.287**	**0.475**	**0.401**	**0.545**	**0.430**	0.644	**0.462**
E5-large-instr.	0.378	0.257	0.257	0.465	0.380	0.535	0.407	0.634	0.439
Google text-embedding-004	0.366	0.257	0.257	0.406	0.345	0.515	0.389	**0.653**	0.434
OpenAI text-embedding-3-large	0.372	0.257	0.257	0.426	0.355	0.545	0.404	**0.653**	0.439

Table 2
Spanish, claim-only retrieval results on a sample of 300 examples from the training set (100 per language).

Retriever	MRR	R@1	nDCG@1	R@3	nDCG@3	R@5	nDCG@5	R@10	nDCG@10
BGE-m3	**0.565**	**0.404**	**0.404**	**0.654**	**0.554**	**0.788**	**0.609**	**0.904**	**0.646**
E5-large-instr.	0.474	0.269	0.269	0.654	0.497	0.769	0.544	0.808	0.557
Google text-embedding-004	0.491	0.327	0.327	0.596	0.482	0.769	0.552	0.846	0.577
OpenAI text-embedding-3-large	0.564	0.423	0.423	0.654	0.561	0.769	0.608	0.846	0.632

3.3.5. Effect of Hybrid Fusion

Tables 3, 4, and 5 sweep the fusion weight α from 0 (BM25-only) through 0.5 (equal blend) to 1 (similarity-only) while holding every other setting fixed to the OpenAI dense backbone and stacked queries. A clear precision-recall trade-off emerges: BM25 dominates deep recall in Arabic (R@15 = 0.970) but collapses on English early precision (MRR = 0.221). Pure similarity flips that pattern, giving English the best head-of-rank scores (MRR = 0.369, R@15 = 0.713) yet losing recall in Arabic. The midpoint $\alpha = 0.5$ offers the most balanced profile, which maintains near-BM25 recall in Arabic (0.911) and near-similarity precision in English, while even posting the highest Spanish R@15 (0.962). These results confirm that a modest blend is the safest cross-lingual choice for downstream QA and verdict stages.

Table 3
Arabic, α-sweep (k = 15)

α (fusion)	MRR	R@1	R@5	R@15
0 — BM25 only	**0.756**	**0.653**	**0.851**	**0.970**
0.5 — Hybrid	0.734	0.634	0.832	0.911
1 — Similarity only	0.668	0.584	0.772	0.832

Table 4
English, α-sweep (k = 15)

α (fusion)	MRR	R@1	R@5	R@15
0 — BM25 only	0.221	0.139	0.307	0.436
0.5 — Hybrid	0.351	0.218	0.525	0.693
1 — Similarity only	**0.369**	**0.257**	**0.525**	**0.713**

Table 5
Spanish, α-sweep (k = 15)

α (fusion)	MRR	R@1	R@5	R@15
0 — BM25 only	**0.571**	**0.423**	**0.788**	0.865
0.5 — Hybrid	0.557	0.385	0.750	**0.962**
1 — Similarity only	0.552	0.404	0.750	0.923

3.3.6. Impact of k (number of retrieved passages)

Reading only the hybrid rows inside Tables 3, 4, and 5, we see how recall widens when the cutoff grows from the first 5 passages to the full 15. Arabic improves from 0.832 (R@5) to 0.911 (R@15); English climbs from 0.525 to 0.693; and Spanish posts the biggest gain, leaping from 0.750 to 0.962. Mean Reciprocal Rank changes by less than 0.02 in any language, showing that extra passages rarely shift the position of the first hit but do rescue edge cases that would otherwise be missed. We standardize on k = 15 for the remainder of the pipeline but set the limit to 8 k-character context budget as the increase in related overhead cost is still considerate.

3.3.7. Stacking Question with claim

We stacked the generated question together with the original claim. The intuition behind this idea is that claims may have inaccurate pieces of information, hence, the presence of the original query can provide a useful signal to the retriever to help better retrieve the gold chunk. The question spells out the "how many?" or "what percent?" giving the encoder clearer clues, while BM25 still sees the original text. We keep this approach the default for our main pipeline. As we unpacked system components and illustrated experiments, we now turn to the results.

4. Results

Our final submission, run on the shared test servers, placed 2nd on the Arabic leaderboard with a macro-average F1 of 0.635 (3 trials), driven by a True-class F1 of 0.542 and a False-class F1 of 0.729; conflicting is naturally 0.0 because that label does not exist in Arabic. On the Spanish leaderboard we finished 4th; the single run we submitted scored a macro-average F1 of 0.244, with class-wise F1s of 0.169 (True), 0.142 (Conflicting), and 0.421 (False), with a notably better F1 in the False class for both largely due to the natural imbalance in the data and its tendency towards the False class. Fig.4a and fig. 4b. It's a basic idea that making more predictions of the False class would usually increase the odds of F1-score of the False class, which is basically the major class, rather than the F1-score of the other two classes. Although we skipped English entirely due to limited compute hours, the two languages we tackled still placed comfortably in competing positions of their respective tables. Two important insights emerge. First, hybrid retrieval with $\alpha \approx 0.5$ proves robust: it consistently beats the BM25-only approach and the similarity-only approach. Second, language-aware labelling insights matter for better LLM instruction tuning. Together, these results confirm that careful prompt engineering and a balanced fusion strategy can offset modest hardware budgets. This closes the empirical evaluation of our system.

5. Conclusion

This study addressed the CLEF-2025 task of fact-checking numerical claims across Arabic, Spanish, and English. We framed the problem as a four-step, LLM-centric pipeline: a multilingual question generator rewrites each claim, a hybrid retriever (OpenAI dense vectors + BM25, α = 0.5) gathers fifteen evidence passages, a document-first LLM extracts a concise answer, and a verdict judge assigns True, False, or Conflicting with an Arabic-specific prompt that collapses ambiguity to False. Extensive ablations showed that each module contributes measurably. Key design choices, including stacking the

generated question with the claim, normalizing fusion scores, and caching LLM calls, kept the system both interpretable and resource-light while respecting language-specific editorial styles. The hybrid retriever outperformed BM25-only or similarity-only baselines in every language. On the hidden test servers our submission ranked 2nd in Arabic (Macro F1 = 0.635) and 4th in Spanish (Macro F1 = 0.244) despite omitting English to conserve GPU hours, evidence that prompt-level language adaptation and balanced retrieval can deliver competitive accuracy under tight hardware and time budgets.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI-GPT-4o: 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. 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.

[2] H. Kamel, Understanding the impact of ai hallucinations on the university community, Cybrarians Journal 62 (2024) 74–91.

[3] S. Bera, et al., Validating numerical information in ai-generated financial summaries, in: Proceedings of the ACL Workshops, 2022.

[4] V. Venktech, et al., Quantemp: Quantitative fact verification across temporal domains, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, 2020. URL: http://arxiv.org/abs/2005.14165. doi:10.48550/arXiv.2005.14165, arXiv:2005.14165 [cs].

[6] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. URL: http://arxiv.org/abs/2203.02155. doi:10.48550/arXiv.2203.02155, arXiv:2203.02155 [cs].

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023. URL: http://arxiv.org/abs/2201.11903. doi:10.48550/arXiv.2201.11903, arXiv:2201.11903 [cs].

[8] H. Sun, X. Liu, Y. Gong, Y. Zhang, D. Jiang, L. Yang, N. Duan, Allies: Prompting Large Language Model with Beam Search, 2023. URL: http://arxiv.org/abs/2305.14766. doi:10.48550/arXiv.2305.14766, arXiv:2305.14766 [cs].

[9] A. Glaese, N. McAleese, J. Aslanides, et al., Improving alignment of language models via human feedback, arXiv preprint arXiv:2204.05862 (2022).

[10] D. Zheng, H. Liu, M. Lee, et al., Judging llm-as-a-judge: Evaluating llms as evaluators, arXiv preprint arXiv:2305.14688 (2023).

[11] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, J. Kaplan, Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL: https://arxiv.org/abs/2204.05862. arXiv:2204.05862.

[12] S. Bubeck, V. Chandrasekaran, R. Eldan, et al., Sparks of artificial general intelligence: Early experiments with gpt-4, arXiv preprint arXiv:2303.12712 (2023).
[13] H. Jiang, X. Lin, Y. Gao, et al., Can language models verify factual claims?, arXiv preprint arXiv:2305.14271 (2023).
[14] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, 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, 2018, pp. 809–819.
[15] Y. Wang, Y. Kordi, S. Mishra, et al., Self-instruct: Aligning language models with self-generated instructions, arXiv preprint arXiv:2212.10560 (2022).
[16] V. Sanh, A. Webson, C. Raffel, et al., Multitask prompted training enables zero-shot task generalization, arXiv preprint arXiv:2110.08207 (2021).
[17] H. W. Chung, L. Hou, S. Longpre, et al., Scaling instruction-finetuned language models, arXiv preprint arXiv:2210.11416 (2022).
[18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2022. URL: https://arxiv.org/abs/2201.11903. doi:10.48550/arXiv.2201.11903. arXiv:2201.11903.
[19] Y. Chen, Y. Zhao, B. Y. Lin, et al., Factuality enhanced language models for numerical reasoning, arXiv preprint arXiv:2302.04279 (2023).