AuthEv-LKolb at CheckThat! 2024: A Two-Stage Approach To Evidence-Based Social Media Claim Verification

Luis Kolb[1], Allan Hanbury[1]

[1]TU Wien, Data Science Research Unit, Favoritenstraße 9-11/194-04, 1040 Vienna, Austria

Abstract

This paper covers our submission to CLEF 2024 CheckThat! Lab task 5: Authority Evidence for Rumor Verification. Misinformation as claims on social media platforms is an ever present issue. We present a two-stage approach to verify claims posted on social media based on evidence posted by authority accounts to the same platform. We conduct experiments to find the optimal setup with respect to the target metrics specified in the CLEF 2024 CheckThat! Lab, where we are participating in Task 5. Our experiments show that Large Language Models, of which we compare GPT-4 and Llama3-70B, are suited to this particular verification task. The paper finally presents areas where further improvements can be explored.

Keywords

fact-checking, natural language processing, information retrieval, CLEF 2024

1. Introduction

This paper covers our submission to CLEF 2024 CheckThat! Lab task 5: Authority Evidence for Rumor Verification. The descriptions for all tasks, including our own, are provided in the conference paper by the lab organizers [1].

There are many options available to large platform operators to combat misinformation on their platforms, like professional fact-checking services or even manually fact-checking claims on their platform. Manually checking every reported post has turned into a task that is no longer a viable option for most large platforms, due to the sheer volume of content uploaded by users. There are improvements to these methods that platforms can implement, like identifying and matching similar claims to already fact-checked claims and reusing the work that already went into fact-checking the original claim. This was already a task at the CLEF 2022 CheckThat! Lab [2]. However, there are also alternative approaches.

Specifically on X.com (formerly Twitter), a community fact-checking system is in place, colloquially called “Community Notes". In a 2022 study, Pröllochs [3] investigated the impact of this feature, and one of the findings was that the feature’s “[...] community-driven approach faces challenges concerning opinion speculation and polarization among the user base – in particular with regards to influential user accounts." (p.11 [3])

In this paper, we present a more automated approach to fact-checking claims on social media, using official government statements on the same platform to verify claims, which could be used both as a stand-alone service, and as a tool to assist human fact-checkers and fact-checking services. There are some drawbacks to relying on official authority accounts rather than the community, which are discussed in Section 5.3.

The official CLEF 2024 CheckThat! Lab Task 5 is defined as: “Given a rumor expressed in a tweet and a set of authorities (one or more authority Twitter accounts) for that rumor, represented by a list of tweets from their timelines during the period surrounding the rumor, the system should retrieve up to 5 evidence tweets from those timelines, and determine if the rumor is supported (true), refuted (false), or unverifiable (in case not enough evidence to verify it exists in the given tweets) according to the evidence" [4].

CLEF 2024: Conference and Labs of the Evaluation Forum, 9-12 September, 2024, Grenoble, France, 2024

kolb.luis@gmail.com (L. Kolb); allan.hanbury@tuwien.ac.at (A. Hanbury)

https://luiskolb.at (L. Kolb)

We experimented with several setups and combinations of different strategies. Our approach involved and tested two stages: a retrieval stage, and a verification stage.

- In the retrieval stage, for a given claim (also referred to as “rumor"), we aim to retrieve evidence from the set of all tweets relevant to that claim.
- In the verification stage, we use the retrieved evidence to predict a label for the claim (REFUTES, SUPPORTS or NOT ENOUGH INFO).

We structure our paper into the following sections: Section 2 introduces the data we are working with, and the target measures we use to evaluate our experiment results. Chapter 3 discusses the main objectives of the experiments we conducted during our participation, while Chapter 4 presents our approach to the task. The results of our experiments are presented in Chapter 5. Finally, Chapter 6 concludes our paper, and presents questions and topics for further research.

2. Task Dataset and Evaluation Measures

The data we are working with consists of various tweet texts. For every tweet making a claim, there is a set of tweets authored by authority sources, only some of which are relevant to the claim. Notably, the tweet texts do include links to attached images that were posted with the tweet, which could contain some additional information (see Section 6 discussing future work). The only metadata directly included in the dataset is the username and the tweet ID (which can be used to fetch more metadata from the twitter API), but these are present only for authority statements, not for claims (which are only a single “text string").

Here is an example of what a rumor to be verified looks like: every rumor is provided as JSON, with an ID, the claim text, and a list of statements, each of which contains the account URL that tweeted the statement, the tweet ID, and the tweet text. Labeled data also includes which of the statements are relevant to the claim.

```
{
  "id": "AuRED_142",
  "claim": "Naturalization decree in preparation: Lebanese passports for
  sale?! https://t.co/UuQ7yMbSWJ https://t.co/Jf1K1NbZJD",
  "statements": [
    [
      "https://twitter.com/LBpresidency",
      "1555424541509386240",
      "The Information Office of the Presidency of the Republic: What was
      published by the French newspaper “Liberation” about the “selling” of
      Lebanese passports to non-Lebanese is false and baseless news."
    ],
    ...
  ]
}
```

The dataset consists of 160 rumors overall, 128 of which were available with ground truth. Our approach did not involve learning-to-rank [5]. So, in our case, the dataset size is only relevant insofar as a larger and more diverse dataset is likely to cover a wider range of scenarios and topics, on which the proposed system could be tested.

For this task, the following target measures are considered when evaluating performance, as specified in the Task description [6]:

- Macro-F1 as the primary measure for the overall verification performance, which averages the F1 score for each of the three labels to account for class imbalance.

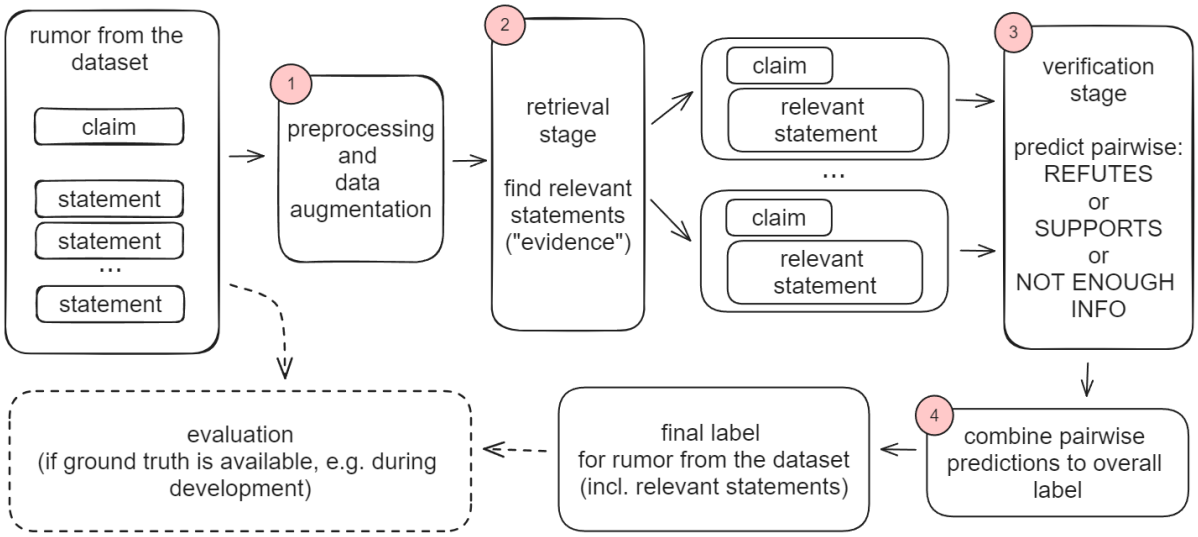

Figure 1: Visualization of our system for verifying a rumor. Numbers 1–4 indicate components with multiple configuration options.

- Strict Macro-F1 as a secondary measure for verification performance, which additionally considers found evidence. For this measure, a "true positive" needs to have a correct label for the rumor, as well as an overlap of at least one piece of evidence between ground-truth evidence and found evidence. More overlap does not increase the Strict F1 score.
- MAP (Mean Average Precision) as the primary measure for retrieval effectiveness.
- R@5 (Recall at 5 "items", as the system should retrieve at most 5 statements) as a secondary measure for retrieval effectiveness.

3. Experiment Design

For this paper, we ran a set of experiments, the results of which are presented in Section 5. We also made a submission to the CheckThat! Lab. This submission and the experiments are separate. The experiments serve to evaluate the effectiveness of different configurations for our proposed setup, and the submission is created using three of those configurations, since each team could submit up to three runs to the CheckThat! Lab.

Our proposed setup is illustrated in Figure 1. This setup was selected based on initial experimentation with a setup oriented after a paper on "Stance Detection" by Haouari et al. [7], which was refined over the course of development. We narrowed down the methods we used in initial experiments to the methods described in Section 4.

In the following list, each number refers to a numbered component in Figure 1. Our experiments quantitatively evaluate:

1. The impact of preprocessing and adding external data about the authority.
2. Methods of retrieving relevant statements ("evidence") in the retrieval stage.
3. The performance of transformer-based approaches for the "verification stage".
4. The impact of different options to influence how the pairwise scores from the verification stage should be combined into the overall label for the rumor.

4. Experiment Setup

Our setup consists of two stages: retrieval of evidence related to the current rumor, and a verification stage using the retrieved evidence from the previous stage to fact-check the rumor. We can also optionally include some preprocessing and data augmentation steps.

4.1. Preprocessing and Data Augmentation

Preprocessing approaches and strategies to combine the individual predictions were also part of our experiments. We aim to obtain the best performing setup of preprocessing strategies for both the retrieval and verification steps, and the best scoring strategies for the verification step.

Prepocessing and data augmentation (Figure 1 component number 1) are optional features:

- Data augmentation adds the Twitter display name and/or the Twitter author bio to the statement text, depending on the configuration.
- Preprocessing cleans up the text: remove line breaks, some special characters like quotes and hashtags, URLs, the pattern "RT @<username>" (added by the Twitter API for quote tweets) and emojis.

4.2. Retrieval Stage

For the retrieval stage (Figure 1 component number 2), there are multiple viable options. The main retrieval methods we focused on (and ran experiments for) were:

- PyTerrier to create a simple BatchRetrieve pipeline of (BM25) » (PL2). See the PyTerrier documentation for details.[1]
- Cosine distances between embeddings obtained through the OpenAI Embeddings API.

These methods are rather simple, but effective enough for this task. So long as a single relevant source is retrieved, the powerful LLM in the verification stage is able to correctly predict the judgment.

4.3. Verification Stage

For the verification stage (Figure 1 component number 3), we experimented with two major transformer-based approaches:

- A fine-tuned version of BART (specifically bart-large-mnli, available on Hugging Face), a sequence-to-sequence autoencoder by Facebook (Meta AI) [8], for zero-shot Natural Language Inference (NLI) [9].[2] We initially used NLI models for the verification stage, which use a classification approach to classify a combined text input as ENTAILMENT or CONTRADICTION. However the deep natural language understanding of LLMs allows the model to navigate somewhat complex reasoning tasks very effectively. In fact, they outperform the NLI models we used by a wide margin.
- Large Language Models (LLMs), for the submission GPT-4 by OpenAI [10] (specifically the model version named GPT-4-1106-preview), since it performed the best of all available models at the time of the submission period. Due to the relatively low complexity of the reasoning (which is somewhat similar to natural language inference) in the verification step, we theorize that most sufficiently large LLMs would perform similarly (e.g. Llama3-400b or Claude Opus). At the scale of Llama3-70b we saw significant performance drops compared to GPT-4-Turbo.

Our CheckThat! Lab submission was created using gpt-4-1106-preview as the LLM. After uploading the submission, we re-ran our experiment setup, for which we used gpt-4o-2024-05-13, as it was cheaper and faster to run, and was the newest available LLM from OpenAI.

For OpenAI completions, we invoke the LLM by using the OpenAI Assistants API, with each claim-evidence pairing creating a new thread, and the system prompt being set in the assistant.[3] For the assistant configuration, we used a temperature of 0.01 and a top-p of 0.5. These values should encourage

[1] https://pyterrier.readthedocs.io/en/latest/terrier-retrieval.html
[2] https://huggingface.co/facebook/bart-large-mnli
[3] https://platform.openai.com/docs/api-reference/assistants

consistent responses.[4] Llama3 completions are obtained using the Hugging Face Inference API with the default parameter values and the model Llama3-70B-Instruct, as more powerful Llama3 models are not yet available.[5]

The LLM is prompted in the verification stage with a prompt template that is populated with both the claim and the authority statement, and instructed using the system prompt to adhere to the output format, and to only use information from the prompt, and not to use its domain knowledge or knowledge from training data. The LLM will predict not only a label, but also a confidence in the label between 0 and 1, which will be used to combine the pairwise labels in the next step.

The system prompt, which gives the LLM instructions it must adhere to, is shown below. The OpenAI LLM Assistant API always adhered to this system prompt during our experimentation. We also activated the "JSON-mode" in the Assistants configuration, which ensures answers follow the format specified in the system prompt, though the system prompt on its own would likely be effective enough to ensure this behavior.

```
You are a helpful assistant doing simple reasoning tasks.
You will be given a statement and a claim.
You need to decide if a statement either supports the claim ("SUPPORTS"),
refutes the claim ("REFUTES"), or if the statement is not related to the
claim ("NOT ENOUGH INFO").
USE ONLY THE STATEMENT AND THE CLAIM PROVIDED BY THE USER TO MAKE
YOUR DECISION.
You must also provide a confidence score between 0 and 1, indicating
how confident you are in your decision.
You must format your answer in JSON format, like this:
{"decision": ["SUPPORTS"|"REFUTES"|"NOT ENOUGH INFO"],
"confidence": [0...1]}
No yapping.
```

Below is a real input message to the LLM (primed with the previous system prompt). In this example, the data was preprocessed and had no external data added:

```
"Statement from Authority Account 'LBpresidency': ''The
Information Office of the Presidency of the Republic denies
a false news broadcast by the MTV station about Baabda Palace
preparing a decree naturalizing 4 000 people and recalls that
it had denied yesterday the false information published by the
French magazine 'Liberation' about the same fabricated news ''"

Claim: "Naturalization decree in preparation: Lebanese passports for sale !"
```

Since we score every combination of rumor and evidence separately, we have to combine them to produce an overall label prediction. As part of our experiments, we tested (Figure 1 component number 4):

- Weighting ("scaling") prediction confidence scores by retrieval score. The retrieval stage, in addition to the top-5 documents, also returns the associated score used to compute the ranking, which optionally can be used here.
- Normalizing retrieval scores, as different retrieval systems return retrieval scores on different scales.
- Including versus ignoring NOT ENOUGH INFO predictions for the final label score calculation.

[4] https://medium.com/@1511425435311/understanding-openais-temperature-and-top-p-parameters-in-language-models-d2066504684f

[5] https://huggingface.co/docs/hub/en/models-inference

Once we have obtained label predictions and for every claim-statement pairing, we weigh the confidence the LLM in the verification stage predicted using the retrieval score (if this feature is set active in the configuration), and then calculate the mean of the predicted scores (confidences) to obtain our overall label prediction. If the summed, averaged scores cross a significance threshold, we predict the respective SUPPORTS or REFUTES label. The threshold is not tuned or learned, rather it is set manually at 0.15 such that two opposing predictions of roughly equal confidence cancel each other out, unless one predictions is much more significant than the other, opposing prediction. Thus, the threshold accounts for some variation between two roughly equally strong predictions. Our experiments show that for this data set, this simple approach of combining predictions is sufficient.

Since SUPPORTS predictions are positive, and REFUTES predictions are negative, taking the mean of the two predictions scores emulates a voting system with votes being weighted by the prediction confidences. In this system, NOT ENOUGH INFO predictions do not contribute to the final overall label, as a NOT ENOUGH INFO prediction from the LLM does not indicate any leaning toward either SUPPORTS or REFUTES. Optionally, we include the NOT ENOUGH INFO prediction in the average, lowering the total overall score – potentially below the significance threshold.

This type of task is related to “stance detection" of authorities, which was introduced in a paper by Haouari et al. (who are also the organizers of the 2024 CheckThat! Lab task 5) in 2023 [7]. Our approach follows up on their paper, and expands the implementation to also retrieve evidence from a predefined dataset. Graves [11] lists three families of approaches to automatic fact verification, one of which is "[...] consulting authoritative sources" [11]. Manually consulting a third-party authority is definitely a valid tool for in-depth fact-checkers, and our system aims to assist these fact-checkers by finding statements an authoritative source already posted publicly, and predicting the stance of the source to the rumor or claim.

The resources used during development and for the submission are listed here:

- For OpenAI embeddings and GPT-4-Turbo completions we used the OpenAI API.[6]
- Llama3-70b completions were obtained from the Hugging Face "Inference for Pros" API.[7]
- BM25, PySerini and TF-IDF retrieval methods as well as bart-large-mnli for verification were computationally cheap enough to effectively run on a local desktop PC (AMD Ryzen 5, Nvidia GTX 970, 16GB memory).

5. Results and Discussion

We participated in the CheckThat! Lab Task 5 [4], and independently ran experiments to find the best configuration options for our approach. The results of each are reported here, in their own subsections.

5.1. Experiment Results

To test the various configuration options we created, we ran automated experiments on the dev split of the dataset (containing 32 rumors to be verified using the included timeline) to answer this set of research questions:

- To what extent can tweets (“evidence”) relevant to a claim be retrieved from timelines of authority accounts, given an initial claim, a set of authority accounts and the timelines of those authority accounts?
- To what extent can a claim, given a list of tweets (“evidence”), accurately be identified as being supported by the evidence (true), refuted by the evidence (false), or unverifiable (not enough evidence to verify it)?
- To what extent can a pipeline combining the approaches from RQ1 and RQ2 refute or support a claim, automatically retrieving evidence from the timelines of authority accounts?

[6] https://platform.openai.com/docs/overview
[7] https://huggingface.co/docs/api-inference/index

Table 1
Experiment results for retrieval configurations on the dev set.

Rank	MAP	R@5	Retrieval	Preprocess	Author Name	Author Bio
1	0.688	0.754	PyTerrier	True	False	True
2	0.674	0.747	Embeddings	False	False	False
3	0.671	0.728	Embeddings	True	True	False
4	0.659	0.752	PyTerrier	True	True	True
4	0.659	0.752	PyTerrier	True	True	True
5	0.657	0.708	PyTerrier	True	False	False
6	0.643	0.717	Embeddings	True	True	True
7	0.643	0.717	PyTerrier	False	True	False
8	0.641	0.710	Embeddings	True	False	False
9	0.641	0.708	PyTerrier	False	False	False
10	0.640	0.657	Embeddings	False	False	True
11	0.637	0.708	PyTerrier	True	True	False
12	0.634	0.719	Embeddings	False	True	False
13	0.633	0.699	PyTerrier	False	True	True
14	0.628	0.675	PyTerrier	False	False	True
15	0.620	0.708	Embeddings	False	True	True
16	0.590	0.719	Embeddings	True	False	True

Table 2
Differences in average verification performance score over all configurations, for each feature. The positive difference represents the average score increase when value option 1 is used over value option 2.

Feature tested	Value option 1	Value option 2	Macro-F1 Difference	Strict-Macro-F1 Difference
Verification	OPENAI	LLAMA	+0.1911	+0.2066
Retrieval	PyTerrier	Embeddings	+0.0239	+0.0132
Preprocessing	False	True	+0.0139	+0.0117
External Data	False	True	+0.0066	+0.0078
Normalize	False	True	+0.0300	+0.0306
Scale	False	True	+0.0635	+0.0637
Ignore NEI	True	False	+0.0709	+0.0708

For the experiments we performed, which are presented in Table 1 to find the best retrieval configuration, given the features we tested (preprocessing, adding author name and author bio), we did not find significant differences looking only at the retrieval evaluation. Generally, the best MAP performance was obtained by the simple PyTerrier retrieval method of scoring with BM25, then re-ranking using PL2 (divergence-from-randomness), using preprocessing and including the author bio in the statement text. It seems that preprocessing slightly improves retrieval performance overall. For the secondary measure, Recall@5, PyTerrier also performed the best.

In our approach, we ran the experiments to optimize the system for the use case of verification, as a "pipeline" from start to finish (claim and timeline input, to overall label with evidence output). Table 2 lists changes in score when a feature is actively used in a configuration, versus when it is not. It also shows the score difference between experiments that used LLAMA3 and those that used GPT-4, and changes in verification score between experiments with each retrieval method described above. The features that were tested are described in Section 4. In Table 2, "Ignore NEI" means ignore NOT ENOUGH INFO (NEI) predictions for the overall score.

Since we ran experiments in all possible permutations of our configuration options, we calculate the mean score of every configuration where a feature is used, and do the same for every configuration where it is not used (for example, Preprocessing "True" vs. "False", or in the case of Verification methods "OPENAI" vs. "PyTerrier"). The difference in average score gives an indication of the score impact of the feature value. A positive score difference in Table 2 means the average score of the configurations

using value option 1 was higher than those using value option 2. In most cases, the difference is not meaningful.

Using GPT-4 over LLAMA3 yields the highest performance gain on average, a noticeable Macro-F1 score increase of about 0.2. This is not surprising, as the GPT-4 model is much more powerful, as mentioned previously. It would be interesting to see score differences on other comparably large language models, like Claude Opus or Google Gemini. However, that comparison is outside the scope of this paper.

Additionally, it would be interesting to see the influence of different retrieval methods on the verification performance. In our experiments, the difference between retrieval methods is rather small. As mentioned previously, the LLMs in the verification stage are powerful enough that a single piece of relevant evidence usually suffices to predict the correct label. Running the experiments on a more diverse dataset with different retrieval methods in a larger search space might hinder the verification stage from functioning properly. If no relevant evidence is found, the system is likely to predict NOT ENOUGH INFO – as it should.

The best performing configurations (at rank 1 and 2, see Table 5 in the Appendix) of the system yielded the best results when not using any preprocessing. Preprocessing nearly always removes some amount of signal along with the noise in the data, which might hurt LLM performance more than in helps. Roughly two thirds of all configurations achieving the highest scores used no preprocessing. Overall, the mean Macro-F1 score of system configurations using preprocessing is lower by 0.0139 in our experiments, see Table 2.

Proposed features like scaling by retrieval score, normalizing retrieval score to [0...1] and including external data did not have a significant impact in our experiments with this dataset. The impact of excluding NOT ENOUGH INFO predictions is noticeable, since in our configuration, the final label is created by averaging the confidences of all pairwise predictions by the LLM, and if the average over that list passes a threshold, a REFUTES or SUPPORTS label is predicted. Including NOT ENOUGH INFO predictions with a value of 0 simply lowers the average score, which at a retrieval-k of 5 pairs can be significant enough to make a difference. In this case, including the NOT ENOUGH INFO predictions in the average score results in the system being presumably too cautious to perform adequately.

In some cases, the verifier fails to correctly classify SUPPORTS or REFUTES. During our testing, in each of those cases, the system predicts NOT ENOUGH INFO overall, which is the ideal fail case. The system never predicted overall SUPPORTS where the actual overall label is REFUTES, or the other way around.

See the Appendix for the full tables, or view the Jupyter Notebook with the full tables on GitHub.[8]

5.2. CLEF Submission Results

In the CheckThat! Lab Task 5, we participated in the challenge for the English dataset. The measures reported by the Lab organizers were MAP and R@5 for retrieval, and Macro-F1 and Strict Macro-F1 scores for verification (see also Section 2). There was a limit of 3 runs able to be submitted per team, only one of which was allowed to use external data not included in the dataset (our run labeled “secondary1" used author display name and author bio from Twitter, if available). The CheckThat! Lab organizers also provide a baseline score. Here, we report our own results, and this baseline, the full leaderboard is available on the CheckThat! Lab Task 5 website.[9]

We submitted three runs, each with different configurations for our setup:

- “primary": No external data and not preprocessing, only OpenAI embeddings with “raw" data.
- “secondary1": OpenAI embeddings for retrieval, with external Twitter data about the author added, and no preprocessing.
- “secondary2": PyTerrier retrieval method, using preprocessed data.

[8] https://github.com/LuisKolb/clef-2024-authority/blob/main/clef/pipeline/eval_experiment_large.ipynb
[9] checkthat.gitlab.io/clef2024/task5

Table 3
Selected results for the English retrieval leaderboard. For each other team, the best submission score is presented here.

Team	Run Label	MAP	R@5	Retrieval	Preprocess	External Data
IAI Group	secondary1	0.628	0.676			
bigIR	primary	0.604	0.677			
Axolotl	primary	0.566	0.617			
Team DEFAULT	primary	0.559	0.634			
AuthEv-LKolb (ours)	primary	0.549	0.587	Embeddings	False	False
AuthEv-LKolb (ours)	secondary2	0.524	0.563	PyTerrier	True	False
AuthEv-LKolb (ours)	secondary1	0.510	0.619	Embeddings	False	True
	(baseline)	0.335	0.445			

Table 4
Selected results for the English verification leaderboard. For all runs, we used GPT-4 as the verification component. For each other team, the best submission score is presented here.

Team	Run Label	Macro-F1	Strict Macro-F1	Retrieval	Preprocess Data	External
AuthEv-LKolb (ours)	secondary1	0.895	0.876	Embeddings	False	True
AuthEv-LKolb (ours)	primary	0.879	0.861	Embeddings	False	False
AuthEv-LKolb (ours)	secondary2	0.831	0.831	PyTerrier	True	False
Axolotl	primary	0.687	0.687			
	(baseline)	0.495	0.495			
Team DEFAULT	primary	0.482	0.454			
IAI Group	secondary1	0.459	0.444			
bigIR	primary	0.458	0.428			

All three runs used GPT-4 as the verification stage as described in Section 4. Preprocessing and external data are described in Section 4. The configuration options for the combination of the pairwise predictions were all set to “False", meaning no scaling or weighting using the retrieval score, and NOT ENOUGH INFO predictions being included in the average used to calculate the overall label.

In the retrieval stage, presented in Table 3, the best score our system achieved was a MAP of 0.549 using the primary run setup. Notably, we achieved a R@5 score of 0.619 using the secondary setup with external data, which would have been 4th on the leaderboard if R@5 was the targeted measure. The highest score was achieved by team “IAI Group", with a MAP of 0.628, who used a “Crossencoder" approach, according to their Run ID on the official leaderboard.

In the verification stage, we achieved the best result with a Macro-F1 of 0.895 in the secondary system using external data (authority display name and authority bio, obtained from Twitter). Our results can be seen in Table 4.

As the leaderboard results show, our approach to retrieval did not work particularly well in comparison to the other participants. However, our verification component significantly outperformed the other participants. Presumably, this demonstrates the strength of Large Language Models in this type of task, where few relevant pieces of evidence are needed to predict correctly, and irrelevant evidence does not introduce significant noise to the overall prediction. Thus, even though our retrieval component was comparatively weaker, the relatively high Recall resulted in good predictions overall.

5.3. Limitations of the Approach

There are a few caveats to this proposed setup, and its utility would likely lie in serving as an additional tool in the toolbox used to combat the spread of misinformation. These caveats are:

- Human fact-checking by neutral sources will most likely be more precise, more reliable and more

trusted, assuming the fact-checkers themselves are seen as neutral and trustworthy (which is influenced by a multitude of factors, as analyzed in the study by Primig [12]).

- In contrast to “traditional" fact-checking, our approach does not verify the actual truth content in a claim, only whether authority sources support or dispute a claim. For this paper, we are working with the definition of “authority" laid out by Haouari et al. in their 2023 paper [13]. Authority sources can be government accounts, for example, a Ministry of Health in a given state could be considered an authority related to rumors or claims about public health related matters in the same state. In general, authorities are considered experts in a given area, but not all experts are necessarily considered authorities. Additionally, an account is considered an authority when a rumor is about the account holder themselves. For example, the dev split of the data set contains a rumor about a journalist being involved in a deadly car crash, and the statement “My loved ones and my people who were busy with me: I am fine [...]" posted by the journalist’s account is considered authority evidence refuting the rumor. Because of examples like this, we included experiments with adding external data like account name and account bio to the data set.
- Another consideration is model selection. For our submission, we used GPT-4-1106-preview, the most recent OpenAI model available at the time. It is important to note that closed models are subject to frequent changes, and “open" models like Llama3-400b should produce more predictable output over longer periods of time.[10] Additionally, closed models are usually subject to content moderation, which could plausibly impact system performance and reliability - the area of fact-checking often deals with controversial claims and statements, after all. Unfortunately, Llama3-400b was not yet publicly available at the time of writing.

Running the system in the best-performing configuration is also the most expensive way to run the system (both in terms of computing time and API costs). There are possible trade-offs, as if deployment “at scale" or “in production" is desired, some compromises could be necessary:

- BM25 performs similarly to the OpenAI embeddings cosine distances method. It is also much cheaper to execute, as it does not require an external API call and the associated tokens.
- For the best performing configuration, external data is included in, or rather, added to the data set. This inclusion slightly improves performance, but also requires another API call, which is also expensive, due to the restructuring of the X.com (formerly Twitter) API.[11]

A recent study by Primig [12] from 2022 looked at the perception of fact-checkers and fact-checking services in the study population. The author found that, while higher trust in media correlates with trust in fact-checking, there is a significant part of the population who view fact-checking services as propaganda tools of the established government. To increase trust in the system, its purpose needs to be clearly stated: *which is to assist users in verifying rumors using official sources*. Those users who distrust and reject official sources out of hand will not find the information provided by our system to be helpful.

6. Conclusion and Perspectives for Future Work

In this paper, we have demonstrated the ability of our proposed setup to generally accurately classify whether official sources SUPPORT or REFUTE unseen rumors in a zero-shot fashion, using the data provided by the task organizers. In a real-world application, some considerations would have to be made with respect to operational aspects like computation costs, as LLMs are expensive to use “at scale". Model selection could also have a significant impact (especially “closed-source" models), as discussed in Section 5.3 .

In future work, improvements can be made, and extensions of the system need to be checked for performance improvements. Intuitive areas for further experimentation and development are:

[10] https://platform.openai.com/docs/changelog
[11] https://developer.x.com/en/docs/twitter-api/getting-started/about-twitter-api

- Do different embedding models for retrieval influence the performance of the verification stage? Do they significantly influence distribution of answers (for example, are there less or more NOT ENOUGH INFO predictions using another embedding model)?
- How can the retrieval stage be improved? Retrieval is essential for any fact-checking system to be able to judge a claim, as the verification stage relies on relevant evidence.
- How well does the system generalize to other domains and social media platforms? The datasets used were mainly focused on a specific geographical region, auto-translated from Arabic, and the topics of the statement-claim pairings were overall relatively topically similar.
- Different translation systems could also impact the reliability and effectiveness of any NLP-based approach, especially if the approach expects English data (like our approach does) and data from other languages has to be automatically translated.
- Does including more metadata improve retrieval or verification performance? How should the different metadata types be included? For example, if a statement is a direct reply or a "quote tweet" of the original tweet containing the claim, it is intuitive that this type of metadata would signal increased relevance.
- Multi-modality: tweets don't only contain text content, but also sometimes images and video data. Does adding this additional information to the tweet content, for example via transcription or use of multimodal capabilities of modern LLMs, improve retrieval or verification performance?

References

[1] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, P. Przybyła, T. Caselli, G. Da San Martino, F. Haouari, C. Li, J. Piskorski, F. Ruggeri, X. Song, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab: Check-worthiness, subjectivity, persuasion, roles, authorities and adversarial robustness, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. Di Nunzio, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[2] P. Nakov, H. Mubarak, N. Babulkov, Overview of the CLEF-2022 CheckThat! Lab Task 2 on Detecting Previously Fact-Checked Claims, in: Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, 2022.

[3] N. Pröllochs, Community-Based Fact-Checking on Twitter's Birdwatch Platform, Proceedings of the International AAAI Conference on Web and Social Media 16 (2022) 794–805. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/19335. doi:10.1609/icwsm.v16i1.19335.

[4] F. Haouari, T. Elsayed, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab Task 5 on Rumor Verification using Evidence from Authorities, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CLEF 2024, Grenoble, France, 2024.

[5] T.-Y. Liu, Learning to Rank for Information Retrieval, Foundations and Trends® in Information Retrieval 3 (2009) 225–331. URL: https://www.nowpublishers.com/article/Details/INR-016. doi:10.1561/1500000016, publisher: Now Publishers, Inc.

[6] 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. doi:10.1007/978-3-031-56069-9_62.

[7] F. Haouari, T. Elsayed, Are authorities denying or supporting? Detecting stance of authorities towards rumors in Twitter, Social Network Analysis and Mining 14 (2024) 34. URL: https://doi.org/10.1007/s13278-023-01189-3. doi:10.1007/s13278-023-01189-3.

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer,

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, 2019. URL: http://arxiv.org/abs/1910.13461. doi:10.48550/arXiv.1910.13461, arXiv:1910.13461 [cs, stat].

[9] B. MacCartney, S. U. C. S. Department, Natural Language Inference, Stanford University, 2009. URL: https://books.google.at/books?id=F55EAQAAIAAJ.

[10] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, C. Heidecke, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, L. Kaiser, A. Kamali, I. Kanitscheider, N. S. Keskar, T. Khan, L. Kilpatrick, J. W. Kim, C. Kim, Y. Kim, J. H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C. M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S. M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. d. A. B. Peres, M. Petrov, H. P. d. O. Pinto, Michael, Pokorny, M. Pokrass, V. H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F. P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J. F. C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J. J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. J. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, L. Kondraciuk, GPT-4 Technical Report, 2024. URL: http://arxiv.org/abs/2303.08774. doi:10.48550/arXiv.2303.08774, arXiv:2303.08774 [cs].

[11] L. Smars, FACTSHEET: Understanding the Promise and Limits of Automated Fact-Checking, 2018. URL: https://www.digitalnewsreport.org/publications/2018/factsheet-understanding-promise-limits-automated-fact-checking/.

[12] F. Primig, The Influence of Media Trust and Normative Role Expectations on the Credibility of Fact Checkers, Journalism Practice 18 (2024) 1137–1157. URL: https://doi.org/10.1080/17512786.2022.2080102. doi:10.1080/17512786.2022.2080102, publisher: Routledge _eprint: https://doi.org/10.1080/17512786.2022.2080102.

[13] F. Haouari, T. Elsayed, W. Mansour, Who can verify this? Finding authorities for rumor verification in Twitter, Information Processing & Management 60 (2023) 103366. URL: https://www.sciencedirect.com/science/article/pii/S0306457323001036. doi:10.1016/j.ipm.2023.103366.

A. Online Resources

The github repository can be found at github.com/LuisKolb/clef-2024-authority. The repository includes all the different components we used for our experiments, and the scripts used to produce our results for the CheckThat! Task 5 submission.

B. Glossary

In this paper, we use some specific words to describe specific concepts:

- “claim": the individual text snippet/sentence(s) that is to be verified (using authority sources)
- “rumor": used interchangeably with claim (in the dataset, every rumor consists of a claim and several statements, and has a "rumor_id")
- “statement": a social media post, in this context posted by an authority accounts
- “evidence": a statement relevant to a specific claim
- “authority": typically official government social media accounts, but also sometimes the individual person a claim is about, and whose social media posts can be used to verify that claim

B.1. Verification Experiment Results and Tables

Configurations with the same score are assigned the same rank, as they produced the same results. Some column names are abbreviated for layout width reasons:

- MF1: Macro-F1 score
- SMF1: Strict-Macro-F1 score
- Pre: whether Preprocessing was used
- ExtData: whether External Data (Author Name and Bio) was used
- IgnNEI: whether NOT ENOUGH INFO (NEI) pairwise predictions are Included in the decision weighting (False) or are Ignored in the decision (True)

Table 5
Experiment results for verification configurations/feature combinations on the dev set. Sorted by Macro-F1. Ranks 1-5.

Rank	MF1	SMF1	Retrieval	Verification	Pre	ExtData	Scale	Norm	IgnNEI
1	0.872	0.856	Embeddings	OPENAI	False	True	False	False	True
1	0.872	0.856	Embeddings	OPENAI	False	True	False	False	False
1	0.872	0.856	Embeddings	OPENAI	False	True	True	False	True
1	0.872	0.856	Embeddings	OPENAI	False	True	False	True	True
1	0.872	0.856	Embeddings	OPENAI	False	True	False	True	False
1	0.872	0.856	Embeddings	OPENAI	False	True	True	True	True
1	0.872	0.872	PyTerrier	OPENAI	False	True	False	False	True
1	0.872	0.872	PyTerrier	OPENAI	False	True	False	False	False
1	0.872	0.872	PyTerrier	OPENAI	False	True	True	False	True
1	0.872	0.872	PyTerrier	OPENAI	False	True	True	False	False
1	0.872	0.872	PyTerrier	OPENAI	False	True	False	True	True
1	0.872	0.872	PyTerrier	OPENAI	False	True	False	True	False
1	0.872	0.856	Embeddings	OPENAI	False	False	False	False	True
1	0.872	0.856	Embeddings	OPENAI	False	False	False	False	False
1	0.872	0.856	Embeddings	OPENAI	False	False	True	False	True
1	0.872	0.856	Embeddings	OPENAI	False	False	False	True	True
1	0.872	0.856	Embeddings	OPENAI	False	False	False	True	False
1	0.872	0.856	Embeddings	OPENAI	False	False	True	True	True
1	0.872	0.872	PyTerrier	OPENAI	False	False	False	False	True
1	0.872	0.872	PyTerrier	OPENAI	False	False	False	False	False
1	0.872	0.872	PyTerrier	OPENAI	False	False	True	False	True
1	0.872	0.872	PyTerrier	OPENAI	False	False	True	False	False
1	0.872	0.872	PyTerrier	OPENAI	False	False	False	True	True
1	0.872	0.872	PyTerrier	OPENAI	False	False	False	True	False
1	0.872	0.856	Embeddings	OPENAI	True	True	True	True	True
1	0.872	0.872	PyTerrier	OPENAI	True	True	False	False	True
1	0.872	0.872	PyTerrier	OPENAI	True	True	False	False	False
1	0.872	0.872	PyTerrier	OPENAI	True	True	True	False	True
1	0.872	0.872	PyTerrier	OPENAI	True	True	True	False	False
1	0.872	0.872	PyTerrier	OPENAI	True	True	False	True	True
1	0.872	0.872	PyTerrier	OPENAI	True	True	False	True	False
1	0.872	0.856	Embeddings	OPENAI	True	False	False	False	True
1	0.872	0.856	Embeddings	OPENAI	True	False	False	False	False
1	0.872	0.856	Embeddings	OPENAI	True	False	True	False	True
1	0.872	0.856	Embeddings	OPENAI	True	False	False	True	True
1	0.872	0.856	Embeddings	OPENAI	True	False	False	True	False
1	0.872	0.856	Embeddings	OPENAI	True	False	True	True	True
2	0.855	0.841	PyTerrier	OPENAI	True	False	False	False	True
2	0.855	0.841	PyTerrier	OPENAI	True	False	False	False	False
2	0.855	0.841	PyTerrier	OPENAI	True	False	True	False	True
2	0.855	0.841	PyTerrier	OPENAI	True	False	True	False	False
2	0.855	0.841	PyTerrier	OPENAI	True	False	False	True	True
2	0.855	0.841	PyTerrier	OPENAI	True	False	False	True	False
3	0.831	0.816	Embeddings	OPENAI	True	True	False	False	True
3	0.831	0.816	Embeddings	OPENAI	True	True	False	False	False
3	0.831	0.816	Embeddings	OPENAI	True	True	True	False	True
3	0.831	0.816	Embeddings	OPENAI	True	True	False	True	True
3	0.831	0.816	Embeddings	OPENAI	True	True	False	True	False
4	0.820	0.820	PyTerrier	OPENAI	False	True	True	True	True
4	0.820	0.820	PyTerrier	OPENAI	False	False	True	True	True
4	0.820	0.820	PyTerrier	OPENAI	True	True	True	True	True
5	0.806	0.790	PyTerrier	OPENAI	True	False	True	True	True

Table 6
Experiment results for verification configurations/feature combinations on the dev set. Sorted by Macro-F1. Ranks 6-30.

Rank	MF1	SMF1	Retrieval	Verification	Pre	ExtData	Scale	Norm	IgnNEI
6	0.723	0.699	Embeddings	OPENAI	False	True	True	True	False
6	0.723	0.699	Embeddings	OPENAI	False	False	True	True	False
7	0.713	0.696	Embeddings	LLAMA	True	True	True	True	False
8	0.700	0.677	Embeddings	OPENAI	True	False	True	True	False
9	0.691	0.661	Embeddings	LLAMA	True	True	True	False	True
9	0.691	0.661	Embeddings	LLAMA	True	True	True	True	True
10	0.682	0.657	PyTerrier	LLAMA	False	True	True	False	True
10	0.682	0.657	PyTerrier	LLAMA	False	True	True	False	False
11	0.661	0.620	Embeddings	LLAMA	False	True	True	True	True
12	0.661	0.632	Embeddings	LLAMA	True	True	False	False	True
12	0.661	0.632	Embeddings	LLAMA	True	True	False	False	False
12	0.661	0.632	Embeddings	LLAMA	True	True	False	True	True
12	0.661	0.632	Embeddings	LLAMA	True	True	False	True	False
13	0.657	0.634	Embeddings	OPENAI	True	True	True	True	False
14	0.651	0.620	PyTerrier	LLAMA	False	True	True	True	True
14	0.651	0.620	PyTerrier	LLAMA	False	False	True	True	True
15	0.647	0.634	PyTerrier	LLAMA	True	True	True	False	True
15	0.647	0.634	PyTerrier	LLAMA	True	True	True	False	False
16	0.645	0.628	PyTerrier	LLAMA	True	True	True	True	True
17	0.643	0.619	PyTerrier	LLAMA	False	False	True	False	True
17	0.643	0.619	PyTerrier	LLAMA	False	False	True	False	False
18	0.640	0.611	PyTerrier	LLAMA	False	True	False	False	False
18	0.640	0.611	PyTerrier	LLAMA	False	True	False	True	False
19	0.637	0.606	Embeddings	LLAMA	True	False	True	False	True
19	0.637	0.606	Embeddings	LLAMA	True	False	True	True	True
20	0.637	0.617	Embeddings	LLAMA	True	False	True	True	False
21	0.630	0.613	PyTerrier	LLAMA	True	False	True	True	True
22	0.628	0.605	Embeddings	OPENAI	False	True	True	False	False
22	0.628	0.605	Embeddings	OPENAI	False	False	True	False	False
23	0.627	0.598	PyTerrier	LLAMA	False	True	False	False	True
23	0.627	0.598	PyTerrier	LLAMA	False	True	False	True	True
23	0.627	0.598	Embeddings	LLAMA	True	False	False	False	True
23	0.627	0.598	Embeddings	LLAMA	True	False	False	False	False
23	0.627	0.598	Embeddings	LLAMA	True	False	False	True	True
23	0.627	0.598	Embeddings	LLAMA	True	False	False	True	False
24	0.626	0.612	PyTerrier	LLAMA	True	True	False	False	False
24	0.626	0.612	PyTerrier	LLAMA	True	True	False	True	False
25	0.624	0.611	PyTerrier	LLAMA	True	False	True	False	True
25	0.624	0.611	PyTerrier	LLAMA	True	False	True	False	False
26	0.618	0.576	Embeddings	LLAMA	False	True	False	False	True
26	0.618	0.576	Embeddings	LLAMA	False	True	False	False	False
26	0.618	0.576	Embeddings	LLAMA	False	True	True	False	True
26	0.618	0.576	Embeddings	LLAMA	False	True	False	True	True
26	0.618	0.576	Embeddings	LLAMA	False	True	False	True	False
27	0.615	0.601	PyTerrier	LLAMA	True	True	False	False	True
27	0.615	0.601	PyTerrier	LLAMA	True	True	False	True	True
28	0.606	0.583	Embeddings	OPENAI	True	True	True	False	False
29	0.605	0.577	PyTerrier	LLAMA	False	False	False	False	False
29	0.605	0.577	PyTerrier	LLAMA	False	False	False	True	False
30	0.595	0.575	Embeddings	LLAMA	True	False	True	False	False

Table 7
Experiment results for verification configurations/feature combinations on the dev set. Sorted by Macro-F1. Ranks 31-42.

Rank	MF1	SMF1	Retrieval	Verification	Pre	ExtData	Scale	Norm	IgnNEI
31	0.590	0.562	PyTerrier	LLAMA	False	False	False	False	True
31	0.590	0.562	PyTerrier	LLAMA	False	False	False	True	True
31	0.590	0.576	PyTerrier	LLAMA	True	False	False	False	True
31	0.590	0.576	PyTerrier	LLAMA	True	False	False	True	True
32	0.585	0.571	PyTerrier	LLAMA	True	False	False	False	False
32	0.585	0.571	PyTerrier	LLAMA	True	False	False	True	False
33	0.557	0.525	Embeddings	LLAMA	False	True	True	True	False
34	0.545	0.521	Embeddings	OPENAI	True	False	True	False	False
35	0.537	0.519	Embeddings	LLAMA	True	True	True	False	False
36	0.537	0.511	PyTerrier	LLAMA	True	False	True	True	False
37	0.489	0.489	PyTerrier	OPENAI	True	True	True	True	False
38	0.485	0.457	PyTerrier	LLAMA	False	True	True	True	False
38	0.485	0.457	PyTerrier	LLAMA	False	False	True	True	False
39	0.468	0.444	PyTerrier	LLAMA	True	True	True	True	False
40	0.453	0.420	Embeddings	LLAMA	False	True	True	False	False
41	0.413	0.413	PyTerrier	OPENAI	False	False	True	True	False
41	0.413	0.413	PyTerrier	OPENAI	True	False	True	True	False
42	0.394	0.394	PyTerrier	OPENAI	False	True	True	True	False