TU WIEN Informatics

Fakten-Checks anhand von Nachweisen von Autoritäts-Accounts

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Luis Kolb, BSc
Matrikelnummer 01622731

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Allan Hanbury

Wien, 1. September 2024

Luis Kolb

Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Informatics

Fact-Checking Claims using Authority Evidence

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Luis Kolb, BSc

Registration Number 01622731

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury

Vienna, September 1, 2024

Luis Kolb

Allan Hanbury

Erklärung zur Verfassung der Arbeit

Luis Kolb, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. September 2024

Luis Kolb

Danksagung

Zuallererst will ich meiner Familie danken: meiner Mutter Traudi, der mit Sicherheit einige der zu-poetischen Ausdrücke zu verdanken sind, die ich aus dem Draft wieder streichen musste, aber trotzdem gern getippt habe. Meinem Vater Helmut, dem ich überhaupt mein Interesse nicht nur für alles Technische zu verdanken habe, sondern auch für Rock 'n' Roll und Gitarrenmusik ganz generell. Meine nicht mehr so kleine Schwester Anna, auf die mich immer stützen konnte, vor, während und sicher noch lange nach dem Studium. Ihr wart unermüdlich für mich da, und ohne euch würde dieses Dokument nicht existieren.

Außerdem darf ich meinem Betreuer Allan Hanbury danken, für die stete Unterstützung nicht nur durch den Schreibprozess, sondern auch um die CLEF-Konferenz in Frankreich, welche eine essenzielle Rolle in der Entstehung dieser Arbeit hatte. Der internationale Austausch in Grenoble hat wahrlich meinen Horizont erweitert, und dank den Kollegen und Kolleginnen der TU Wien war es eine unvergessliche Woche. Moritz, Florina, Patrick, Alaa – es war mir eine Freude!

Zuletzt bedanke ich mich bei meinen Freunden, die mich ein Stück des langen Weges begleitet haben, und besonders bei jenen, die es immer noch tun.

Kurzfassung

Soziale Medien sind effektive Mechanismen zur schnellen Verbreitung von Desinformation und Gerüchten, gezielt oder ohne direkte Absicht dazu. Oft ist es aufwendig, Kontext für die Behauptungen auf diesen Plattformen zu finden. Das Volumen von Informationen aller Art, welche das Internet füllen, erschwert diese Aufgabe weiters.

Wir präsentieren einen Ansatz zur automatisierten Überprüfung von Behauptungen in Social Media Postings unter Verwendung von Nachweisen, welche direkt aus den Kanälen von Autoritäten auf dem gegebenen Thema stammen. Mithilfe dieser Nachweise kann unser Ansatz Behauptungen automatisch als "unterstützt" oder "widerlegt" gekennzeichnet werden, basierend auf den Postings der Autoritäts-Accounts.

In unserem zweistufigen Ansatz filtern wir relevante Postings zu einer Behauptung aus den Kanälen von Accounts, die "Autorität" über das Thema der Behauptung haben. Wir vergleichen mehrere Methoden zum Auffinden dieser relevanten Postings, von simplen lexischen Methoden zu komplexeren Embedding und Transformer-basierten Methoden.

Wir untersuchen auch die Effektivität und Verlässlichkeit von "Large Language Models" (LLMs) zur automatischen Beurteilung des Verhältnisses von Posting und Behauptung, und präsentieren ermutigende Ergebnisse. Effektive und flexible Methoden für diese Art der "Natural Language Inference" sind essentiell, um diese automatisierten Klassifikationen zu treffen - besonders auf Plattformen, wo ein informeller Sprachstil die Norm ist. Die Größe eines Sprachmodells spielt eine wichtige Rolle in der Anwendbarkeit auf diese Aufgabe. Dazu präsentieren wir Vergleiche zwischen verschiedenen Modellgrößen und deren Effektivität auf unserem Datensatz.

Zusätzlich inkludieren wir noch einige unserer Beiträge zur CLEF 2024 Konferenz, im Rahmen derer wir am CheckThat! Lab 2024 teilgenommen haben.

Schlussendlich zeigen wir mögliche Erweiterungen und wichtige Aspekte im Kontext des automatisierten Fact-Checkings zu unserem Ansatz auf.

Abstract

Social media is an opportune delivery vehicle for misinformation to spread quickly and effectively. Finding context for extraordinary claims made on these platforms can be challenging, given the volume of the information available online.

We present an approach to claim verification in social media using evidence retrieved from the social media accounts of authorities on the claim topic. Using our approach, claims can be automatically labeled as “refuted” or “supported” based on the social media post timelines from relevant authorities.

In a two-stage approach we first compare typical retrieval methods to search through a social media timeline and find posts that are relevant to a given claim. Our comparison includes both simple lexical retrieval methods, as well as more complex embedding and transformer-based methods.

Further, we investigate the effectiveness and reliability of Large Language Models (LLMs) as “Natural Language Inference” (NLI) agents, and find promising results. Effective NLI methods are crucial the the development of an automated claim verification approach, especially in a setting where usage of informal language is the norm. The size of a model plays an important role in its performance on this task. To this end, we present comparisons between a range of model sizes and their respective performance on our chosen dataset.

Additionally, we include some of our contributions to the 2024 CLEF conference, where we participated in at the 2024 CheckThat! Lab.

Finally, we discuss possible extensions and caveats to our approach.

Contents

CHAPTER 1

Introduction

1.1 Motivation

As the world is ever more inundated with what can be called rumors at best and outright misinformation at worst, easy access to verification tools is required to promote sovereignty over the digital space.

It is generally accepted that social media is prone to spreading of misinformation. Rumors on social media spread fast and wide [VRA18], and – whether intentionally or unintentionally – such rumors can both influence public discourse and impact individuals. What is more, sowing misinformation is now fully part of hybrid warfare [Sta22], and one way of doing this is posting on social media platforms. Given the increasing volume of rumors on social media, automatic detection and verification of such rumors seems of utmost importance

In this thesis, we propose an approach to verify claims (or "rumors") posted on social media using evidence (social media posts relevant to that claim) obtained from accounts that have specific authority over the topic of that claim.

To find answers to our research questions, we work with data from authority Twitter accounts to demonstrate the effectiveness of the implementation of our proposed approach. The dataset was retrieved and annotated by the CheckThat! Lab Task 5 organizers. We will refer to X.com as "Twitter", as it was formerly known, and posts on X.com as "tweets" (or just "posts" more generally), as we are working with data posted to and obtained from this platform.

We participated in the CheckThat! Lab Task 5 [HES24] of the 2024 CLEF conference[1]. The paper [KH24] we submitted to the conference was later published in the working notes of CLEF 2024 ("Working Notes of CLEF 2024 - Conference and Labs of the

[1] clef2024.imag.fr

Evaluation Forum," 2024) [FFGGSdH24]. While our working notes cover the results of the experiments conducted during our Lab participation, this thesis has a different objective, the research questions being presented in the next subsection (Section 1.2).

From this paper published in the CLEF working notes, we include our findings where relevant to our research in our thesis. The parts of our thesis referencing our CLEF paper are clearly marked as such, citing our paper as a source. These parts are mainly our own submission to the leaderboard, other participants' submissions and approaches to the task, as well as some evaluation scores. We omit other parts of our CLEF paper in this thesis, like the experiment results for specific models we used in the paper, but not in the thesis.

1.2 Research Questions

In this thesis, we answer the following three research questions ("RQs"):

- **RQ1:** 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?
  - (For this RQ, we will assume that the tweets included in the timelines are limited to the period surrounding the time the rumor was published.)
- **RQ2:** To what extent can a claim, given a list of tweets ("evidence"), accurately be identified as being supported by the evidence (true), being refuted by the evidence (false), or being unverifiable (not enough evidence to verify it being available)?
- **RQ3:** 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?
  - (For this RQ, we will assume that a collection of timelines is given, as retrieval of whole "authority timelines" is not included in the scope of this thesis.)

1.3 Structure of the Thesis

This thesis consists of the following chapters:

- **Chapter 2** provides some background on and explains key concepts for the techniques used in this thesis.
- **Chapter 3** discusses related work in the field of fact-checking.
- **Chapter 4** introduces the data and performance measures we use and describes how we answer our research questions from Section 1.2.

- **Chapter 5** describes our approach to evidence retrieval, and **Chapter 6** describes our approach to verification of claims using retrieved evidence.
- **Chapter 7** reports on and discusses the results of our experiments and includes the answers to our research questions.
- **Chapter 8** summarizes and discusses our findings and suggests avenues for further research.

CHAPTER 2

Background

To provide some background, we will briefly introduce some key concepts from the domains of Information Retrieval, Natural Language Processing and Machine Learning this thesis will be drawing on.

2.1 Natural Language Processing

Natural Language Processing (NLP) describes the discipline of working with text created by humans ("natural language") using computers. Since computers communicate and store data differently from humans, text must be represented in such a way that computers can work with the data. A definition offered by E. Liddy in 2001 describes NLP as follows:

> "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications." [Lid01](p.3)

Most of the problems we are tackling in this thesis fall squarely into this definition: identifying semantically relevant documents in a larger collection and analyzing pieces of texts to arrive at a conclusion based on the semantic content of these same texts. To make this task feasible, we employ several computational techniques. Section 2.2 describes some Information Retrieval techniques, which often rely on NLP techniques (such as text embeddings or text preprocessing, which are described there) to represent text in ways a machine can handle more easily. Similarly, Section 2.4 introduces recent techniques that had a sizable impact on the field of NLP.

2.2 Information Retrieval

A popular definition of Information Retrieval is provided by Manning et al., from 2008:

> “Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)” [MRS08](p.1).

The “information need” in the context of this thesis is “relevance”. Relevance in this context means that the documents to be retrieved are in some way relevant with regard to some input. Different applications of information retrieval imply different use cases and their own definitions of documents, input and relevance. Prominent examples of information retrieval are web search engines, where the input is the user’s search text, the data to be retrieved is a website, and relevance is a balance of different factors such as whether the user’s search text appears on the website, algorithmically calculated scores like Google’s PageRank, and many more.

In the context of this thesis, we want to retrieve “documents” from a set of documents (also called a “corpus”), the content of which is about, or contains, the same topic or entities as a given input text (also called “query”). The query is a set of sentences (originally tweets), which make up the claim. The documents are in the form of tweets as well, and the corpus is a collection of timelines from authority sources, which ultimately contain tweets. A definition of what we consider an authority is given in Section 4.2.1.

To facilitate this search, some score must be calculated for every pairing of query and document from the corpus. The documents with the best score are, ideally, the most relevant. There are many components that go into the calculation for such a “relevance score” (and approaches can also be combined!), for example:

- **Lexical matching:** does the text (or part of the text) in the query appear in the text of a document in the corpus? If yes, that document from the corpus is more likely to be relevant to the query.
- **Frequency:** do many documents in the corpus contain some part of the query text? If yes, that part of the query text is not useful in differentiating between relevant and irrelevant documents, and it should have little impact on the score.
- **Preprocessing:** to improve results from the two above components, text can also be changed before the score is calculated. Well known techniques include:
  - Stopword removal to reduce noise. Typical stopwords in the English language are, for example: “the”, “a”, etc. – words that usually don’t contribute any relevant signal.

  - Stemming (or Lemmatization), where terms are transformed into a common form, for example by truncating suffixes (where previously non-matching inflections of the same term would not match) or transforming words into a normalized form (a “lemma”). We don’t make use of these methods in this thesis, but they are commonly used to improve the performance of lexical retrieval algorithms.

- **Query rewriting** in the context of Natural Language Processing (NLP) refers to a family of techniques used to improve the results of retrieval with respect to some metric. For example:
  - To increase precision one could use “Query Segmentation” like described by Bergsma and Wang [BW07]. Query segmentation is a technique where the query is split up into individual semantic units, which is useful to disambiguate queries. Depending on how words in a query are grouped together, the different possible groupings imply different semantic meanings, which influences what documents from the corpus could be relevant.
  - To increase recall, “Query Expansion” could be used: existing databases can be used to insert additional text and improve lexical matching, for example by including synonyms of existing words in the query or writing out an abbreviation, which might not have contributed to the search if the query uses an abbreviation, but the desired document only contains the non-abbreviated, full term.

2.2.1 Lexical Retrieval

There are many lexical methods available to be used in retrieval. Lexical methods generally perform the task of retrieving relevant documents by relying on the individual terms (or “words”) in a document. In this thesis, we use two of the most commonly used algorithms: BM25 and TF-IDF, which we will describe here.

TF-IDF relies on the fact that if a document contains many of the same terms as the query, it is likely more similar than those documents which contain few, or no mentions at all, of the term. This measure is also called “term frequency”. There are, of course, terms which appear in many documents, such as “the” or “a” (also called “stopwords”). Terms which appear in many documents are likely less discriminatory and less helpful in finding a few related documents from among all the documents. Thus, terms which occur frequently are assigned a lower “weight”. Terms which appear rarely in the collection of all documents (and in the query) are more informative. Increasing the score of documents which contain “rarer” terms is called “inverse document frequency”. Together, these approaches form the algorithm called TF-IDF (“Term Frequency, Inverse Document Frequency”).

BM25 is a specific configuration of a retrieval function. Its usage in information retrieval is wide-spread, and it is cheap to compute. BM25 works similarly to TF-IDF, with variations and some tuning parameters set manually by the creators.

Generally, when a matching function relies on “raw” terms, preprocessing the text is important. Typical preprocessing approaches are described in the previous Section.

There are also many other methods to improve the performance of lexical matching approaches, which are more complex in their implementation, like spelling out abbreviations using a knowledge base to include more terms and potentially increase matchability. However, the basic issue remains that a claim and an authority statement might be discussing the same event or thing, but simply use different language or terms (for example, formal speech versus common vernacular), which would be more challenging to an approach based on lexical matching.

2.2.2 Embeddings

The paper introducing the model architectures for the “Word2Vec” algorithm was published in 2013 by Mikolov et al. [MCCD13]. Word2Vec models text by transforming words into vectors in a high-dimensional vector space. The benefit of this technique is that semantic relationships between words are captured. Words appearing in similar contexts are mapped to vectors which are nearby each other in this high-dimensional vector space, and the distance can be computed. As words are represented as vectors, they can also be arithmetically combined. An example Mikolov et al. give in their paper is: “Paris - France + Italy = Rome”, or this example:

> “To find a word that is similar to small in the same sense as biggest is similar to big, we can simply compute vector X = vector(”biggest”) - vector(”big”) + vector(”small”). Then, we search in the vector space for the word closest to X measured by cosine distance and use it as the answer to the question (we discard the input question words during this search). When the word vectors are well trained, it is possible to find the correct answer (word smallest) using this method” [MCCD13](p.5).

Techniques like Word2Vec are called “non-contextual embeddings”, and they operate on words as the smallest unit. If a word is not in the originally trained vocabulary, Word2Vec cannot represent it. Another approach is subword tokenization, where the text is “tokenized”, with each token then being vectorized. Since the smallest possible token is a letter of the alphabet, all words can be represented, if not by word-level or multiple-character tokens, then by single letter tokens at least. This deals with the “out-of-vocabulary” problem and this “contextual embedding” approach is what BERT-like models use. The vector representations of tokens are learned during training, which is also true for Word2Vec, where vector representations of words are learned and saved to a lookup table.

The semantic relationships between tokens are captured in both approaches. To compute the “semantic” distance between vectors A and **B**, we can use cosine similarity:

$$CosineSimilarity(A, B) := cos(\theta) = \frac{A \cdot B}{|A||B|}$$

In this thesis, we experiment with multiple “contextual embedding” models in the retrieval stage, using cosine similarity as a similarity measure. This is also related to Natural Language Processing (NLP), for which we gave some background in Section 2.1. Representing text via embeddings is an NLP technique, and with a similarity measure like cosine similarity it can be used for IR purposes.

2.3 Natural Language Inference

First introduced as a concept in 2009 by McCartney [Mac09], Natural Language Inference (NLI) is a task both in the domain of natural language processing and typically machine learning. The goal, usually, is to infer “ENTAILMENT”, “CONTRADICTION” or “NOT ENOUGH INFORMATION” between two pieces of text (such as a statement and a claim, or a premise and a statement).

For example, does the sentence “Experts say that the sky is usually blue.” entail or contradict the claim “Skies have been green all day today!”? This type of task is similar to classification, where the target label is one of “ENTAILMENT”, “CONTRADICTION” or “NOT ENOUGH INFORMATION”. This task is hard because:

- Natural language is generally complex and culture-dependent, sometimes containing sarcasm, jokes, slang, and so on, increasing complexity.
- Logical reasoning engines usually require specific structures (such as a graph or a semantic web) to work. Extracting, for example, a graph-like structure from two sentences mapping all the objects, relations, properties (adjectives), antonyms, etc. is not trivial. The system then must combine the two graphs, align nodes, and so on to arrive at an answer. This step can involve techniques like Named Entity Recognition to increase matchability of subjects across sentences or graphs.

Often, NLI as a task is compared to Recognizing Textual Entailment (RTE), which in turn is sometimes thought of as a predecessor to NLI. In RTE, the task is to recognize if the meaning of one text is entailed by another. However, the concept of contradiction does not formally exist in this task. RTE is described in detail in the 2010 article by Dagan et al. [DDMR10].

2.4 Deep Learning and Transformers

The approaches we employ in this thesis involve, at some stages, the use of pre-trained language models. These models were created via a process called “Deep Learning” (DL). We will first give a definition of “Machine Learning” (ML), cited from B. Mahesh:

> “Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed.” [Mah19](p.381)

According to Wikipedia[1], the phrase “without being explicitly programmed“ was originally coined in 1959, but it first appeared in a publication in 1996 [KBAK96]. Moving on, a widely used definition of Deep Learning (DL) is the following:

> “Deep learning [. . .] is learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.” [ZYL+18](p.233)

This definition is presumably a paraphrase of the first paragraph of the Wikipedia entry, which cites a paper called “Deep Learning” by Y. LeCun [LBH15], which itself, however, does not give a designated definition of DL. The paper by Zhang et al. [ZYL+18] discusses problems with this and other definitions of DL, and highlights areas where the definitions are lacking.

For the purposes of this thesis, we will consider DL as a specific type of Machine Learning (ML), in which features and relations of features are learned from a large corpus of training data. The learned features and relations in a language model are useful to us because we can use the model to process and generate natural language texts.

One of the most revolutionary papers in the field of machine learning and deep learning in the last few years was the paper introducing the transformer architecture by Vaswani et al. [VSP+17], with the paper introducing BERT by Devlin et al. [DCLT19] releasing a year later and transforming NLP research and rapidly advancing many disciplines within NLP. The paper introducing BART by Lewis et al. [LLG+19] was released another year later, and yet again advanced many of the disciplines in the field of NLP. Generally, transformers rely on the attention mechanism to components in a sequence, like words (or tokens) in a sentence, to influence their generation. For language models, this mechanism allows relevant words in a sentence to influence generated tokens, enabling long-range dependencies and references to context often found in natural language.

Even more recently, another advance in transformer-based approaches has rapidly advanced the possibilities in the field of NLP: Large Language Models (LLMs). Large Language Models like, for example, Llama-3.1 [DJP+24] differ from the language models like BERT above in terms of capability and versatility, as well as in terms of scale and training data. While BERT-based models usually have to be fine-tuned on task-specific data to be useful on a specific task, LLMs are more versatile and more capable of performing a wide range of tasks.

LLMs are usually transformer-based deep learning models, the first of which arguably was introduced in 2020 with the GPT-3 model [BMR+20] . Via the attention mechanisms

[1] wikipedia.org/wiki/Machine_learning

built into their architecture (to take context into account) and huge amounts of training data, LLMs learn to model one or more languages better than any other approach known today. Since the initial GPT-3 model, many more models have been introduced, and recent rapid increases in the trainable parameter count of LLMs have dramatically improved their general performance [ZZL$^+$23]. During training, LLMs learn to model a language such that generative models can generate their own text. Importantly, the text this class of model can generate adheres to the (grammatical and semantic) rules of the language(s) it is trained on. Self-attention in the model architecture makes the model respect related input data.

Text generation applications of these models excel at many tasks such as code generation, creative writing and document summarization. We theorize that a performant enough LLM will be able to perform the basic inference tasks required to perform the role of a verification component given a claim-evidence pairing. As we described earlier, performing this role is very similar to performing the task of Natural Language Inference (NLI). In our experiments, we apply LLMs to the task of NLI, described in Section 2.3, and report the results in Section 7.

CHAPTER 3

Related Work

In this section, we cover relevant papers and methods positioned in the field of general fact-checking and adjacent topics.

3.1 Traditional Fact-Checking Approaches

The task of "Fact-Checking" in the field of computer science is introduced in a paper by Riedel and Vlachos in 2014 [VR14]. They define it as "[. . .] the assignment of a truth value to a claim made in a particular context" [VR14](p.19). "Traditional" fact-checking already happened before this paper, of course, for example by media organizations and journalists in, for example, a political context [GA19] . At the time their paper was published, there also existed professional fact-checking services reliant on humans, which provided fact-checks for popular rumors, like PolitiFact and Snopes.

Riedel and Vlachos consider this task an ordinal classification task, with stages that map neatly onto well-established NLP disciplines:

- Claim extraction can be thought of as a sentence classification problem
- Question answering over a knowledge base and information extraction to obtain a basis of context for the verdict
- Logic-based textual entailment (see Section 2.3 for a brief description of RTE, the predecessor of NLI) to obtain a verdict

Additionally, even in 2014 the authors pointed out that not only journalists are involved in both creating and fact-checking information, but that ordinary citizens could benefit from fact-checking information on their own, and that some citizens themselves have become sources of information that should be fact-checked.

In their survey from 2022, Guo et al. [GSV22] also define claim verification as a RTE problem. They list typical retrieval strategies, some of which we implemented in our paper:

- Commercial search APIs
- Lucene indices, which we included in our experiments via PySerini (Sparse Retrieval)
- Entity linking
- Ranking functions like TF-IDF and BM25, which we included in our experiments
- Learned representations, which we included in our experiments as Text Embeddings
- Re-ranking via methods like stance detection

The verification of more complex claims may require multiple pieces of evidence, which must be combined in some way. For example, the conceptually simplest approach is to concatenate the pieces of evidence, and then proceed with a method like RTE as though there was only one piece of evidence. Guo et al. cite multiple papers that discuss possible approaches to dealing with this issue, most of which are graph-based, such as the system by Schlichtkrull et al. [SKO+21]. For our system, we take an alternative approach of scoring separately and then combining verdicts as described in Section 6.

3.2 Additional Tasks Involved in Fact-Checking

In most fact-checking systems, there are additional tasks to consider depending on the intended application, also listed by Guo et al. [GSV22]:

- Claim detection and determining the check-worthiness of a claim: this task is outside the scope of our thesis, as the dataset we use already provides claims and associated evidence candidates (the timelines of the relevant authority accounts).
- Claim matching: if a claim is repeated, for example in a rumor that spreads around on social media, it can be beneficial to see if a claim has been fact-checked previously. If previous fact-checks can be reused, the work necessary to fact-check it again can be avoided. Zeng et al. [ZAZ21] summarize some of the approaches that can be applied to this task in their 2021 survey. Shaar et al. [SBDSMN20] discuss how this task can be formulated as a ranking retrieval task, and stress the importance of claim matching not only with respect to human fact-checkers and their limited resources, but also as an integral part of automated fact-checking systems.
- Justification for the predicted label: we do not implement this task in our thesis. There are multiple approaches summarized by Guo et al. in their survey [GSV22] to increase the explainability of a fact-checking system, though they all have their drawbacks. Briefly:

  - Highlighting tokens that heavily influence attention mechanics inside the verification model as an explanation. Using attention scores to produce explanations is problematic, as removing tokens with low or high attention scores can influence the label prediction in unpredictable ways. As such, justifications produced relying on attention scores can be unfaithful [JW19].
  - Rule-based systems relying on knowledge bases. This approach is limited by the knowledge base and availability of the information that can be used to create the justification. Any fact-check using information that does not exist in a previously existing knowledge base (usually in the form of a triple) cannot be justified.
  - Generating textual explanations for why a prediction was made. A paper from Atanasova et al. [ASLA20] is criticized by Guo et al. for "[. . .] assum[ing] fact-checking articles provided as input during inference, which is unrealistic" [GSV22](p.187). However, this approach is prone to producing plausible explanations even if the underlying prediction is wrong and not supported by the evidence, which can be thought of as a type of hallucination.

- Some fact-checking systems and datasets define their own "labels" for the outcome of a fact-check, like "pants-fire" [Wan17], "half-true" or "half-false". As such, there is no "official" set of labels a fact-checking system must use to classify claims. There is also the aspect of checking for support or contradiction, which as a set of labels has an entirely different type of meaning.

- Authority finding: in our thesis, we use evidence from authority social media accounts. As we use a dataset provided to us, we don't retrieve authorities and timelines as a whole ourselves. Haouari et al. published a paper in 2023 [HEM23] detailing approaches to the task of finding authorities.

3.3 Alternative Approaches to Fact-Checking

There are alternative approaches to the NLP-driven and machine-learning-driven methods. On Twitter, there exists a feature which was previously called "Birdwatch" and is now known as "Community Notes". Community Notes are created by Twitter users, and were originally intended do display missing context below misleading tweets (if enough trustworthy users, also called "contributors", give a relevant rating to that Community Note). This approach does not rely on professional human fact-checkers or end-to-end automated fact verification systems, and faces its own challenges:

- Biases and polarization in the sources: A community note must contain a link to a source of "evidence" to back up its contents. A 2024 study by Kangur et al. [KCS24] analyzes sources cited in community notes, and finds that there are relevant political divisions within the community, and that there are patterns in the sources that get cited. The most-cited sources are tweets on Twitter itself and

Wikipedia, both crowd-sourced platforms. Sources are mainly biased left-leaning and score high in factuality, which, according to the authors, suggest a left-leaning trend in the political makeup of the community of contributors. Right-leaning sources are found to be both generally lower in factuality and more supportive of tweets (as opposed to "correcting" a tweet) than left-leaning sources, which could imply the existence of echo-chambers among right-leaning contributors.

- Impact on the spread of misinformation: A study by Chuai et al. in 2023 [CTPL23] presents their findings on the impact of the Community Notes feature since its release in 2021. The authors found that the feature did not significantly impact the spread of misinformation (measured in terms of interactions with misleading tweets). A possible explanation could be the response time of the Community Note feature, which the authors suggest is too slow and takes too long to significantly impact the viral spread of misinformation on the platform.
- Influential accounts: A study by Pröllochs from 2022 [Pro22] finds that the community-driven approach has drawbacks when fact-checking tweets posted by influential accounts with a large following. Community Notes on tweets by these accounts, for example on a tweet by a prominent politician, are more likely to be perceived as incorrect or argumentative. Accounts with large followings, the study found, tend to have polarized and fragmented social networks, which presents a challenge to a crowd-sourced approach to fact-checking.

3.4 Semi-Automated Fact-Checking State-of-the-Art

Today, the most popular and effective methods to do automated fact-checking are based on both deep learning methods and graph-based methods, with pre-trained language models.

While quantification is not trivial due to a lack in standardization across fact-checking disciplines, a survey in 2023 by Vladika and Matthes [VM23] focused on approaches to scientific fact-checking specifically. They list 6 datasets for fact-checking and claim verification in the domain of science: SciFact, PubHealth, Climate-FEVER, HealthVer, COVID-Fact, CoVERT. They also provide the performance (in terms of the F1 score) of approaches they were able to find on each dataset, with three out of the six datasets only listing a baseline performance, two datasets listing the performance of one implemented approach each, and the SciFact dataset listing 5 different implementations and their scores. While there are some differences in the document/evidence retrieval step, all the approaches listed utilize a transformer-based architecture for claim verification: Longformer, T5, BERT or some fine-tuned BERT-based model.

There are also leaderboards for some fact-checking datasets on paperswithcode.com[1]. The most popular dataset for fact-checking is FEVER by Thorne et al [TVC$^+$18]. Over

[1] paperswithcode.com/sota/fact-verification-on-fever

the past few years, top scores have been improved by advances in the field. Graph neural networks like KGAT by Liu et al. [LXSL20], transformers, seq2seq models like ProofVER [KRV22] and improvements to the capabilities of transformers have advanced the top performance scores on this dataset. Currently, the best performing approach is called BEVERS, introduced in a paper by DeHaven and Scott in 2023 [DS23], which uses a claim verification component based on the DeBERTa model [HLGC21] (not listed on the paperswithcode.com leaderboard).

Overall, deep learning and neural methods are leading the state-of-the-art in claim verification, one of the crucial steps in fact-checking [ALA23]. Deep learning-based retrieval models can be combined in a re-ranking pipeline where a simpler and computationally cheaper algorithm like BM25 pre-filters the majority of the search space. The more powerful and resource-intensive models can then be used on this smaller search space, increasing retrieval efficiency in terms of time and resources. The state-of-the-art model we use for retrieval is NV-Retriever-v1 [MOX$^+$24], a new model at the time of writing. It is described further in Section 5.1.

3.5 Stance Detection

A task that is similar to claim verification (via authority evidence as context) is “stance detection”. The task of “stance detection of authorities towards rumors” was formulated by Haouari et al. in 2023 as:

> “Given a rumor expressed in a tweet and a tweet posted by an authority of that rumor, detect whether the tweet supports (agrees with) the rumor, denies (disagrees with) it, or not (other)”. [HE24](p.2)

The approach Haouari et al. took [HE24] involved fine-tuning a BERT model using a dataset they created themselves, comprised of Arabic Twitter posts. They also investigate approaches to alleviate class imbalance and published the “AuSTR” dataset along with the paper.

In essence, this is an earlier version of the task we are attempting. Importantly, in our thesis, we don’t aim to assign a truth value at all. Our approach instead attempts to predict support or refutal from authority sources. Verifying factual claims usually involves claim extraction and retrieval of facts from a knowledge base, which are then used to verify the factuality of a claim.

A study paper by Cruickshank et al. [CN24] already investigated the validity of using LLMs for stance detection as a general NLP task. They systematically tested multiple LLMs and prompting schemes using manually annotated social media sets that are traditionally used for other NLP tasks. Like us they highlight the benefits of zero-shot inference, but also investigate performance improvements via few-shot inference and fine-tuning. Additionally, they highlight related work specific to the task of stance detection, which can be found in Table 1 of their paper, where nearly all the work they survey use some sort of OpenAI GPT model.

In short, Cruickshank et al. find that the models they used don't consistently outperform the accuracy baselines provided along with the dataset at the time each dataset was published. In most of these baselines, the baseline accuracy is provided by a model relying on "traditional" machine learning methods. However, "specific prompting schemes" can improve the performance of LLM-based approaches to stance detection to the point where they outperform the previously mentioned "traditional" methods. Fine-tuning LLMs also did not necessarily increase performance. A problem the authors encountered throughout their study was the generation of "invalid" answers by the LLMs, which did not adhere to the answering scheme laid out in the prompt.

The 6 datasets the authors use for evaluation have different topics and contexts, where the meaning of "stance" depends on the dataset. They alleviate this issue of different labels by modifying the prompt given to the LLM based on the dataset. The authors use 10 different LLMs in their experiments, with the largest model (in terms of parameter count) being the "Falcon 40B Instruct" model. They also do some fine-tuning using LoRA [HSW+21], but find that fine-tuning actually tends to generally decrease performance.

As mentioned previously, the authors did not find any LLM that consistently outperformed established baselines "out of the box". They also had issues with "invalid" responses, where about half of the responses could not be parsed into a single, valid label. Filtering for only valid responses, and using few-shot prompting or chain-of-thought prompting, Cruickshank et al. "[. . .] observe that the accuracy scores for the good results consistently and significantly outperform the baseline supervised approaches" [CN24](p.14).

Comparing the LLMs Cruickshank et al. used to the LLMs that are available today and which we use in our thesis shows a drastic performance gap not only in terms of model size (parameter count), but also in terms of output consistency and stability. The industry and research around LLMs are progressing rapidly, which likely explains the performance gap we see between their results and the results we present in our thesis. Many of the experiments conducted in their study and issues highlighted in their paper are, however, still relevant today and will be for quite some time.

3.6 CLEF 2024 CheckThat! Lab Task 5

During the 2024 edition of the CheckThat! Lab at CLEF we participated in the shared Lab Task 5. The task is set up with these guidelines:

- "Definition: 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. This task is offered in both Arabic and English."[2]

- Datasets for development and evaluation are provided[3].
- Submission guidelines, baselines and evaluation scripts are provided[4].
- There are four leaderboards showing submission results from participants: Evidence Retrieval (Arabic), Evidence Retrieval (English), Verification (Arabic), Verification (English).

We only participated in the English part of the shared task, where we scored the highest on the leaderboard for Verification (English). Other participants implemented systems with a wide variety of components, some of which we will take a closer look at here.

For English retrieval, the top spot on the leaderboard was achieved by team "IAI Group" (consisting of the members Aarnes, Setty, Galuščáková from University of Stavanger, Norway) using a cross-encoder from HuggingFace. This team used the model "cross-encoder/ms-marco-MiniLM-L-12-v2" without fine-tuning on the train split of the dataset[5]. Their approach demonstrates that retrieval without fine-tuning can work adequately. Unfortunately, they did not publish a paper or working notes on their approach for the CheckThat! Lab Task 5.

Team Axolotl published their approaches as working notes [PF24] for the CLEF 2024 conference. They experimented with several setups, making use of both lexical and semantic retrieval methods in various combinations for retrieval. For verification, they experimented with both LLMs and more traditional transformer models. The details are available in their paper [PF24].

Table 7.7 in Section 7.4 contains the submission results and approaches used by other participating teams.

[2] From the official task website: checkthat.gitlab.io/clef2024/task5
[3] gitlab.com/checkthat_lab/clef2024-checkthat-lab/-/tree/main/task5/data
[4] gitlab.com/checkthat_lab/clef2024-checkthat-lab/-/tree/main/task5
[5] This was confirmed by one of the "IAI Group" members via email.

CHAPTER 4

Experiment Setup

To answer our research questions, we conducted and evaluated experiments. In these experiments, we test different configurations of components and features to find the most effective configuration for this specific task of evidence retrieval and evidence-based fact-checking. This section describes the framework for our experiments.

4.1 Performance Measures

To answer our research questions, we will consider these performance metrics...

- ...for RQ1 (retrieval only):
  - Recall@5
  - Mean Average Precision (MAP)
- ...for RQ2 (verification only) and RQ3 (verification using retrieved evidence):
  - Macro-F1 score
  - Strict-Macro-F1 score

These performance metrics are identical to the target performance metrics used by the CheckThat! Lab Task 5 organizers to score submissions for their leaderboards during the submission period. We also select these measure for this thesis. While CheckThat! Lab Task 5 used MAP as the primary measure and Recall@5 as the secondary measure, we are more interested in Recall for our approach as finding any relevant evidence is the objective of RQ1. Macro-F1 is also well-suited to our objectives, as our dataset is somewhat imbalanced, and Macro-F1 provides a class-balanced score indicating overall performance. Strict-Macro-F1 is a special measure, where a rumor label is only considered correct if at least one piece of relevant evidence was retrieved. We use Recall@5 and Macro-F1 as primary performance measures, with MAP and Strict-Macro-F1 serving as secondary performance measures.

4.2 Dataset

To conduct our experiments, we use a dataset provided by the CheckThat! Lab at CLEF 2024, intended for use in the CheckThat! Lab Task 5.

4.2.1 Dataset Structure

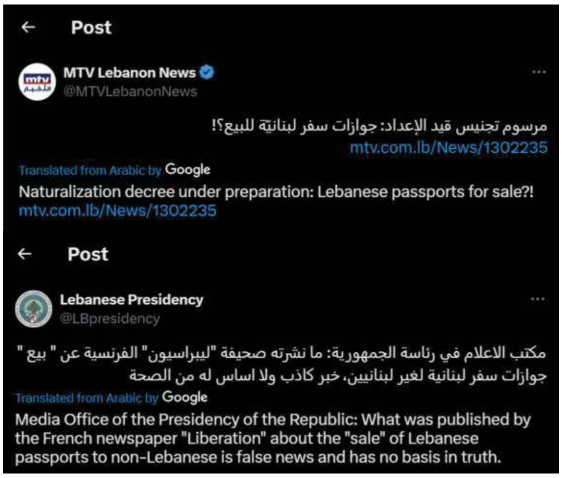
The dataset contains JSON-lines, each of which represents a rumor and a set of tweets from authority sources that could be relevant in regard to that rumor. Additionally, each rumor was "labelled" by the task organizers ("SUPPORTS" or "REFUTES" or "NOT ENOUGH INFO"), and – if the rumor is verifiable, meaning the label is "SUPPORTS" or "REFUTES" – also provides the relevant subset of tweets that were used by the task organizers to label the rumor.

The dataset consists of three "splits": "train", "dev" and "test". The split "test" was intended to be used as data for a "blind" run, and the predictions on the "test" set were submitted to the CheckThat! Lab Task 5 team. The results from the submissions of each team at the Lab Task were published on a leaderboard. Due to the nature of this task, no labels were obviously provided for the "test" data. Since we need to evaluate our systems for this thesis, we will not use the "test" data. The "train" and "dev" splits are identical in terms of structure, they simply contain different rumors. While we could fine-tune pre-trained models on the "train" or "dev" data, we believe that a zero-shot approach is the best fit for the out-of-domain performance we want, to cover a broad array of topics and posts. For this reason, we will simply combine the "train" and "dev" splits into a single dataset.

The rumors and timelines were originally posted to Twitter in Arabic. Additionally, the dataset was translated from Arabic to English by the task organizers using the Google Translate API. This English version of the dataset is what we use in this thesis. Due to the translation step, however, some rumors and tweets only consist of the text "ISSUE: COULDN'T TRANSLATE" – we drop those rumors, and do not use them in this thesis. Overall, we dropped 13 rumors (11 from "train", 2 from "dev").

After dropping rumors with translation issues, we are left with 30 (32-2) rumors in the "dev" split, and 85 (96-11) rumors in the "train" split, which we combine into a dataset with 115 (30+85) rumors to be used for the purposes of evaluating our approach. We will refer to this combined data as "our dataset" from here on out.

We define an authority or "authorities" in the context of this thesis as follows, after Haouari et al. [HEM23]:



MTV Lebanon News
@MTVLebanonNews

مرسوم تجنيس قيد الإعداد: جوازات سفر لبنانيّة للبيع؟!
mtv.com.lb/News/1302235

Naturalization decree under preparation: Lebanese passports for sale?!
mtv.com.lb/News/1302235

Lebanese Presidency
@LBpresidency

مكتب الاعلام في رئاسة الجمهورية: ما نشرته صحيفة "ليبراسيون" الفرنسية عن " بيع "
جوازات سفر لبنانية لغير لبنانيين، خبر كاذب ولا اساس له من الصحة

Media Office of the Presidency of the Republic: What was published by the French newspaper "Liberation" about the "sale" of Lebanese passports to non-Lebanese is false news and has no basis in truth.

Figure 4.1: The claim tweet and one of the relevant evidence tweets included in rumor AuRED_142 (`https://x.com/LBpresidency/status/1555424541509386240` and `https://x.com/MTVLebanonNews/status/1555393952378937346`)

> “Authorities” are not necessarily government entities or government bodies but are defined as an entity with authoritative knowledge over the domain of the claim. Authorities are considered experts in their field, but not all experts are considered authorities, as described by Haouari et al. [HEM23]. In cases where a claim is about a specific entity, that entity is considered an authority, since the claim is about the entity itself.

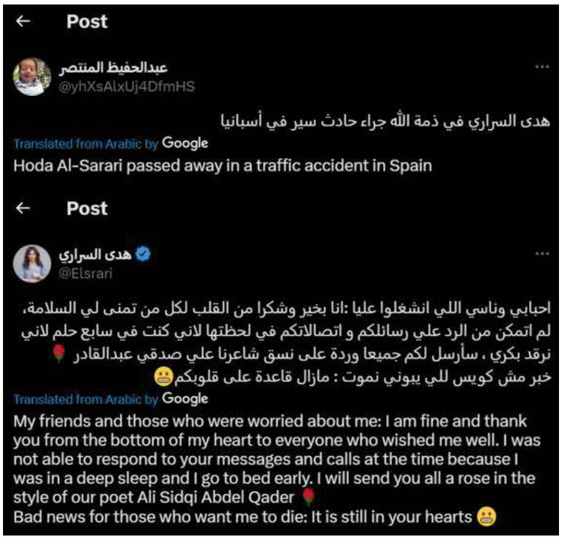
Figure 4.1 shows screenshots of (part of) the tweets that make up one rumor in our dataset. In this case, the authority is the Lebanese Presidency, on the topic of ongoings in Lebanon. Figure 4.2 shows some of the tweets that make up another rumor in our dataset, this time the claim stating that a person has “passed away in a traffic accident”, and the authority being the person the claim was about. These examples illustrate how what an authority source is can depend on the claim being made.

As stated previously, for this thesis we exclude the tasks of claim detection, claim matching and authority finding. These tasks can be fulfilled by other systems in the field, and we work with the data in the CheckThat! Lab Task 5 dataset described above, provided and labeled by the CheckThat! Lab organizers.

4.2.2 Addressing Training Data Inclusion of our Dataset

The dataset we are using was released to the public in the beginning of 2024, and we do not believe that the labeled data leaked into the training data of the models we are using. The tweets were all originally posted in Arabic and translated to English.

For “open models” like the Llama family of models the corpus of training data is known, and the weights are fixed and available online. The models are not continually updated,



← Post

عبدالحفيظ المنتصر
@yhXsAlxUj4DfmHS

هدى السراري في ذمة الله جراء حادث سير في أسبانيا

Translated from Arabic by Google

Hoda Al-Sarari passed away in a traffic accident in Spain

← Post

هدى السراري
@Elsrari

احبابي وناسي اللي انشغلوا عليا :انا بخير وشكرا من القلب لكل من تمنى لي السلامة،
لم اتمكن من الرد علي رسائلكم و اتصالاتكم في لحظتها لاني كنت في سابع حلم لاني
نرقد بكري ، سأرسل لكم جميعا وردة على نسق شاعرنا علي صدقي عبدالقادر 🌹
خبر مش كويس للي يبوني نموت : مازال قاعدة على قلوبكم😬

Translated from Arabic by Google

My friends and those who were worried about me: I am fine and thank you from the bottom of my heart to everyone who wished me well. I was not able to respond to your messages and calls at the time because I was in a deep sleep and I go to bed early. I will send you all a rose in the style of our poet Ali Sidqi Abdel Qader 🌹
Bad news for those who want me to die: It is still in your hearts 😬

Figure 4.2: The claim tweet and one of the relevant evidence tweets included in rumor AuRED_160 (https://x.com/yhXsAlxUj4DfmHS/status/1436813633607159810 and https://x.com/Elsrari/status/1436952831215476739)

and the "knowledge cutoff" signifying the last date of data collection is earlier than the publication of the data we use.

Proprietary models like the GPT family of models very likely include data from social media, and are likely to be updated continually. However, the dataset we use is not available in an easy-to-scrape format. We also conducted simple experiments to check adherence to the prompt and test for inclusion of our dataset, and are optimistic that our dataset was not included at the time of writing. For these tests, see Section 7.6.

4.3 Experiment Framework

To answer our research questions, we create a framework with implementations of the components visualized in Figure 4.3. Figure 4.3 consists of multiple numbered blocks, which map onto how data flows through our framework. We step through the numbered blocks here, referencing which Sections describe them in more detail:

- Block 1: the dataset and its structure are described in the the above paragraphs, here in Section 7 .
- Blocks 2 and 3: our preprocessing strategies, as well as the approaches we use for evidence retrieval are expanded on in Section 5.
- Block 4: we describe the models we compare for the task of claim verification in Sections 6.1 and 6.2.

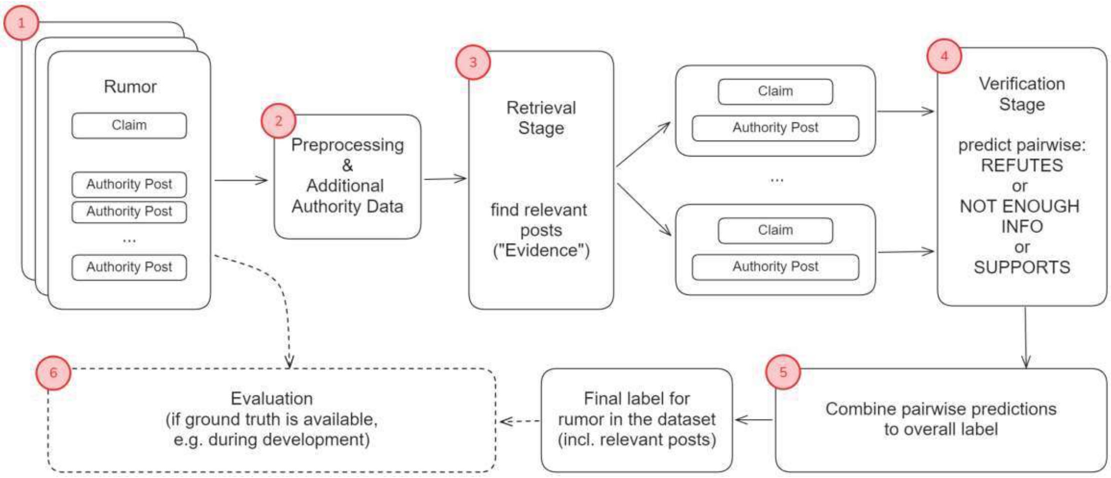


Figure 4.3: Diagram showing the framework we use for our experiments.

- Block 5: further details on how the final, overall rumor label is created can be found in Sections 6.3 and 6.4.

- Block 6: the results of the experiments we conduct using the implemented components in our framework and evaluate using the ground-truth labels are provided and discussed in Section 7. In Section 7, we also answer our research questions.

Along with the answers to our research questions, we conduct additional experiments to evaluate the validity of our approach. These are the experiments we conducted:

- Finding the approach that produces the best achievable performance score in each research question setting: RQ1 (best possible retrieval stage), RQ2 (best possible verification stage assuming perfect evidence) and RQ3 (best possible verification using our own retrieved evidence).

- Comparing the three Llama-3.1 models against each other to evaluate the impact of model size (in terms of parameter count) on the verification stage performance.

- Impact of the parameters “temperature” and “top_p” in LLM approaches to claim verification.

- Consistency between different runs using the same GPT-4o-mini model: do the predictions arbitrarily change across multiple executions using the same model and input data? This experiment aims to highlight consistency and examines if the predictions are simply “hallucinated”.

- Testing for logical consistency in GPT models by both inverting the label and editing the rumor text such that the flipped label would be accurate. This experiment aims to find our if the predictions the GPT models produce are simply memorized, or if they are actually generated depending on the data we provide via the prompt.

To implement the framework described above, we use Python and set up a repository[1] containing the source code as well as the experiment results. For the larger models like the GPT or Llama-3.1 models, we use external APIs from third parties to obtain our predictions. For the GPT models, we use the OpenAI Assistant API. For the Llama-3.1 models, we use our own deployed endpoints on Microsoft Azure. To obtain the NV-Retriever-v1 embeddings, we use the NVIDIA NIM API.

[1] github.com/LuisKolb/thesis

CHAPTER 5

Retrieving Authority Statements as Evidence

To verify claims using authority evidence, we first need to find evidence which is relevant to a given claim. Our dataset provides a set of timelines for each claim, from around the surrounding time period each claim was published to Twitter. We need to retrieve a number of tweets from these timelines to use as the basis of our verification stage.

Retrieving a limited number of rumors before verification serves two purposes:

- It reduces the computational resources required for the verification process.
- More importantly, it limits the noise that would be introduced by verifying each pairing of claim and evidence. If no retrieval were to take place before verification, we would have to score each pairing of claim and evidence individually. For every non-relevant tweet, ideally, we would predict "NOT ENOUGH INFO" such that the non-relevant tweets would not be considered when predicting an overall label for the rumor. However, this "score everything" approach is more likely to introduce noise and more prone to mispredictions, as a much higher number of individual predictions are produced. Prefiltering via the retrieval stage ensures that only a few pieces of likely relevant evidence are considered for the final prediction, preempting the noise caused by mostly non-relevant "evidence".

5.1 Evidence Retrieval Methods

There are several methods for efficient document retrieval from a corpus of documents. For our purposes, the documents are tweets by authority accounts. The corpus we are retrieving from is unique for each claim. This is contrary to the approach of retrieving evidence from a single, large corpus of documents (or a “knowledge base”) and using those as evidence to fact-check claims.

We tested two general approaches for evidence retrieval, with different implementations of each general approach. We also implement a basic version of a cross-encoder approach (the general idea is also described by Rosa et al. [RBJ+22]). We compare the performance of these general approaches in our thesis: lexical retrieval methods, cross-encoders for retrieval and embeddings for retrieval.

The lexical retrieval methods (as described in Section 2.2.1) we implemented are:

- BM25 with default parameters
- TF-IDF

For the cross-encoder approach, we use the Sentence Transformers (“SBERT”) library to locally run a very basic re-ranking pipeline consisting of a Bi-Encoder and a Cross-Encoder. One of the teams at the CLEF 2024 CheckThat! Lab Task 5 used a cross-encoder approach for retrieval, achieving the high score on the English retrieval leaderboard. We also implement a basic cross-encoder approach, but do not fine-tune it (as stated, we aim to investigate the performance of our approaches in a zero-shot setting). We expect this approach not to perform well, and not to achieve the high score of the other team.

Like we described in Section 2.2.2, we can use embedding models to compare pairings of text (in our case, claim and authority posting), and find the pairings which score the highest according to our similarity measure. The authority postings from the top pairings are returned as the “retrieved evidence” for the given claim. To compare a sampling of multiple embedding models, our experiments cover three different embedding models:

- “multi-qa-distilbert-cos-v1”[1] (768 dimensions).
- “NV-Retriever-v1” [MOX+24] by NVIDIA, based on Mistral-7b (4069 embedding dimensions and 32k input tokens, which is more than sufficient for our data), implemented via the Nvidia NIM API[2]. This model is also available via HuggingFace[3].
- One of the OpenAI embedding models: “text-embedding-3-small” (proprietary, 1536 dimensions).

Since some models like NV-Retriever-v1 are expensive to run for a large amount of input data, we utilize a re-ranking pipeline, where documents are first ranked using a simple BM25 algorithm, and the NV-Retriever-v1 model only runs on a subset of the original

[1] huggingface.co/sentence-transformers/multi-qa-distilbert-cos-v1
[2] build.nvidia.com/explore/discover
[3] huggingface.co/nvidia/NV-Retriever-v1

input data. We found that this combination of a simple lexical retrieval algorithm and a powerful semantic embedding model works well.

NV-Retriever-v1 is also actually based on a Large Language Model (LLM). Mistral-7b-v0.1 is used as the foundational model that NV-Retriever-v1 is built on. More information about the techniques used to create this model can be found in the paper by its creators, Moreira et al. [MOX+24]. NV-Retriever-v1 (and NV-Embed models, introduced in a previous paper by Lee et al. [LRX+24]) uses training data comprised of different publicly available datasets covering various retrieval tasks. These datasets are published in their respective papers, and while they do contain some datasets from other CLEF Labs (like BioASQ), we are happy to report that our CheckThat! dataset is not included. The datasets used to fine-tune the NV-Retriever-v1 model we use in this thesis are listed in Appendix C of the paper by Moreira et al. [MOX+24].

5.2 Preprocessing

Preprocessing can improve lexical matchability, as mentioned in Section 2.2.1. There are many traditional methods in NLP, like stemming or lemmatization. However, these methods inherently remove some information (or “signal”) – even normalizing capitalization (usually to lower-case) removes information that could potentially be useful later on in the verifier stage, so preprocessing should most likely be restricted to the retrieval stage.

Additionally, tweets sometimes include additional “symbols” or content, such as:

- Hashtags, which might contain valuable information required for an accurate verification prediction. We do not strip hashtags away completely, but simply remove the “#” character.
- Username mentions, which are prefixed with an “@” character. Similar to hashtags, we strip away the “@” character, but leave the username, as it might be necessary to “understand” the contents of the tweet.
- The pattern “RT @<username>:”, which is part of the text provided by the Twitter API. This pattern indicates that the tweet is a “Quote Tweet”, a platform-specific signal which we do not use for this thesis. Accordingly, we strip out this pattern.
- Links, which are part of the tweet text provided by the Twitter API. These links can refer to other tweets in case of a “Quote Tweet”, media like photos or videos, and so on. Multimodal input is excluded from our thesis, though it could be integrated into the verification stage, if the verification is powered by a LLM with multimodal capabilities, which are becoming more common. For our thesis, we strip out the links.

We also remove special characters and emojis, which might contain some information, but generally introduce more noise. Especially emojis are sometimes semantically ambiguous.

5.3 Introducing Additional Data

A tweet presented to a human reader contains more information than just the tweet text: the tweet date and the author's name are visible, as well as attached media such as pictures, images or video. If desired, the author's bio can easily be viewed on their profile. For this thesis, we experimented with adding additional data which is not present in the original data set. In our implemented setup, the author's "@username" is always added to the tweet text. If an optional feature is enabled, the author's display name and the author's bio will be also added before the text of the statement. In some cases, this additional information could improve retrieval, as well as the verification later. Most authority statements do not state the affiliation of the authority, but such information could be required to find the relevant statements from multiple tweets in multiple timelines from different authorities.

In other cases, adding the author's name or other information (like the Twitter bio) is actually crucial for the relevant evidence to be able to be found. For example: there is a claim that a person died in a traffic accident. The timeline we are retrieving from contains the statement "Thank you to all my friends, I am fine [. . .]" by the same person the claim is about. If the claim mentions the person by name, but the authority post does not contain the authority's own name, the retrieval stage could miss that authority post. The issue here is the intersection of similar language, words or tokens between claim and authority post. With the addition of the author's name or Twitter bio, this evidence could now be found more easily by a retrieval stage. This example actually exists in our dataset, as can be seen in the original tweets shown in Figure 4.2 in Section 4.2.1.

In this section, we described why a retrieval stage is necessary, the components we implemented to represent different approaches to retrieval from a timeline of authority posts, and how the input data could be processed to improve retrieval performance. The performance of each component in combination with different data preprocessing steps is evaluated in Section 7.1.

CHAPTER 6

Using Authority Evidence for Verification

Posts on social media are often formulated using informal, casual language. This presents a challenge to traditional NLP methods. Today, models with a transformer architecture ("transformers") are state-of-the-art when it comes to natural language understanding, especially Large Language Models (LLMs). So, for the verification stage, we are exclusively looking at transformer-based approaches to fact verification.

As mentioned in Section 2, we will focus on "zero-shot" approaches, where we don't train or fine-tune the models we use on our own data. We want our approach to be generally applicable across multiple topics and platforms, and do not want to limit our approach to only a select few, fine-tuned topics posted in the format of a tweet.

The following two Sections 6.1 and 6.2 describe two different approaches to claim verification: using BERT-based models, and using LLMs for claim verification. In our framework implementation, we set up the models in way that these models provide their prediction to a set specification. Set up like this, the claim verification stage can use different models to generate verification "predictions". We use this claim verification stage with different models to compare their performance on the task of claim verification in Sections 7.2 and 7.3.

6.1 BERT-Based Verification

A transformer-based approach we implement uses RoBERTa [LOG^{+}19] as a foundational model, which was introduced in 2019. One of the tasks for which fine-tuned versions of RoBERTa are available is Natural Language Inference (see Section 2.3). The "roberta-large-mnli" model is only one of the models fine-tuned on the MultiNLI dataset, as there are multiple models available which were already fine-tuned on this dataset. Another fine-tuned model we implement is based on BART [LLG$^+$19], a pre-trained seq2seq model also introduced in 2019. As these models are rather similar across the board in terms of performance and approach (compared to other similar transformer models fine-tuned on NLI datasets), we decided to include only these two models in our experiments: “FacebookAI/roberta-large-mnli”[1] and “facebook/bart-large-mnli”[2]. The fine-tuned models are available on and were obtained from HuggingFace.

6.2 Large Language Models for Verification

Like we described in Section 2.4, LLMs are trained to perform a wide range of NLP tasks. In our thesis we compare the performance of different LLMs as verification models, similar to the task of Natural Language Inference (NLI, see Section 2.3).

We can create an input in such a way that the model will respond predictably (also called “prompting”). For our purposes, we take the pairing of claim and authority posting and insert them into the prompt to the LLM. A special system prompt instructs the model to only consider data supplied in the prompt to formulate the answer, and to respond in a specified JSON-format template of {“label”: ..., “confidence”: ...}.

Generally, the prompt we supply asks the model to perform the task of NLI, where the answer should provide a label and score indicating how much the authority posting entails or contradicts the claim. We also give the LLM the option to respond with an answer of “NOT ENOUGH INFO”, if the pieces of text are not related. Since our approach always scores a fixed number of claim-evidence pairings, it is very likely that some pairings are not related to each other.

The system prompt will be consistent across all LLM-based approaches, and it is as follows:

> 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 given 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.

[1] huggingface.co/FacebookAI/roberta-large-mnli

[2] huggingface.co/facebook/bart-large-mnli

There is a question that remains, though: does an LLM have the capabilities to reason? A very recent paper by Mirzadeh et al. [MAS$^+$24] states "[...] that current LLMs are not capable of genuine logical reasoning; instead, they attempt to replicate the reasoning steps observed in their training data ." [MAS$^+$24](p.1). Their paper studied the mathematical reasoning capabilities of current LLMs. For our task, however, we investigate the capabilities of LLMs with respect to the natural language inference steps required to produce the correct label.

Hallucinations, which refer to LLM output that contains untrue statements, were not an issue for our use case. We don't ask the LLM to generate facts, and therefore hallucinations don't meaningfully influence the trustworthiness of the output, as we demonstrate via our experiments in Section 7. To verify that our verification step yields consistent predictions, and not simply random answers, we also compared the predictions of the sample model executed multiple times. We compare the label and score produced by these different "runs", and the variation between the runs with respect to the performance measures.

Most LLMs also support setting parameters that control the behavior of the LLM, like "temperature" and "top_p". We also tested the impact of these settings on consistency between different executions of the same model.

Not all LLMs are equal, however. Each model is different depending on the training data, input count, special instructions and "safeguards" by the model provider. The LLama3 model of families is available with three different parameter counts: 8 billion, 70 billion and 405 billion. We tested these models against each other, and against two other popular OpenAI models: GPT-4o and GPT-4o-mini, the most capable and efficient OpenAI models available at the time of writing. Unfortunately, the OpenAI model weights (and exact parameter counts) are not publicly available, while the Llama3 weights are publicly available.

Nearly all top "inference-as-a-service" models like models from OpenAI or Anthropic are subject to moderation restrictions, which could impact the LLMs' willingness to provide a SUPPORTS or REFUTES label for political or otherwise sensitive content. However, some models that are "open-weight" – such as the family of Llama models – can be hosted on your own infrastructure. These models have their own restrictions via a usage policy that has to be agreed to in order to download and legally use the models.

6.3 Configuration Options for the Verification Stage

In the verification stage, we provide three multiple configuration options that influence the verification stage across all implementations, no matter the underlying model used for verification:

- "Scale": can be true or false. Controls whether the confidence score produced by the verification model for a given pairing of claim and evidence is scaled (multiplied) by the score returned by the retrieval stage. In theory, since we retrieve a set number of evidence pieces per rumor (top-k evidence pieces, where k=5), scaling up the importance of predictions that were based on a highly relevant (according to the retrieval score) piece of evidence should give relevant predictions more impact, and vice-versa for non-related pieces of evidence. For claim-evidence pairings with a prediction of “NOT ENOUGH INFO”, this feature has no impact on those predictions (since a “NOT ENOUGH INFO” prediction is always scored 0).

- “Normalize”: can be true or false. Controls whether the score returned by the retrieval stage is normalized to [0, 1]. The maximum and minimum scores that all the scores get normalized to are global across the dataset being used. In our implementation, the retrieval stage produces a TREC-formatted file containing the raw, un-normalized retrieval scores, and – if the feature is enabled – retrieval scores are then normalized. Normalizing the retrieval scores after retrieval already happened of course has no impact on the retrieval stage, and is only optionally used in conjunction with the “Scale” feature described previously.
- “Ignore NEI” (“Ign.NEI”): can be true or false. Controls whether “NOT ENOUGH INFO” (NEI) predictions are included in the total average confidence calculation. As described in more detail in Section 6.4, the overall label is produced from the averaged confidences, if the total average passes a threshold. NEI predictions will always be zero, thus including NEI predictions will lower the total absolute score (Ign.NEI=false), while not including NEI predictions in the total average will make it easier for the threshold to be passed in either direction - positive or negative (Ign.NEI=true). An example: if the system retrieves 5 rumors, and creates predictions for each of the 5 pairings, where 1 pairing has a score of +0.7 (REFUTES with a confidence of 0.7) but 4 other, irrelevant pairings score 0 each, the calculated overall score of $(0.7+0+0+0+0)/5 = +0.14$ would not pass the threshold of 0.15. The overall label would be NEI, instead of REFUTES. If NEI predictions are ignored, the overall score would simply be +0.7, giving an overall label of REFUTES.

In our paper documenting our approach and results for CLEF 2024 CheckThat! Task 5 [KH24], we evaluated these three features over all the combinations of features. In the paper, Table 2 detailed the score difference over all combinations of features. We again provide the results for completeness in Table 6.1. Since we evaluated the impact in terms of score for each feature, we used our previous research to decide on the configuration of these three features that we are using in this thesis:

- Ignoring NOT ENOUGH INFO predictions in the overall label prediction calculation (“Ign.NEI” = True).
- Not scaling confidence by the score returned from the retrieval stage (“Scale” = False).

Feature / Component 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
ExternalData	False	True	+0.0066	+0.0078
Normalize	False	True	+0.0300	+0.0306
Scale	False	True	+0.0635	+0.0637
IgnoreNEI	True	False	+0.0709	+0.0708

Table 6.1: 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. This table is included directly from our earlier paper [KH24].

- Normalizing retrieval score to [0...1] ("Normalize"). This feature being set to False resulted in a small increase in Macro-F1-score in our paper [KH24], but in this thesis, since we are not scaling confidence scores by retrieval score ("Scale" = False), this feature is not relevant and not used.

Aside from the three features described above, which we set to a fixed value for this thesis, we provide two more features in our implemented system:

- "Preprocessing" the data similarly to what is described in Section 5.2.
- Adding external data from the authority's Twitter account (referred to as "ExternalData" in Table 6.1). Similar to Section 5.3, additional information about the authority's Twitter display name and Twitter bio can also be added to the dataset before verification. Including this additional data along with the authority post passed as evidence could be necessary to obtain a correct prediction in circumstances similar to those described in Section 5.3.

It is important to keep in mind that the values for the components "Verification" and "Retrieval", as well as for the features "Preprocessing" and "ExternalData" will vary depending on the experiment we report the results for in Section 7. In this thesis we have implemented the retrieval and verification stages using models like the Llama-3.1 family of models for verification, or the re-ranking BM25 and NV-Retriever-v1 pipeline for retrieval, which are different to the implementations from our earlier paper. The components we implemented for these two stages are described in more detail in Section 5 and Section 6, respectively. These two rows in Table 6.1 are only included for completeness. Whether text preprocessing or addition of authority account data is done is either fixed per experiment, or the results of different combinations are reported.

6.4 Aggregating Pairwise Predictions

Since for each rumor we retrieve (up to) 5 pieces of evidence, which are then individually passed to the verification stage as claim-evidence pairings, we need to somehow combine (up to) 5 individual claim-evidence predictions into an overall label per rumor.

To combine multiple claim-evidence pairwise predictions from the verification model, and produce an overall label for a rumor, we use a simple arithmetic formula. Each prediction from the verification model contains a label and a "confidence score" (based on the claim-evidence pairing). The way confidence scores are created depends on the verification model: the RoBERTa-based MNLI approach produces a score for each NLI label and outputs the label with the highest score as its prediction, which is also the confidence score in that label. The LLM-based approach is instructed via the system prompt to both produce a label and assign a confidence value to its decision between [0, 1].

Parsing the response from any verification stage model, we combine the predicted label and confidence score to map the response to a value between [-1, 1]. This prediction can be understood to semantically mean:

- Score between (0, +1] indicates a REFUTES prediction, with a value close to +1 indicating strong confidence
- Score of exactly 0 indicates a NOT ENOUGH INFO prediction, with no preference towards either SUPPORTS or REFUTES
- Score between [-1, 0) indicates a SUPPORTS prediction, with a value close to -1 indicating strong confidence

In our thesis (and the implementation of the approach we present in it), a positive value means the piece of evidence "REFUTES" the claim, a negative confidence score means the piece of evidence "SUPPORTS" the claim, and a confidence score of zero indicates a model response of "NOT ENOUGH INFO".

In short, calculating the mean of these individual scores produces a value between [-1, 1], which we can use to return our final overall label with the same semantics from the above paragraph (negative = "REFUTES", positive = "SUPPORTS"). If the mean of the scores is zero or close to zero, we are likely to have:

- Received exclusively "NOT ENOUGH INFO" predictions – in which case we want to return NEI as the overall label
- Received conflicting predictions of similar confidence, for example 1 "REFUTES", 1 "SUPPORTS", 3 "NOT ENOUGH INFO" - in which case we would want the system to return NEI as the overall label. Semantically this means the verification stage received two claim-evidence pairings, in one of which the evidence supports the claim

and in the other the evidence refutes the claim. The other three claim-evidence pairings were predicted to be unrelated.

- If, taking the above example, the “REFUTES” model prediction is much more confident than the “SUPPORTS” prediction, we may want to return “REFUTES” as the overall label, or vice-versa.

For the reasons listed above, we implement a threshold of a low value that the absolute value of the mean of the scores needs to exceed to predict a “REFUTES” or “SUPPORTS” label. The threshold is not learned and is set to 0.15. The threshold is set such that combining values indicating conflicting label predictions usually results in an overall label of “NOT ENOUGH INFO”. In the case of combining conflicting predictions, only if the predictions in one direction are much stronger or more numerous will an overall label other than “NOT ENOUGH INFO” be produced.

There are additional, optional features influencing overall label creation, which are listed in the previous Section 6.3.

This is the formula for overall label creation if the “Scale” and “Normalize” features are turned off:

$$\text{Overall Score} = \frac{\sum_{p \in \text{predictions}}(\text{prediction value}_p)}{\#\text{ of predictions}}$$

Here, the prediction value is a value between [-1,+1] as described above. As previously described, if the value of the Overall Score is:

- greater than +0.15: the overall label is “REFUTES”
- between -0.15 and +0.15: the overall label is “NOT ENOUGH INFO”
- less than -0.15: the overall label is “SUPPORTS”

The formula above is the one being used in the thesis. As described in Section 6.3, we do not include the retrieval score in our overall label calculation. If the retrieval score were to be included, this is the formula for overall label creation if the “Scale” and “Norm” features are turned on:

$$\text{Overall Score} = \frac{\sum_{p \in \text{predictions}}(\text{prediction value}_p) \times \text{Retrieval Score}_p}{\#\text{ of predictions}}$$

Retrieval score being the score given to the piece of evidence by the retrieval stage, normalized to [0,1]:

Model Name	**Architecture**	**Configuration Options (each With/Without)**
Llama-3.1-405B-Instruct	LLM	Preprocess Text; Add Author Data
Llama-3.1-70B-Instruct	LLM	Preprocess Text; Add Author Data
Llama-3.1-8B-Instruct	LLM	Preprocess Text; Add Author Data
GPT-4o	LLM	Preprocess Text; Add Author Data
GPT-4o-mini	LLM	Preprocess Text; Add Author Data
FacebookAI/roberta-large-mnli	Transformer (RoBERTa)	Preprocess Text; Add Author Data
facebook/bart-large-mnli	Transformer (BART)	Preprocess Text; Add Author Data

Table 6.2: Overview of models implemented for the verification stage. Each model can be used with and without preprocessing the input text, and with or without adding additional author data (Twitter display name and bio) to the input text. We present the results of the comparisons between these models in Section 7.

$$\text{Retrieval Score}_p = \frac{(\text{retrieval score}_p - \text{mininum retrieval score})}{(\text{maximum retrieval score} - \text{mininum retrieval score})}$$

The minimum and maximum retrieval score values are global across all retrieved evidence.

Above, we describe how our system handles conflicting evidence for a rumor. However, our dataset has a special characteristic we need to point out - it does not usually contain evidence that directly contradicts other evidence. Usually, if a rumor is labeled as “SUPPORTS” or “REFUTES”, the tweets from the timeline that are marked as “relevant evidence” (according to the annotators) each agree with the ground-truth overall label. While not as relevant for our specific dataset, it is important to keep the possibility of genuinely conflicting evidence in mind, as other datasets (or real-world use cases) might differ from our dataset in this aspect.

In this section, we presented our approach to claim verification. Table 6.2 gives an overview over the models we use for our experiments in the next Section, as well as the configuration options with which we test each model.

CHAPTER 7

Evaluation and Results

To answer the research questions formulated in Section 1.2 we conducted experiments using a system implementing the approaches from Sections 5 and 6. We also conducted a fail-case analysis to investigate drawbacks and potentially negative implications for our approach, to investigate under which circumstances issues arise.

In our previous conference working notes [KH24] we reported the findings of our experiments during our participation in the shared task. Since then, we have improved our implementations and added several new approaches and their implementations:

- **Multiple new models:** the Llama-3.1 family of models, for our CLEF 2024 CheckThat! Lab paper we used the Llama-3-8b and Llama-3-70b models in our submission. In this thesis, we report the results for the whole Llama-3.1 family of models (8b, 70b and 405b). For the OpenAI models we present results GPT-4o and GPT-4o-mini, instead of GPT-4-Turbo.
- For our CLEF 2024 CheckThat! Lab submission we only ran experiments on the dev split of the dataset and uploaded our results for the test dataset, for which the labels are not publicly available. In this thesis, we use the combined train and dev dataset splits instead of dev (and test for the competition) that we describe in more detail in Section 4.2.
- We include new retrieval models, namely NV-Retriever-v1 and a CrossEncoder, and implement two re-ranking approaches.
- We report our results running experiments to check consistency of repeated generation with the same model and data and report the impact of different settings for model parameters “temperature” and “top_p”. Additionally, we present a direct comparison between the three model sizes of the Llama-3.1 family of models with regards to claim verification performance.

- We also investigate the likelihood of our dataset being included in the GPT-4o training data.

7.1 Experiment Results for the Retrieval Stage

For the retrieval stage, we report the results of the 5 retrieval approaches we implemented:

- BM25 (with default parameters)
- TF-IDF
- Re-ranking pipeline of BM25 > NV-Retriever-v1
- Cosine distance between embeddings obtained from the OpenAI model "text-embedding-3-small"
- Re-ranking pipeline of Sentence Transformers "sentence-transformers/msmarco-distilbert-cos-v5" into the Cross-Encoder "cross-encoder/ms-marco-MiniLM-L-6-v2"

For each model, we evaluate and report the performance in terms of Recall@5 and Mean Average Precision (MAP). Our preprocessing approach and our approach to adding additional data about the authority ("author") are described in more detail in Section 5.2. Using our dataset described in Section 4.2, we report the results for 4 "variations" of our dataset: with and without preprocessing, with and without additional author data (Twitter display name, and Twitter bio).

- Without preprocessing and without additional author data (nopre-nonam-nobio): the "raw" data from the dataset.
- With preprocessing and without additional author data (pre-nonam-nobio): the "raw" data from the dataset, but preprocessed.
- With preprocessing and with additional author data (pre-nam-bio): we would expect lexical retrieval to be best here.
- Without preprocessing but with additional author data (nopre-nam-bio): we would expect semantic retrieval to be best here, as this includes the most information.

The general, non-specialized retrieval approaches we present the results of in Table 7.1 here do not perform particularly well. However, the best Recall@5 score is obtained by the BM25-NV-Retriever-v1 re-ranking implementation (with default parameters), with a Recall@5 of about 0.71. It performs the best across all data set variations but adding author information worsens the score. Preprocessing also lowers the score, but not as much as adding author information. As described in Section 5.1, we can also be sure that

Rank	Recall@5	MAP	Retrieval Method	Dataset Variation
1	0.714	0.671	rerank-nv-embed-v1	nopre-nonam-nobio
2	0.707	0.680	rerank-nv-embed-v1	pre-nonam-nobio
3	0.679	0.650	rerank-nv-embed-v1	nopre-nam-bio
4	0.675	0.647	rerank-nv-embed-v1	pre-nam-bio
5	0.645	0.580	bm25	pre-nam-bio
6	0.637	0.565	bm25	nopre-nam-bio
7	0.634	0.567	bm25	pre-nonam-nobio
8	0.633	0.575	openai	pre-nam-bio
9	0.631	0.588	openai	nopre-nonam-nobio
10	0.628	0.568	openai	nopre-nam-bio
11	0.620	0.558	bm25	nopre-nonam-nobio
12	0.618	0.531	tfidf	nopre-nam-bio
13	0.618	0.571	openai	pre-nonam-nobio
14	0.611	0.491	tfidf	nopre-nonam-nobio
15	0.602	0.504	tfidf	pre-nonam-nobio
16	0.598	0.538	tfidf	pre-nam-bio
17	0.531	0.499	rerank-sbert-crossencoder	pre-nonam-nobio
18	0.527	0.496	rerank-sbert-crossencoder	nopre-nonam-nobio
19	0.516	0.494	rerank-sbert-crossencoder	nopre-nam-bio
20	0.516	0.492	rerank-sbert-crossencoder	pre-nam-bio

Table 7.1: Experiment results showing the difference (in terms of Recall@5 and MAP) between setups using various retrieval approaches on different dataset variations (with and without preprocessing, with and without additional authority information).

our dataset was not included in the fine-tuning training data of the retriever model, as the authors of NV-Retriever-v1 include a list of datasets used in their paper [MOX$^+$24].

The second-best retrieval results are obtained with pure BM25 (with default parameters) on the preprocessed dataset with all additional author information – display name and Twitter bio – added (hence “pre-nam-bio”). It is, along with TF-IDF, also extremely fast to compute due to the simplicity of the algorithm. Overall, adding the embedding model on top of BM25 improved retrieval.

To answer Research Question 1: The best approach we found was a re-ranking pipeline of BM25 and NV-Retriever-v1, which worked well with any configuration of the dataset. The highest score we obtained was a Recall@5 of 0.714475 using the default Okapi BM25 implementation, and the NV-Retriever-v1 embedding model.

7.2 Experiment Results for the Verification Stage

For each model, we evaluate and report the performance in terms of Macro-F1 and Strict-Macro-F1. Like in the previous Section 7.1, we report the results for 4 variations

Macro-F1	Strict-Macro-F1	Model	DS Settings
0.814	0.814	Llama-3.1-8b-Instruct	pre-nonam-nobio
0.943	0.943	Llama-3.1-70b-Instruct	pre-nonam-nobio
0.985	0.985	Llama-3.1-405b-Instruct	pre-nonam-nobio

Table 7.2: Experiment results showing the difference in performance of the three Llama-3.1 models.

of our dataset: with and without preprocessing, with and without additional author data (Twitter display name, and Twitter bio).

- Without preprocessing and without additional author data (nopre-nonam-nobio): the “raw” data from the dataset.
- With preprocessing and without additional author data (pre-nonam-nobio): the “raw” data from the dataset, but preprocessed.
- With preprocessing and with additional author data (pre-nam-bio).
- Without preprocessing but with additional author data (nopre-nam-bio).

Assuming the setting of all relevant evidence being provided (the setting of RQ2), we can report the results of our experiment concerning model size in Table 7.2. The Llama-3.1 family of models comes in 3 sizes: 8b, 70b and 405b. The highest Macro-F1 was obtained by the most powerful model, Llama-3.1-405b-Instruct.

This experiment gives us some insight into the “weight class” of model that is required to tackle this task. There are some tricky rumors included in the dataset we use, and only the most powerful 405b model got them all right (with two exceptions).

In the cases where the 405b model does not predict the correct label, it predicts “NOT ENOUGH INFO” (the intended “failure mode”). The other models sometimes make the worst kind of mistake in this specific task: predicting overall “SUPPORTS” on a rumor labeled “REFUTES”, or the other way around.

Notably, the misclassified rumors largely overlap between models – the rumors misclassified by the 405b model were also misclassified by the two other, weaker models. We further analyze this in Section 7.5.

We conducted an experiment where the parameters “temperature” and “top_p” were set to different values and evaluated the output, looking for changes. The models we used here were the OpenAI model GPT-4o-mini, and the 70B variant of the Llama-3.1 family of models. The parameter values we tested and the Macro-F1 differences are listed in Table 7.3.

While we observed no meaningful impact in terms of Macro-F1-score across the GPT-4o-mini runs, we observed a clear drop in performance when both temperature and top_p

Macro-F1	temperature	top_p	Model Name
0.909	1	1	GPT-4o-mini
0.909	0.1	1	GPT-4o-mini
0.909	1	0.1	GPT-4o-mini
0.909	0.5	0.5	GPT-4o-mini
0.924	1	1	Llama-3.1-70B-Instruct
0.974	0.1	1	Llama-3.1-70B-Instruct
0.985	1	0.1	Llama-3.1-70B-Instruct
0.985	0.5	0.5	Llama-3.1-70B-Instruct

Table 7.3: Experiment results comparing verification models (in terms of Macro-F1) using different values for the parameters "temperature" and "top_p".

Predicted Labels Match	Evidence Scores Run 1	Evidence Scores Run 2	Evidence Scores Run 3	Rumor ID
Yes	[0.0, -0.95, 0.0, 0.0, -0.9]	[0.0, -0.95, 0.0, 0.0, -0.8]	[0.0, -0.95, 0.0, 0.0, -0.9]	AuRED_111
Yes	[+0.95]	[+0.95]	[+0.9]	AuRED_148
Yes	[-1.0, 0.0]	[-0.95, 0.0]	[-0.95, 0.0]	AuRED_125

Table 7.4: Experiment results from running the same OpenAI GPT-4o-mini model 3 times and calculating the differences between confidence returned in each of the runs. Each row represents a rumor where a difference in the predictions was detected. The values in the columns Evidence Scores Run 1-3 show the predicted values for all available claim-evidence pairings per run.

were set to lower values. "top_p" controls the proportion of tokens considered for output, while "temperature" controls creativity of the output. Theoretically, lower values should influence the model output to be more predictable and deterministic. Setting lower values for both resulted in improved performance in the Llama-3.1-70B model.

To test for hallucinations and randomness in the output of the OpenAI models, we tested consistency between runs. These runs used the same set of settings, prompt and data with the GPT-4o-mini model (temperature=0.01, top_p=0.5). As described in Section 6.4, a prediction of "SUPPORTS" and a confidence score of, for example, 0.95 maps to a value of -0.95. A prediction of "REFUTES" with a confidence score of, for example, 0.95 would map to +0.95. A "NOT ENOUGH INFO" predicted label maps to a value of 0.0. The values in the columns Evidence Scores Run 1-3 show the predicted scores for a piece of evidence, for example: a list of [-0.95, 0.0] means there were two claim-evidence pairings, and the model predicted the first pairing as "SUPPORTS" with a confidence of 0.95, and the second pairing as "NOT ENOUGH INFO".

We filtered the results for predictions that did not exactly match across our dataset and

Rank	Macro-F1	Strict-Macro-F1	Verifier Method	Model Identifier
1	0.985	0.985	llama3-1-405b	Meta-Llama-3.1-405B-Instruct
2	0.960	0.960	llama3-1-70b	Meta-Llama-3.1-70B-Instruct
3	0.937	0.937	openai-4o	GPT-4o
4	0.903	0.903	openai-4o-mini	GPT-4o-mini
5	0.803	0.803	llama3-1-8B	Meta-Llama-3.1-8B-Instruct
6	0.771	0.771	transformers-roberta	FacebookAI/roberta-large-mnli
7	0.705	0.705	transformers-bart	facebook/bart-large-mnli

Table 7.5: Experiment results for different verification models using the preprocessed dataset without additional authority account information, in the RQ2 setting using only relevant, hand-labeled evidence.

found three rumors where different confidence scores were returned for the same claim-evidence pairing. These three rumors and their respective values for each claim-evidence pairing are reported in Table 7.4. The maximum difference in scores returned by the OpenAI model using the same settings, prompt and data was 0.1. Even in the only cases where predictions don't exactly match, the difference is very small, at most between confidences of -0.9 and -0.8.

For our final experiment in RQ2, we compare various implementations of retriever models on the preprocessed dataset without adding additional author information. We assume perfect evidence retrieval for RQ2, so the evidence in the claim-evidence pairings sent to the different retrieval stages consists of the claim and evidence directly from the dataset, which was annotated as relevant evidence by a human (the CheckThat! Lab Task 5 authors, who provided the dataset).

The models in rank 6, 7 and 8 use the transformer models identified by the strings in the Model Identifier column, sourced from huggingface.

The LLMs (GPT-4o family, Llama-3.1 family) all take parameters, which were set to "temperature=0.2" and "top_p=1.0", respectively.

To answer Research Question 2: The best approach we found through our experiments was the Llama-3.1-405B-Instruct model, obtaining a Macro-F1-score of 0.985261. The 70B variant of this model family follows closely behind, with a Macro-F1-score of 0.960686.

7.3 Experiment Results for an Integrated Pipeline

For each model, we evaluate and report the performance in terms of Macro-F1 and Strict-Macro-F1. Like in the previous Section 7.1, we report the results for 4 variations of our dataset: with and without preprocessing, with and without additional author data (Twitter display name, and Twitter bio).

- Without preprocessing and without additional author data (nopre-nonam-nobio): the "raw" data from the dataset.
- With preprocessing and without additional author data (pre-nonam-nobio): the "raw" data from the dataset, but preprocessed.
- With preprocessing and with additional author data (pre-nam-bio).
- Without preprocessing but with additional author data (nopre-nam-bio).

The LLMs we compare, namely GPT-4o and GPT-4o mini, as well as the Llama-3.1 family of models, all take parameters, which were set to the default parameters temperature=0.2 and top_p=1.0, respectively. The other models were initialized with default parameters. All of the runs we present the results of in Table 7.6 use the best retrieval system we found in RQ1, the re-ranking pipeline consisting of BM25 and a re-ranking head of NV-Retriever-v1.

To answer Research Question 3: The best approach we found uses the GPT-4o model and the re-ranking retrieval pipeline of BM25 and NV-Retriever-v1 (which consistently performed the best in RQ1), achieving top scores across all four dataset variations, with a maximum Macro-F1-score of 0.858523 (though all the variations are very close in terms of score). Closely following is the less powerful GPT-4o-mini model, with a maximum Macro-F1-score of 0.810380. Generally, the less powerful models (in terms of model size) perform worse, receiving progressively lower scores.

Interestingly, the Llama-3.1-70B model sometimes outperforms the more powerful Llama-3.1-405B model. Among the Llama-3.1 family of models the dataset settings (preprocessing, additional authority information) seem to have more of an impact, broadening the score spread between the setups using the same model but different dataset settings.

The Llama-3.1 70B and 405B models also lose more performance in the RQ3 setting without perfect evidence. In the RQ2 setting, they outperformed the GPT-4o and GPT-4o-mini models, which don't lose as much Macro-F1-score compared to RQ2 Table 7.5.

7.4 CheckThat! Lab Submission Results

For completeness, we also present the results we achieved during our participation in the CheckThat! Lab Task 5 at CLEF 2024. We have already published our paper in the conference proceedings [KH24].

As part of the shared task, we were able to run our system on a previously unseen split of the dataset (called "test"), for which the labels were not available. Manually looking through the data, we think the "test" split of the dataset was harder to predict accurately than the "dev" split, which we used to evaluate our system in our CheckThat! working notes paper. We also think our combined "train" and "dev" dataset, which we use in this

Rank	Macro-F1	Strict-Macro-F1	Verifier Model	DS Settings
1	0.858	0.850	GPT-4o	pre-nam-bio
2	0.854	0.846	GPT-4o	pre-nonam-nobio
3	0.850	0.842	GPT-4o	nopre-nonam-nobio
4	0.844	0.840	GPT-4o	nopre-nam-bio
5	0.810	0.797	GPT-4o-mini	nopre-nam-bio
6	0.803	0.795	GPT-4o-mini	nopre-nonam-nobio
7	0.793	0.784	GPT-4o-mini	pre-nonam-nobio
8	0.781	0.768	GPT-4o-mini	pre-nam-bio
9	0.774	0.762	Meta-Llama-3.1-70B-Instruct	pre-nonam-nobio
10	0.767	0.759	Meta-Llama-3.1-70B-Instruct	nopre-nonam-nobio
11	0.766	0.751	Meta-Llama-3.1-405B-Instruct	pre-nonam-nobio
12	0.746	0.735	Meta-Llama-3.1-405B-Instruct	nopre-nonam-nobio
13	0.721	0.701	Meta-Llama-3.1-405B-Instruct	pre-nam-bio
14	0.709	0.691	Meta-Llama-3.1-405B-Instruct	nopre-nam-bio
15	0.708	0.696	Meta-Llama-3.1-70B-Instruct	nopre-nam-bio
16	0.705	0.692	Meta-Llama-3.1-70B-Instruct	pre-nam-bio
17	0.631	0.620	Meta-Llama-3.1-8B-Instruct	pre-nam-bio
18	0.609	0.597	Meta-Llama-3.1-8B-Instruct	pre-nonam-nobio
19	0.580	0.568	Meta-Llama-3.1-8B-Instruct	nopre-nonam-nobio
20	0.507	0.500	Meta-Llama-3.1-8B-Instruct	nopre-nam-bio
21	0.322	0.322	FacebookAI/roberta-large-mnli	nopre-nonam-nobio
22	0.312	0.303	FacebookAI/roberta-large-mnli	pre-nonam-nobio
23	0.298	0.280	facebook/bart-large-mnli	nopre-nam-bio
24	0.283	0.275	facebook/bart-large-mnli	nopre-nonam-nobio
25	0.270	0.270	facebook/bart-large-mnli	pre-nam-bio
26	0.264	0.264	FacebookAI/roberta-large-mnli	pre-nam-bio
27	0.225	0.216	FacebookAI/roberta-large-mnli	nopre-nam-bio
28	0.219	0.210	facebook/bart-large-mnli	pre-nonam-nobio

Table 7.6: Experiment results for combinations of different verification models and dataset settings using retrieved evidence. All runs used the retrieved evidence from the re-ranking pipeline consisting of BM25 and NV-Retriever-v1 from RQ1 (Section 7.1, “rerank-nv-embed-v1” in Table 7.1).

Team	Run Label	Macro-F1	Strict-Macro-F1	Retrieval Approach	Verification Approach
AuthEv-LKolb (ours)	secondary1	0.895	0.876	Embeddings + External Data	GPT-4
AuthEv-LKolb (ours)	primary	0.879	0.861	Embeddings	GPT-4
AuthEv-LKolb (ours)	secondary2	0.831	0.831	PL2	GPT-4
Axolotl	primary	0.687	0.687	Re-Ranker (BM25 & Llama3-8B)	Llama3-8B
	(baseline)	0.495	0.495		
Team DEFAULT	primary	0.482	0.454	COLBERT	not published
IAI Group	secondary1	0.459	0.444	Cross-Encoder	not published
bigIR	primary	0.458	0.428	KGAT	KGAT

Table 7.7: Selected results for the English verification leaderboard. For all runs, we used GPT-4 as the verification component. For each of the other teams, the best submission score is presented here. Retrieval and Verification Approach shows the method(s) the team used for the task, if the team disclosed their methods.

thesis, is more difficult than test, since our system generally got lower Macro-F1-scores when being evaluated on this combined dataset (compared to the “dev” split).

We ran 3 configurations of the implementation of our system at the time and uploaded our retrieved evidence and predicted labels. The authors of the shared task evaluated each groups’ results and published the scores for both retrieval and verification, each in English and Arabic. We only handed in results for the English part of the dataset. In Table n, we again present the results of our 3 submissions, the baseline, and the best results from each of the other 4 participating teams.

7.5 Fail-Case Analysis

In this section we highlight and analyze some cases where the system fails to predict the correct overall label. We will use the results produced by running the Llama-3.1 size comparison experiment for Research Question 2, the scores of which are presented in Table 7.2. We use the RQ2 setting (where we have “perfect” evidence), since we want to test the verification models, not the retrieval. We use these three models as they all have interesting fail cases. The dataset variation used preprocessing but did not add additional author information.

Importantly, we use only (up to) the first 5 pieces of evidence labeled as relevant by the dataset authors, even if there are more pieces of evidence labeled as relevant. If there are less than 5 pieces of evidence labeled as relevant, we use as many as are available. This is because we decided to limit ourselves to a top-k retrieval of 5 pieces of evidence for this thesis. It is possible that crucial evidence is missing from the relevant evidence, but this is only an issue if the other pieces of evidence are not sufficient to classify a rumor as SUPPORTS or REFUTES on their own. We would consider this circumstance a mislabeling of evidence as relevant.

If the system does not predict the correct label, there are three possible cases:

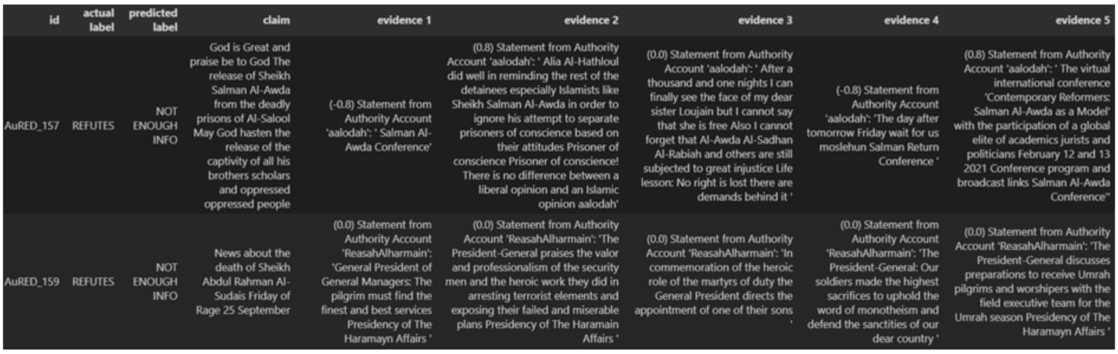
- *CASE 1:* the overall predicted label is SUPPORTS, but the actual label is REFUTES, or vice-versa (“producing the opposite label”).
- *CASE 2:* the actual label is REFUTES or SUPPORTS, but the overall predicted label is NOT ENOUGH INFO (“judging too cautiously”).
- *CASE 3:* the actual label is NOT ENOUGH INFO, but the overall predicted label is REFUTES or SUPPORTS (“judging too easily”). This case does not come up in the analysis presented here, as we have perfect evidence, and the system produces a NOT ENOUGH INFO overall label if there is no evidence. Rumors labeled as NOT ENOUGH INFO in the RQ2 setting have no evidence labeled as “relevant”, so would only be relevant in a RQ3 setting – not in a RQ2 setting.

In each Image 7.1, 7.2 and 7.4 the claim and evidence are presented like this:

- The claim and evidence text are presented in the image, with the score in front (in parentheses) of the evidence tweet text.
- The score indicates the label prediction and confidence score the model provided (for the respective claim-evidence pairing). Here are some examples:
  - (0.9) <tweet text> = positive confidence = REFUTES
  - (0.0) <tweet text> = neutral prediction = NOT ENOUGH INFO
  - (-0.9) <tweet text> = negative confidence = SUPPORTS

Llama-3.1-405B-Instruct: the model only makes two mispredictions across the entire dataset. Both mispredictions are of the CASE 2 type. Once because 2 pieces of evidence are judged as SUPPORTS, while 2 other pieces of evidence are judged as REFUTES, causing a tie.

The other misprediction is a result of not judging any piece of evidence as either SUPPORTS or REFUTES. Looking at the actual evidence tweet texts, none of them actually appear as relevant enough to classify the rumor as SUPPORTS or REFUTES given their text content. As we explained above, this seems to be a mislabeling of evidence as “relevant” by the dataset authors. It is possible that the original Arabic text



id	actual label	predicted label	claim	evidence 1	evidence 2	evidence 3	evidence 4	evidence 5
AuRED_157	REFUTES	NOT ENOUGH INFO	God is Great and praise be to God The release of Sheikh Salman Al-Awda from the deadly prisons of Al-Salool May God hasten the release of the captivity of all his brothers scholars and oppressed oppressed people	(-0.8) Statement from Authority Account 'aalodah': ' Salman Al-Awda Conference'	(0.8) Statement from Authority Account 'aalodah': ' Alia Al-Hathloul did well in reminding the rest of the detainees especially Islamists like Sheikh Salman Al-Awda in order to ignore his attempt to separate prisoners of conscience based on their attitudes Prisoner of conscience Prisoner of conscience! There is no difference between a liberal opinion and an Islamic opinion aalodah'	(0.0) Statement from Authority Account 'aalodah': ' After a thousand and one nights I can finally see the face of my dear sister Loujain but I cannot say that she is free Also I cannot forget that Al-Awda Al-Sadhan Al-Rabiah and others are still subjected to great injustice Life lesson: No right is lost there are demands behind it '	(-0.8) Statement from Authority Account 'aalodah': 'The day after tomorrow Friday wait for us moslehun Salman Return Conference '	(0.8) Statement from Authority Account 'aalodah': ' The virtual international conference 'Contemporary Reformers: Salman Al-Awda as a Model' with the participation of a global elite of academics jurists and politicians February 12 and 13 2021 Conference program and broadcast links Salman Al-Awda Conference''
AuRED_159	REFUTES	NOT ENOUGH INFO	News about the death of Sheikh Abdul Rahman Al-Sudais Friday of Rage 25 September	(0.0) Statement from Authority Account 'ReasahAlharmain': 'General President of General Managers: The pilgrim must find the finest and best services Presidency of The Haramayn Affairs '	(0.0) Statement from Authority Account 'ReasahAlharmain': 'The President-General praises the valor and professionalism of the security men and the heroic work they did in arresting terrorist elements and exposing their failed and miserable plans Presidency of The Haramain Affairs '	(0.0) Statement from Authority Account 'ReasahAlharmain': 'In commemoration of the heroic role of the martyrs of duty the General President directs the appointment of one of their sons '	(0.0) Statement from Authority Account 'ReasahAlharmain': 'The President-General: Our soldiers made the highest sacrifices to uphold the word of monotheism and defend the sanctities of our dear country '	(0.0) Statement from Authority Account 'ReasahAlharmain': 'The President-General discusses preparations to receive Umrah pilgrims and worshipers with the field executive team for the Umrah season Presidency of The Haramayn Affairs '

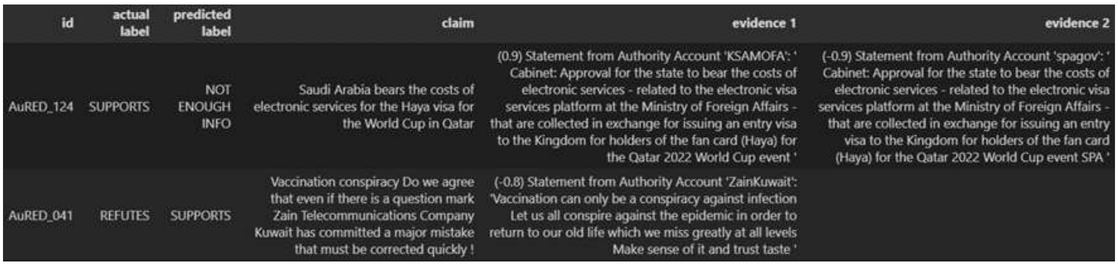
Figure 7.1: Misclassifications by Llama-3.1-405B-Instruct on our dataset in the RQ2 setting.

would give enough context to accurately classify the rumor, but the machine translation to English loses some critical information. It is also possible that not including media content posted along with a tweet (that would be available to a human labeling the data) contributes to missing some necessary information. In this specific case, the model was unable to detect that the claim of a person's death is contradicted by the fact that the person is posting on social media. Our setup only incorporates text content, not social media metadata like time of posting, and this is why the correct label could not be found.

Image 7.1 shows the texts passed to the model (both claim and each evidence) and the models' prediction and confidence represented by the score prepended to the evidence text.

Llama-3.1-70B-Instruct: the verification stage using this model produces 6 wrong overall labels on our dataset. 3 misclassifications of type CASE 2 ("being too cautious"), and 3 misclassifications of type CASE 1 ("producing the opposite label"). Do demonstrate a CASE 1 and CASE 2 misprediction, we will look at two rumors here, which are also displayed as in Image 7.2:

- In the first rumor, the verification stage produces a "being too cautious" CASE 2 misprediction. This is due to the two available pieces of evidence receiving opposing label predictions, causing a tie. However, both evidence tweets contain basically the same text, except they were posted from difference accounts. A possible interpretation is that the LLM only associated one of the account handles with "Saudi Arabia", which is named in the claim. Additional author information could maybe improve the results in this type of fail case, provided the name or bio contain the required information to successfully make this association.
- The second rumor is a misprediction, assigning the opposite label compared to the one assigned by the human labeler. There is only one piece of evidence, so presumably the human labeler deemed the evidence tweet sufficient to refute the claim. Looking at the claim and evidence texts, it seems again that some information was



id	actual label	predicted label	claim	evidence 1	evidence 2
AuRED_124	SUPPORTS	NOT ENOUGH INFO	Saudi Arabia bears the costs of electronic services for the Haya visa for the World Cup in Qatar	(0.9) Statement from Authority Account 'KSAMOFA': ' Cabinet: Approval for the state to bear the costs of electronic services - related to the electronic visa services platform at the Ministry of Foreign Affairs - that are collected in exchange for issuing an entry visa to the Kingdom for holders of the fan card (Haya) for the Qatar 2022 World Cup event '	(-0.9) Statement from Authority Account 'spagov': ' Cabinet: Approval for the state to bear the costs of electronic services - related to the electronic visa services platform at the Ministry of Foreign Affairs - that are collected in exchange for issuing an entry visa to the Kingdom for holders of the fan card (Haya) for the Qatar 2022 World Cup event SPA '
AuRED_041	REFUTES	SUPPORTS	Vaccination conspiracy Do we agree that even if there is a question mark Zain Telecommunications Company Kuwait has committed a major mistake that must be corrected quickly !	(-0.8) Statement from Authority Account 'ZainKuwait': 'Vaccination can only be a conspiracy against infection Let us all conspire against the epidemic in order to return to our old life which we miss greatly at all levels Make sense of it and trust taste '	

Figure 7.2: Two selected misclassifications by Llama-3.1-70B-Instruct on our dataset in the RQ2 setting.

lost in the machine translation from Arabic to English, and possibly by the preprocessing. The claim alleges a "vaccination conspiracy" by Zain Telecommunications Company, while the evidence tweet is a post by Zain Telecommunications Company opposing "vaccination conspiracies". This information, however, does not seem to come across properly in the translated text of the tweet – especially because wordplay is involved in the evidence tweet text. Image 7.3 shows the source tweet on Twitter.

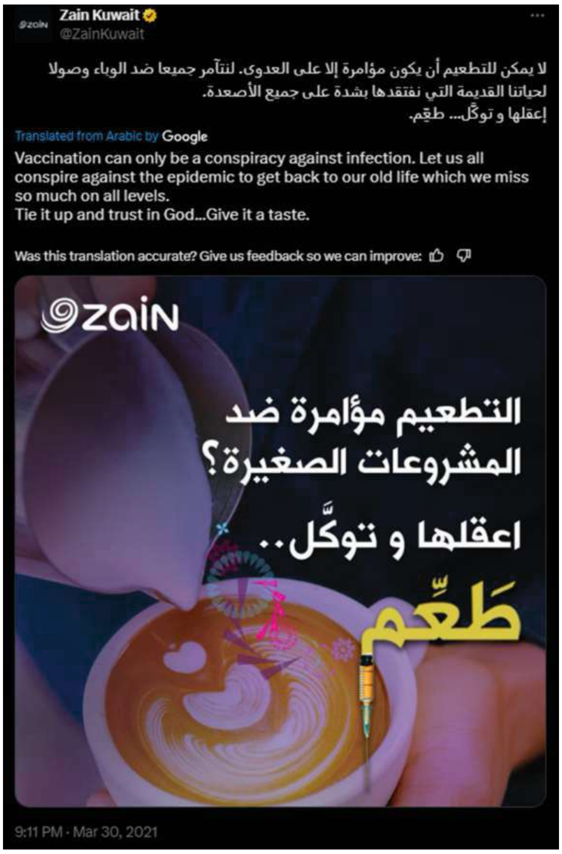
Llama-3.1-8B-Instruct: the verification stage using this model produces 21 wrong overall labels on our dataset. 11 misclassifications of type CASE 2 ("being too cautious"), and 10 misclassifications of type CASE 1 ("producing the opposite label"). To sample another type of misprediction, here we will be looking at one individual rumor for this model, displayed in Image 7.4. This example shows that the model is simply weaker than the other models with more parameters in this family of models. The evidence tweet is very clearly opposing the claim, even saying verbatim "[...] This rumor is not true". The model still classified this claim-evidence pairing as SUPPORTS.

In Table 7.8 we list all the misclassifications (and their type) produced by each model, grouped by rumor id. We can see that every rumor that the most powerful 405B got wrong, the weaker models also got wrong, while the 405B model judged everything else correctly (based on the overall rumor label generated). This gives us an idea of the performance other than pure Macro-F1-score. The more powerful models make fewer errors, and they are generally of the more "benign" error type CASE 2 ("judging too cautiously").

Some of the issues highlighted here will be further discussed in Section 8.2.

7.6 Inclusion in GPT Training Data

Another basic experiment we conducted aims to find our if the predictions the GPT models produce are simply memorized, or if they are actually generated depending on the data we provide via the prompt. To examine the inclusion of our CheckThat! Lab



Zain Kuwait
@ZainKuwait

لا يمكن للتطعيم أن يكون مؤامرة إلا على العدوى. لنتآمر جميعا ضد الوباء وصولا
لحياتنا القديمة التي نفتقدها بشدة على جميع الأصعدة.
إعقلها و توكّل... طعّم.

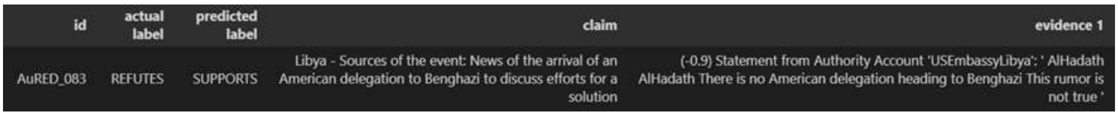
Translated from Arabic by Google

Vaccination can only be a conspiracy against infection. Let us all conspire against the epidemic to get back to our old life which we miss so much on all levels.
Tie it up and trust in God...Give it a taste.

Was this translation accurate? Give us feedback so we can improve:

9:11 PM · Mar 30, 2021

Figure 7.3: Original tweet from “Zain Telecommunications Company” opposing “vaccination conspiracies” (`https://twitter.com/ZainKuwait/status/1376975348114526215`, accessed 22.09.2024).



id	actual label	predicted label	claim	evidence 1
AuRED_083	REFUTES	SUPPORTS	Libya - Sources of the event: News of the arrival of an American delegation to Benghazi to discuss efforts for a solution	(-0.9) Statement from Authority Account 'USEmbassyLibya': ' AlHadath AlHadath There is no American delegation heading to Benghazi This rumor is not true '

Figure 7.4: One selected misclassification by Llama-3.1-8B-Instruct on our dataset in the RQ2 setting.

Rumor ID	**405B**	**70B**	**8B**
AuRED_014			Case 1
AuRED_089			Case 1
AuRED_135			Case 2
AuRED_141			Case 1
AuRED_152			Case 1
AuRED_157	Case 2	Case 1	Case 1
AuRED_094			Case 1
AuRED_002			Case 1
AuRED_051			Case 2
AuRED_096			Case 2
AuRED_149			Case 1
AuRED_098			Case 2
AuRED_138			Case 2
AuRED_087			Case 2
AuRED_079			Case 2
AuRED_159	Case 2	Case 2	Case 2
AuRED_131			Case 1
AuRED_041		Case 1	Case 2
AuRED_142			Case 2
AuRED_099			Case 2
AuRED_083			Case 1
AuRED_085		Case 2	
AuRED_124		Case 2	
AuRED_116		Case 1	

Table 7.8: Rumors each Llama-3.1 model variant misclassified in the RQ2 setting described in Section 7.2.

Task 5 dataset (and its labels) in the OpenAI GPT models' training data, we took the following steps:

1. We flip the labels from SUPPORTS to REFUTES, and vice versa. We also drop any rumors with labels that are NOT ENOUGH INFO, and drop the timeline part of the data. We will only need the claim, label and relevant evidence for this experiment.

2. We manually alter the text contents of the new rumors so that the claim-evidence pairings match the inverted, new labels semantically. The changes made to the rumors can be checked on our GitHub repository[1]. We also tried to keep the

[1] The individually edited rumor JSON files are located under the */data* folder at this link: github.com/LuisKolb/thesis/tree/main/lkae/RQ2/inverting-check

changes minimal, like inverting or negating a verb to invert the meaning of an entire sentence.

3. Run the verification stage as in RQ2, with claim-evidence pairings that consist of our manually altered data.

If our dataset is included in the GPT models, and the label predictions to our claim-evidence pairings are “memorized” instead of actually inferred from the input text we provide, we would expect to observe an overall pattern across overall predictions on a rumor where the GPT model predicts the original label, instead of the “new” inverted label.

In our results, this does not occur often. Performance scores are somewhat degraded overall (compared to the experiments with the original datasets), but this is likely caused by the manual edits we added. Since we also drop NOT ENOUGH INFO labels, this is not a valid choice anymore, but it is still available in the prompt to the GPT model. Overall, we can state that the predictions don’t seem to be produced from the latent knowledge (or “memory”) available to the GPT model via its training data, but rather are generated based on the supplied evidence.

CHAPTER 8

Conclusion and Discussion

In this chapter, we summarize the findings and contributions of this thesis, discuss important aspects and concerns relevant to the approach we present, and highlight avenues for further research.

8.1 Contributions

We present an approach to verifying authority support or denial of rumors using evidence retrieved from the authorities' account.

We also contribute the conference paper (working notes) we published with our participation in the CheckThat! Lab at CLEF 2024: "AuthEv-LKolb at CheckThat! 2024: A Two-Stage Approach To Evidence-Based Social Media Claim Verification" [KH24]. The code we used in that paper is available as open-source online via GitHub[1] under a GPL-3.0 license.

The code we produced for this thesis, along with our results, is also available as open-source online via GitHub[2] under an MIT license. However, the main contributions of this thesis are the answers to our research questions, which we summarize again here:

- **Research Question 1:** *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?* The best approach we found was a re-ranking pipeline of BM25 and NV-Retriever-v1, which worked well with any configuration of the dataset. The highest score we obtained was a Recall@5 of 0.714 using the default Okapi BM25 implementation, and the NV-Retriever-v1 embedding model.

[1] github.com/LuisKolb/clef-2024-authority
[2] github.com/LuisKolb/thesis

- **Research Question 2:** *To what extent can a claim, given a list of tweets ("evidence"), accurately be identified as being supported by the evidence (true), being refuted by the evidence (false), or being unverifiable (not enough evidence to verify it being available)?* The best approach we found given the RQ2 setting with perfect evidence was the Llama-3.1-405B-Instruct model, obtaining a Macro-F1-score of 0.985. The 70B variant of this model family follows closely behind, with a Macro-F1-score of 0.960.

- **Research Question 3:** *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?* The best approach we found uses the GPT-4o model and the re-ranking retrieval pipeline of BM25 and NV-Retriever-v1, achieving top scores across all four dataset variations, with a maximum Macro-F1-score of 0.858. Closely following is the less powerful GPT-4o-mini model, with a maximum Macro-F1-score of 0.810. Generally, the less powerful models (in terms of model size) perform worse, receiving progressively lower scores.

8.2 Discussion

There are some caveats and issues, some of which we already mentioned in previous Sections. Here, we want to highlight and discuss some possible benefits and drawbacks of our approach:

Dataset issues & preprocessing: as we stated in Section 7.8, some misclassifications could have been caused by the way the data was machine translated, and possibly via our preprocessing. The original evidence tweets were all posted in Arabic, and the machine translation provided by the dataset authors contains texts that are sometimes ambiguous in English, while they might not be ambiguous in the original language. Preprocessing could also have lost some necessary information within the texts, which might be exacerbated by the unnatural sentence structure inherent to the translations from Arabic to English.

Media & social network metadata in evidence tweets: while some evidence tweets might obviously contain the information necessary to verify a claim, the human viewer (or labeler, in our case) has access to more information than we provide to our verification stage. We don't make use of images attached to evidence to verify a claim, nor do we include time of posting or whether the evidence tweet is a reply to a tweet containing the claim.

In general, the dataset construction is not ideal or representative of how the approach to fact-checking with authority evidence would be implemented in a real, production system (for example, at a social media company or as a tool for professional fact-checkers). More information or signals would need to be included, as we discussed earlier.

Public Trust in AI Systems and LLMs: for any fact-checking system to be effective, there needs to be trust by the user. Given that we explore the application of LLMs to the domain of fact-checking, it is crucial to be transparent how we use these new technologies, building trust [FAI+24]. It is also important to stress that our approach does not use LLMs as a source of truth. We utilize LLMs as highly capable models that have learned complex patterns of natural language better than other available approaches, and are more capable of dealing with the casual, informal language used on social media, for example. These models perform very well as Natural Language Inference machines, as our results have shown – and this is all we are using them for.

Implications of using Authority Sources for Fact-Checking: another important point we will reiterate here is that our approach does not do "fact-checking" as it is traditionally understood. We don't use knowledge bases or similar information providers as sources of truth to verify factual claims about our world. Rather, our approach should give users the ability to easily find the stance that authorities relevant to a claim have toward that claim: whether the authorities support or refute ("deny") a rumor. Whether the user actually trusts the authority, or whether they themselves would consider the authority account an actual authority on a given rumor is up to the user to decide. As we have mentioned previously, finding authorities is a separate task and not covered in this thesis. However, in this thesis we work under the premise that an authority is not necessarily a government account on social media, but we use a broader definition of "authority" that is described in Section 4.2.1, along with some illustrative examples and references.

Usage in the Real World: Deploying our approach as-is using a relatively large model like Llama-3.1-405B would likely be very expensive computationally. There is definitely still a cost-benefit trade-off, both in verification and retrieval (Could BM25 be "good enough"? Is re-ranking with NV-Retriever-v1 beneficial enough for the desired use case to justify the added computational cost?). Depending on the use case and "narrowness" of topics expected when using our approach in a real production environment, it could be beneficial to use a weaker model with fewer parameters and fine-tune it on the task and the platforms' own data, potentially improving performance and lowering cost.

8.3 Future Work

This area of research is rapidly evolving, attempting to address some of the concerns arising from the ever-increasing influence social media has both on our lives and public discourse. For future work, we propose the following issues for investigation:

Multimodality: tweets (and posts on social media in general) often include images or videos. In some cases, these may contain information that is required to verify a claim accurately. Extracting this information is possible in multiple ways, depending on the architecture of the systems for retrieval and verification. For example, a simple approach could be Optical Character Recognition (OCR) to extract text from images, which could then be added to the post. Some accounts even provide alt-text for media in their posts, which should be descriptive. Even directly including images or video would be feasible given the multimodal capabilities of popular LLMs today.

Integration of social media-specific signals: posts (and authority evidence) on social media platforms are uploaded or created within a context, for example as a reply to another post, or (on Twitter specifically) as a "quote tweet". These social network, social-media specific signals could be very pertinent to include – especially for relevance to the claim and possibly useful in the verification stage [HAAY23].

Multilingualism: the dataset we used for this thesis was provided both in Arabic and English, where the English version was translated using Machine Translation (MT). Our general approach is not specific to data in the English language, and could be adapted and evaluated in other languages.

Explainability: trust in fact-checking systems is important for their acceptance by the general public. Trust and confidence in "AI" and LLMs by the general public are not well-established due to issues like, for example, hallucinations in LLMs. It is important to mention that our approach does not verify factual truths about our world but aims to classify support or refutal of claims from accounts with unique qualification to opine on a topic. Generating explanations along with a claim-evidence label prediction could improve explainability, as those textual explanations could be provided along with the prediction, helping humans understand how this specific predicted label came to be – increasing trust in the system.

Chain-of-Thought Prompting: as mentioned in the previous suggestion on explainability, generating explanations before generating the label prediction on a claim-evidence pairing could improve the results as a side-effect of generating explanations. Wei et al. [WWS^{+}23] explored the pattern "chain-of-thought" prompting to improve reasoning in conversational models and saw improved results. The paper by Cruickshank et al. [CN24] evaluated multiple prompting schemes, and found that chain-of-thought prompting (and few-shot prompting, where representative examples are given to the LLM before the actual input) improve LLM performance.

List of Figures

List of Tables

Bibliography

[ALA23] Nora Alturayeif, Hamzah Luqman, and Moataz Ahmed. A systematic review of machine learning techniques for stance detection and its applications. *Neural Computing & Applications*, 35(7):5113–5144, 2023.

[ASLA20] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. Generating Fact Checking Explanations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7352–7364, Online, July 2020. Association for Computational Linguistics.

[BMR$^+$20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, May 2020.

[BW07] Shane Bergsma and Qin Iris Wang. Learning Noun Phrase Query Segmentation. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 819–826, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[CN24] Iain J. Cruickshank and Lynnette Hui Xian Ng. Prompting and Fine-Tuning Open-Sourced Large Language Models for Stance Classification, March 2024. arXiv:2309.13734 [cs].

[CTPL23] Yuwei Chuai, Haoye Tian, Nicolas Pröllochs, and Gabriele Lenzini. The Roll-Out of Community Notes Did Not Reduce Engagement With Misinformation on Twitter, July 2023. arXiv:2307.07960 [cs].

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019. arXiv:1810.04805 [cs].

[DDMR10] Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches – Erratum. *Natural Language Engineering*, 16(1):105–105, January 2010.

[DJP+24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 Herd of Models, July 2024.

[DS23] Mitchell DeHaven and Stephen Scott. BEVERS: A General, Simple, and Performant Framework for Automatic Fact Verification, March 2023. arXiv:2303.16974 [cs].

[FAI+24] Md Meftahul Ferdaus, Mahdi Abdelguerfi, Elias Ioup, Kendall N. Niles, Ken Pathak, and Steven Sloan. Towards Trustworthy AI: A Review of Ethical and Robust Large Language Models, June 2024. arXiv:2407.13934 [cs].

[FFGGSdH24] Guglielmo Faggioli, Nicola Ferro, Petra Galuščáková, and Alba García Seco de Herrera, editors. *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*. CLEF 2024. Grenoble, France, 2024. Publication Title: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum.

[GA19] L. Graves and M. Amazeen. *Fact-checking as idea and practice in journalism*. Oxford University Press, 2019.

[GSV22] Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, February 2022.

[HAAY23] Suhaib Kh Hamed, Mohd Juzaiddin Ab Aziz, and Mohd Ridzwan Yaakub. A review of fake news detection approaches: A critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion. *Heliyon*, 9(10):e20382, October 2023.

[HE24] Fatima Haouari and Tamer Elsayed. Are authorities denying or supporting? Detecting stance of authorities towards rumors in Twitter. *Social Network Analysis and Mining*, 14(1):34, January 2024.

[HEM23] Fatima Haouari, Tamer Elsayed, and Watheq Mansour. Who can verify this? Finding authorities for rumor verification in Twitter. *Information Processing & Management*, 60(4):103366, July 2023.

[HES24] Fatima Haouari, Tamer Elsayed, and Reem Suwaileh. Overview of the CLEF-2024 CheckThat! Lab Task 5 on Rumor Verification using Evidence from Authorities. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*, Grenoble, France, 2024.

[HLGC21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, October 2021. arXiv:2006.03654 [cs].

[HSW+21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, October 2021. arXiv:2106.09685 [cs].

[JW19] Sarthak Jain and Byron C. Wallace. Attention is not Explanation, February 2019.

[KBAK96] John R. Koza, Forrest H. Bennett, David Andre, and Martin A. Keane. Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In John S. Gero and Fay Sudweeks, editors, *Artificial Intelligence in Design '96*, pages 151–170. Springer Netherlands, Dordrecht, 1996.

[KCS24] Uku Kangur, Roshni Chakraborty, and Rajesh Sharma. Who Checks the Checkers? Exploring Source Credibility in Twitter's Community Notes, June 2024. arXiv:2406.12444 [cs].

[KH24] Luis Kolb and Allan Hanbury. AuthEv-LKolb at CheckThat! 2024: A Two-Stage Approach To Evidence-Based Social Media Claim Verification. 2024.

[KRV22] Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. ProoFVer: Natural Logic Theorem Proving for Fact Verification, July 2022. arXiv:2108.11357 [cs] version: 2.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. Publisher: Nature Publishing Group.

[Lid01] Elizabeth Liddy. Natural Language Processing. *School of Information Studies - Faculty Scholarship*, January 2001.

[LLG^{+}19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, October 2019. arXiv:1910.13461 [cs, stat].

[LOG^{+}19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019. arXiv:1907.11692 [cs].

[LRX^{+}24] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models, May 2024. arXiv:2405.17428 [cs].

[LXSL20] Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. Fine-grained Fact Verification with Kernel Graph Attention Network. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351, Online, July 2020. Association for Computational Linguistics.

[Mac09] Bill MacCartney. NATURAL LANGUAGE INFERENCE. 2009.

[Mah19] Batta Mahesh. *Machine Learning Algorithms -A Review*, volume 9. January 2019. Journal Abbreviation: International Journal of Science and Research (IJSR) Publication Title: International Journal of Science and Research (IJSR).

[MAS$^+$24] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models, October 2024.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. arXiv:1301.3781 [cs].

[MOX$^+$24] Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. NV-Retriever: Improving text embedding models with effective hard-negative mining, July 2024. arXiv:2407.15831 [cs].

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, New York, 2008. OCLC: ocn190786122.

[PF24] Andrea Pasin and Nicola Ferro. SEUPD@CLEF: Team Axolotl on Rumor Verification using Evidence from Authorities. 2024.

[Pro22] Nicolas Proellochs. Community-Based Fact-Checking on Twitter's Birdwatch Platform. *Proceedings of the International AAAI Conference on Web and Social Media*, 16:794–805, May 2022.

[RBJ$^+$22] Guilherme Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. In Defense of Cross-Encoders for Zero-Shot Retrieval, December 2022. arXiv:2212.06121 [cs].

[SBDSMN20] Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. That is a Known Lie: Detecting Previously Fact-Checked Claims. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3607–3618, Online, July 2020. Association for Computational Linguistics.

[SKO$^+$21] Michael Sejr Schlichtkrull, Vladimir Karpukhin, Barlas Oguz, Mike Lewis, Wen-tau Yih, and Sebastian Riedel. Joint Verification and Reranking for Open Fact Checking Over Tables. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6787–6799, Online, August 2021. Association for Computational Linguistics.

[Sta22] Georgiana-Daniela Stanescu. Ukraine conflict: the challenge of informational war. 2022.

[TVC$^+$18] James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. The Fact Extraction and VERification (FEVER) Shared Task. In James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal, editors, *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[VM23] Juraj Vladika and Florian Matthes. Scientific Fact-Checking: A Survey of Resources and Approaches, May 2023. arXiv:2305.16859 [cs].

[VR14] Andreas Vlachos and Sebastian Riedel. Fact Checking: Task definition and dataset construction. In Cristian Danescu-Niculescu-Mizil, Jacob Eisenstein, Kathleen McKeown, and Noah A. Smith, editors, *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22, Baltimore, MD, USA, June 2014. Association for Computational Linguistics.

[VRA18] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science (New York, N.Y.)*, 359(6380):1146–1151, March 2018.

[VSP$^+$17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, June 2017. arXiv:1706.03762 [cs].

[Wan17] William Yang Wang. “Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[WWS$^+$23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, January 2023. arXiv:2201.11903 [cs].

[ZAZ21] Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. Automated Fact-Checking: A Survey, September 2021. arXiv:2109.11427 [cs].

[ZYL$^+$18] W.J. Zhang, Guosheng Yang, Yingzi Lin, Chunli Ji, and Madan M. Gupta. On Definition of Deep Learning. In *2018 World Automation Congress (WAC)*, pages 1–5, June 2018.

[ZZL$^+$23] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A Survey of Large Language Models, March 2023.