Informatics

Multilingual and Crosslingual Fact-Checked Claim Retrieval

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

eingereicht von

Iva Pezo, BSc
Matrikelnummer 12224159

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Allan Hanbury
Mitwirkung: Univ.Ass. Dipl.-Ing. Moritz Staudinger

Wien, 24. April 2025

Iva Pezo

Allan Hanbury

Informatics

Multilingual and Crosslingual Fact-Checked Claim Retrieval

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Data Science

by

Iva Pezo, BSc
Registration Number 12224159

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury
Assistance: Univ.Ass. Dipl.-Ing. Moritz Staudinger

Vienna, April 24, 2025

Iva Pezo

Allan Hanbury

Erklärung zur Verfassung der Arbeit

Iva Pezo, 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.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel" habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 24. April 2025

Iva Pezo

Acknowledgements

I want to thank everyone who guided and supported me throughout my studies.

Firstly, I sincerely thank my supervisor, Prof. Allan Hanbury, for his support, guidance, and availability throughout this project. His encouragement led me to participate in the SemEval shared task and contribute to a conference paper, both of which were invaluable learning experiences. I am grateful to Assistant Moritz Staudinger for his continuous support, insightful feedback, and willingness to help whenever needed, which helped me refine my work and overcome challenges along the way.

But most of all, I want to thank my family for their continuous support throughout my education and beyond. And to my friends, in Zagreb and Vienna, for making everything easier.

Kurzfassung

Mit dem zunehmenden Einfluss der sozialen Medien wird es immer wichtiger, die Richtigkeit von Online-Informationen zu gewährleisten. Die automatisierte Überprüfung von Fakten umfasst mehrere Stufen, darunter die Erkennung von Behauptungen, die Priorisierung, das Abrufen von Beweisen, die Vorhersage des Wahrheitsgehalts und die Generierung von Erklärungen. Eine wichtige, aber oft übersehene Komponente ist das Abrufen von bereits geprüften Behauptungen, was zur Bekämpfung von Fehlinformationen beiträgt, indem neue Behauptungen mit bestehenden Faktenprüfungen abgeglichen werden.

In dieser Arbeit entwickeln wir ein mehrsprachiges und sprachübergreifendes System zur Abfrage von faktengeprüften Behauptungen, das auf einer hybriden Abfrage-Pipeline basiert, die lexikalische und dichte Abfragemodelle kombiniert. Wir evaluieren systematisch verschiedene Retrieval- und Reranking-Strategien und zeigen, dass hybride Ensembles Effizienz und Effektivität effektiv ausbalancieren und einzelne Retriever übertreffen. Während Reranking das sprachübergreifende Retrieval signifikant verbessert, bleibt seine Wirkung in einsprachigen Umgebungen begrenzt, was die Effektivität eines gut konzipierten Ensembles gegenüber immer komplexeren Ranking-Ebenen unterstreicht.

Darüber hinaus analysieren wir die Auswirkungen von Vorverarbeitungsschritten, vergleichen Modelle in Bezug auf Abrufleistung, Ausführungszeit, Anzahl der Parameter und Speicherverbrauch und führen eine Fehleranalyse durch, um die wichtigsten Einschränkungen zu ermitteln. Schließlich diskutieren wir mögliche Verbesserungen und zukünftige Forschungsrichtungen, um die Suche nach mehrsprachigen Faktenchecks zu verbessern.

Unser Ansatz wurde bei SemEval-2025 Task 7 angewandt, wo wir Ergebnisse und Erkenntnisse aus unserer Teilnahme präsentieren.

Abstract

With the growing influence of social media, ensuring the accuracy of online information has become increasingly important. Automated fact-checking involves multiple stages, including claim detection, prioritization, retrieval of evidence, veracity prediction, and explanation generation. A crucial yet often overlooked component is retrieving previously fact-checked claims, which helps combat misinformation by matching new claims with existing fact-checks.

In this work, we develop a multilingual and crosslingual fact-checked claim retrieval system based on a hybrid retrieval pipeline that combines lexical and dense retrieval models. We systematically evaluate different retrieval and reranking strategies, demonstrating that hybrid ensembles effectively balance efficiency and effectiveness, outperforming individual retrievers. While reranking significantly enhances crosslingual retrieval, its impact in monolingual settings remains limited, highlighting the effectiveness of well-designed ensembling over increasing complex ranking layers.

Additionally, we analyze the impact of preprocessing steps, compare models in terms of retrieval performance, execution time, number of parameters and memory usage, and conduct an error analysis to identify key limitations. Finally, we discuss potential improvements and future research directions to enhance multilingual fact-check retrieval.

Our approach was applied to SemEval-2025 Task 7, where we present results and insights gained from our participation.

Contents

CHAPTER 1

Introduction

Social media has revolutionized how information is consumed and shared, providing instant access to news and a wide range of perspectives. Unlike traditional media, where content is filtered through editorial standards, social media allows anyone to publish and share information, often without verification. As a result, the spread of misinformation has accelerated, making it increasingly difficult to distinguish between true and misleading claims. Misinformation is more than just a source of confusion; it has the ability to shape public opinion and deepen political divides. Its impact can be seen in real-world events [HBMC24], from public health crises fueled by false claims to election interference driven by misleading narratives.

Traditional fact-checking, by journalists and expert organizations, has long been used to combat misinformation. Although reliable, this manual approach is slow and labour-intensive, and lacks the scalability to keep up with the speed and volume of online information. Adding to the challenge, misinformation is frequently reformated and reshared on different platforms, often altered in wording, format, length, or even language, making detection and verification even more difficult. This is an even bigger issue in low-resource languages, where fact-checking efforts may be limited or less accessible [BBVEF$^+$24, KLPRM21].

To address these challenges, automated fact-checking systems have appeared as a promising solution, using advances in natural language processing and information retrieval to detect and verify claims efficiently. A key component of these systems is previously fact-checked claim retrieval (PFCR) [PSM$^+$23]— a process that matches new claims with fact-checks that have already been verified. This step is crucial for preventing the spread of recurring misinformation, reducing the need for redundant fact-checking efforts, and ensuring that users quickly access verified information [PZ24].

In this thesis, we present our system, which employs a hybrid retriever-reranker architecture for verified claim retrieval. Our approach focuses on zero-shot retrieval [SLG$^+$24, TRR$^+$21], avoiding model training or fine-tuning to ensure applicability across diverse topics, languages, and platforms. By utilizing out-of-the-box pre-trained models, our system delivers competitive performance while minimizing resource demands and development overhead. This strategy highlights the viability of pre-trained models as effective tools for multilingual and crosslingual fact-checking tasks, demonstrating their ability to generalize without task-specific adaptations. Further, we analyze the impact of preprocessing, compare retrieval efficiency across different model configurations, and conduct error analysis to identify key limitations.

We participated in SemEval-2025 Shared Task 7[1] which tackles the challenge of multilingual and crosslingual PFCR. The task is divided into two subtasks: monolingual and crosslingual retrieval. In the monolingual subtask, the search space is restricted to fact-checked claims in the same language as the query claim. In contrast, the crosslingual subtask allows retrieval across multiple languages, enabling fact-checks in one or more languages to be retrieved for a query in a different language. The monolingual subtask includes data for English, German, French, Arabic, Spanish, Portuguese, Malay, and Thai, with Polish and Turkish added to the test set.

1.1 Research Questions

This thesis answers the following research questions ("RQs"):

• **RQ1**: Which preprocessing and data augmentation techniques (such as translation, stop-word removal, stemming, and spell-correction) most effectively enhance the performance of BM25 for monolingual claim retrieval, and how do these techniques impact different types of retrieval errors, such as false positives and false negatives?

• **RQ2**: How do translation-based approaches for crosslingual retrieval compare to multilingual models in terms of S@10[2], execution time and memory usage?

• **RQ3**: Which retriever-reranker combinations yield the most consistent performance for monolingual claim retrieval in terms of S@10 across languages?

1.2 Structure of the Thesis

This thesis is structured as follows:

- Chapter 2 provides background on natural language processing (NLP) and information retrieval (IR) and offers an overview of the retrieval and the fact-checking pipelines.

[1] https://disai.eu/semeval-2025/

[2] Success-at-k is a metric that evaluates whether at least one relevant document appears within the top k retrieved results.

- Chapter 3 describes the fact-checking datasets and gives an overview of the related work.
- Chapter 4 describes the used datasets, evaluation measures and the experiment details.
- Chapter 5 explains the key modules of our system.
- Chapter 6 analyses the results of the monolingual subtask.
- Chapter 7 analyses the results of the crosslingual subtask.
- Chapter 8 sums up the conclusions and possible future research directions.

CHAPTER 2

Background

In this chapter, we provide background of the key concepts from the domains of natural language processing and information retrieval necessary for understanding the methods and approaches explored in our work.

2.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of computer science and artificial intelligence (AI) that focuses on enabling computers to process, understand, and generate natural language.

The roots of NLP can be traced back to the 1950s, when Alan Turing published "Computing Machinery and Intelligence" [TUR50], in which he proposed the now-famous "Turing Test." While the Turing Test remains an important philosophical benchmark, modern NLP systems are evaluated using more practical metrics, such as accuracy in translation, question-answering, and language modeling tasks.

Between the 1950s and the early 1990s symbolic NLP was the main approach for solving NLP tasks [Jon94]. The main premise of symbolic NLP is that a computer can emulate natural language understanding if we provide it with a set of rules and it applies those rules to the data it confronts.

By the 1980s and 1990s, the development of machine learning algorithms led to a shift in NLP from rule-based systems to statistical methods. This transformation was driven by the increasing availability of large text datasets and increased computational power, which enabled data-driven models to outperform manually designed linguistic rules.

In the 2000s, Artificial Neural Networks(ANNs) [WC03], inspired by the structure of the human brain, became the foundation of deep learning advancements in NLP. Sequence-based models like Recurrent Neural Networks (RNNs) [She20] and Long Short-Term

Memory (LSTM) [HS97] networks became widely used for language modelling, machine translation, and speech recognition. Despite their success, RNNs had a limitation in capturing long-range dependencies in text.

In the early 2010s, word embeddings such as Word2Vec [MCCD13], GloVe [PSM14], and fastText [BGJM16] were introduced. These methods represent words as dense vectors in continuous vector spaces, capturing semantic relationships and contextual meaning more effectively than previous frequency-based approaches.

By the late 2010s, transformer-based models revolutionised NLP. Starting with Google's Bidirectional encoder representations from transformers (BERT) [DCLT18], and followed by models like GPT (Generative Pre-trained Transformer) [BMR+20] and T5 (Text-to-Text Transfer Transformer) [RSR+19]. These architectures introduced self-attention mechanisms, setting new standards in tasks like machine translation, text summarization, and conversational AI. Since then, transformers have become the foundation of modern NLP, surpassing older statistical and rule-based methods in both accuracy and efficiency.

2.2 Information Retrieval and the Retrieval Pipeline

Information retrieval (IR) [MRS08] is a field of computer science and information science that focuses on identifying and retrieving relevant information from large collections of unstructured or semi-structured data. The goal of IR is to efficiently provide users with the most relevant results (documents, resources) based on their queries. As multiple documents may be relevant, they are often ranked according to their relevance score to the user's query.

The term "information retrieval" was first introduced in the 1950s [Moo52], when automated retrieval systems were developed to replace manual indexing.

The earliest IR systems from the 1950s and 1960s relied on Boolean retrieval models, where documents were retrieved based on exact keyword matches. In the 1960s, IBM developed the Storage and Information Retrieval System (STAIRS) [SL65], which introduced key IR concepts such as the vector space model and term weighting.

By the 1970s and 1980s, probabilistic retrieval models emerged, leading to the development of models like BM25 [RWJ+94], which remains a strong baseline for modern IR tasks. These probabilistic models improved upon earlier approaches by incorporating term frequency and document length into relevance calculations, making retrieval more robust.

In the 1990s, the rise of the World Wide Web led to the development of early search engines that helped users navigate through the expanding online information space. Web search engines needed to rank documents efficiently, leading to the development of link-analysis-based methods such as PageRank. Unlike traditional IR models, which relied on term frequency, PageRank [BP98] introduced the concept of authority ranking, using hyperlink structures to assess a webpage's importance.

Since the 2000s, IR has evolved significantly, driven by advancements in machine learning and NLP. Traditional term-based retrieval models, such as BM25, have been augmented with neural ranking models that leverage deep learning to improve document relevance estimation. The introduction of deep learning models enabled vector-based retrieval using dense representations rather than sparse keyword-based methods. Transformer-based models such as BERT [DCLT19] revolutionized retrieval by allowing contextualized word embeddings, significantly improving query understanding and document ranking. Neural IR methods, including dense retrieval models, replaced traditional sparse models by leveraging pre-trained embeddings to retrieve semantically relevant documents.

In the context of this thesis, we are retrieving fact-checked claims from a large collection (corpus). The queries used in this work are the claims extracted from social media posts. To determine the relevance between the queries (posts) and the documents (fact-checked claims), we calculate a similarity score between every query-document (claim - fact-check) combination in the corpus. In the ideal case, the most relevant documents have the highest similarity score for a given query.

Modern information retrieval systems commonly adopt a retriever-reranker architecture, which combines both lexical and semantic retrieval models to efficiently find and prioritize relevant documents. This two-stage approach ensures a balance between speed, recall, and ranking precision by first retrieving a broad set of candidate documents and then refining their ranking through a more computationally intensive process [KZL+20].

2.2.1 Retrieval Stage

The retrieval stage is responsible for retrieving an initial candidate set of documents that are most relevant to the user's query. Typically, retrieval models return the top-k results, where k varies depending on the system's constraints and goals. This step prioritizes speed and recall, ensuring that no potentially relevant documents are missed [CFG+21].

Retrieval models can be broadly classified into two categories:

1. Lexical Retrieval (Sparse Representations)
2. Semantic Retrieval (Dense Representations)

Lexical Retrieval

Traditional text retrieval systems rely on exact term matching between queries and documents. The ranking of retrieved documents is typically based on the frequency of query terms within a document, as well as their inverse document frequency (for models like TF-IDF and BM25), without considering term order, semantics, or contextual relationships. These systems work well for retrieving information when the exact wording of the query matches the document's content but struggle in cases where meaning is conveyed differently.

Term-based retrieval systems face several challenges, including polysemy (words with multiple meanings), synonymy (different words with the same meaning), and lexical gaps (cases where a query and relevant documents use different vocabulary). These limitations can lead to incomplete or inaccurate retrieval.

Among the traditional IR methods, TF-IDF (Term Frequency-Inverse Document Frequency) [SJ88] and BM25 (Best Matching 25) [RWJ+94] are widely used due to their effectiveness in ranking documents based on term importance and document relevance.

TF-IDF is a statistical measure used to evaluate the importance of a word within a document relative to a larger corpus. It balances term frequency (TF) — how often a word appears in a document — with inverse document frequency (IDF) — how rare a word is across all documents. Words that appear frequently in a single document but rarely in others have a higher importance, allowing relevant documents to be ranked higher.

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where:

- t is the term (word),
- d is the document,
- $\text{TF}(t, d)$ is the term frequency, representing how often term t appears in document d,
- $\text{IDF}(t)$ is the inverse document frequency, reducing the weight of common words.

Term Frequency (TF) is given by:

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

where:

- $f_{t,d}$ is the number of times term t appears in document d,
- and the denominator normalizes it by the total count of all terms t' in d.

Inverse Document Frequency (IDF) is calculated as:

$$\text{IDF}(t) = \log\left(\frac{N}{1 + \text{DF}(t)}\right)$$

where:

- N is the total number of documents in the corpus,
- $\mathrm{DF}(t)$ is the number of documents containing term t. The "+1" in the denominator prevents division by zero.

BM25 is a ranking function built upon TF-IDF. It improves TF-IDF by introducing a saturation function and document length normalization. Unlike term frequency scoring, BM25 accounts for repeated occurrences of a term, preventing long documents from being unfairly favoured. It also includes a parameter (b) that controls the weight assigned to term frequency and document length. BM25 is one of the most effective and widely used retrieval models and is often used as a baseline in modern IR research, in our approach, we use BM25 as a lexical retriever.

$$\mathrm{BM25}(d,q)=\sum_{t\in q}\mathrm{IDF}(t)\cdot\frac{f_{t,d}\cdot(k_1+1)}{f_{t,d}+k_1\cdot(1-b+b\cdot\frac{|d|}{\mathrm{avgdl}})}$$

where:

- q is the query containing multiple terms t,
- d is the document,
- $f_{t,d}$ is the frequency of term t in document d,
- k_1 is a hyperparameter controlling term frequency scaling (typically set between 1.2 and 2.0),
- b is a hyperparameter controlling document length normalization (typically set to 0.75),
- $|d|$ is the length of document d (word count),
- avgdl is the average document length in the corpus.

The Inverse Document Frequency (IDF) component in BM25 is defined as:

$$\mathrm{IDF}(t)=\log\left(\frac{N-\mathrm{DF}(t)+0.5}{\mathrm{DF}(t)+0.5}+1\right)$$

where:

- N is the total number of documents in the corpus,
- $\mathrm{DF}(t)$ is the number of documents containing term t. The "+0.5" term prevents extreme values for rare words.

Semantic Retrieval

Advances in neural networks and pre-trained language models have enabled a shift from traditional term-based retrieval to semantic retrieval, also known as dense retrieval. Unlike term-based methods that rely on exact word matching, semantic retrieval captures the meaning of queries and documents using dense vector representations. This allows for more flexible and accurate retrieval, as it can match conceptually similar text even if the exact words differ, overcoming synonymy, polysemy, and lexical variation [DCLT19].

The retrieval process consists of two main steps:

1. **Encoding**

   Encoding is the foundation of semantic retrieval, transforming raw text into structured numerical representations that enable efficient and meaningful comparisons between queries and documents. The quality of these representations directly impacts retrieval performance, as it determines how well the system captures relationships between queries and documents.

   Traditional encoding methods, such as one-hot and multi-hot encoding, rely on high-dimensional sparse vectors that lack the ability to model semantic similarities. These approaches treat words as independent entities, failing to capture contextual relationships or meanings.

   In contrast, modern embedding techniques generate low-dimensional dense representations that preserve semantic information while improving computational efficiency. These embeddings are produced using pre-trained language models, which map semantically related words and phrases closer together in a continuous vector space. This allows retrieval systems to go beyond simple keyword matching, enabling more accurate and context-aware results. As embeddings are lower in dimensionality in comparison to traditional methods, they allow for faster similarity computations, reduce the memory footprint and computational cost.

   There are two primary encoding architectures used in dense retrieval:

   - ***Bi-Encoders*** independently encode queries and documents into fixed-size dense vectors using a shared encoder [LMR+24]. These embeddings are then compared using a similarity function. By precomputing and storing document embeddings, bi-encoders enable efficient large-scale retrieval. However, because queries and documents are encoded separately, they struggle to capture fine-grained interactions between them.
   - ***Cross-Encoders*** process query-document pairs together, allowing them to capture rich, context-specific interactions. Unlike bi-encoders, cross-encoders do not precompute embeddings — each document must be encoded together with the query at runtime, resulting in significantly higher computational costs.

The choice of the encoding architecture influences the efficiency and effectiveness of semantic retrieval, balancing between computational speed and accuracy during the retrieval.

2. **Similarity Search**

The next step in the retrieval process is identifying the documents whose embeddings are the most similar to the query embedding. To measure this similarity, we use cosine similarity, which is well-suited for text-based retrieval tasks:

Cosine similarity measures the cosine of the angle between two vectors. It is defined as:

$$\text{CosineSimilarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B are the embedding vectors of the query and document. Cosine similarity ranges from -1 (completely opposite) to 1 (identical), with 0 indicating orthogonal (unrelated) vectors [LPS16].

The choice of similarity measure depends on the embedding model and the nature of the retrieval task. Cosine similarity is generally preferred for text-based embeddings, as it emphasizes the directionality of vectors, which aligns best with semantic similarity. In the PFCR task, our goal is to rank fact-checked claims based on their semantic closeness to the query. Since cosine similarity focuses on vector orientation rather than magnitude, it ensures that the embedding space effectively captures semantic relationships, making it well-suited for this retrieval task.

Alternative similarity functions, such as the dot product (which considers magnitude) and Euclidean distance (which measures absolute distance), are less suited for our task. The dot product is more useful when magnitude carries meaning, and Euclidean distance is commonly used in clustering rather than retrieval.

For clarity we note that similarity functions are used with lexical models and bi-encoder-based semantic models. In contrast, cross-encoders generate a single scalar value (a relevance score), eliminating the need for a separate similarity function.

While lexical retrieval is computationally efficient and interpretable, it can suffer from vocabulary mismatches and synonym gaps [KZL+20]. On the other hand, semantic retrieval improves query understanding but may require computationally expensive models and large storage for dense vectors [CFG+21].

First-stage retrieval is designed to be fast and efficient, as it operates over a massive corpus. However, it does not always rank the most relevant documents at the top, especially in cases of ambiguous queries or noisy text. This limitation is addressed by the reranking stage, which refines the ranking by incorporating deeper semantic understanding [KZL+20].

2.2.2 Reranking Stage

The reranking stage refines the initially retrieved documents by assigning more precise relevance scores, therefore improving ranking accuracy and retrieval precision. Unlike the first-stage retrieval, which prioritizes efficiency in retrieving a broad set of relevant candidates, reranking employs more sophisticated models, typically the previously described semantic cross-encoder retrievers, to evaluate the contextual and semantic similarity between queries and documents [YNL21, BBVEF+24]. These models offer deeper semantic understanding compared to first-stage retrievers, allowing for more accurate ranking and better identification of the truly relevant documents.

Due to their computational cost, rerankers cannot efficiently process large volumes of documents. To maintain a balance between efficiency and effectiveness, reranking is applied only to the top candidates retrieved in the first stage. This ensures that computational resources are allocated to the most promising candidates, maximizing ranking precision without significantly impacting performance.

By combining a fast first-stage retrieval with a high-precision reranking model, the retriever-reranker pipeline provides an effective, scalable solution for modern information retrieval tasks [CLMS23, CFG+21].

2.2.3 Ensembler

An ensembler combines retrieval results from multiple models (retrievers or rerankers), taking advantage of the strength of each, such as precise word matching, English-specific retrieval, or multilingual semantic understanding. It is a valuable component for retrieval tasks as it allows the integration of diverse retrieval strategies tailored for specific languages and dataset characteristics. The central component of an ensembler is its aggregation function, which determines how the results from the individual models are combined:

- **Reciprocal Rank Fusion (RRF)** [CCB09] aggregates the rankings assigning a score to each document d based on the reciprocal value of its rank between different retriever models:

  $$R_{\text{score}}(d) = \sum_{r \in R} \frac{1}{k + \text{rank}_r(d)}$$

  where R is the set of retrievers, $\text{rank}_r(d)$ is the rank of document d assigned by the retriever r, and k is a constant added to prevent division by zero.

- **Majority Voting** ranks fact-checks based on their frequency in the top 10 results across multiple retrieval models. Given a set of N retrievers, each retrieving a ranked list of fact-checks, the final score for a fact-check d is computed as:

  $$S(d) = \sum_{i=1}^{N} \nVdash (d \in R_i)$$

where R_i represents the top 10 results retrieved by the i-th model, and $\mathbb{1}(d \in R_i)$ is an indicator function that equals 1 if d appears in R_i, otherwise 0.

This method prioritizes documents retrieved by multiple models, assuming consensus indicates higher relevance. However, it may be ineffective when retrievers have low overlap in their top results, as relevant fact-checks appearing in only a few models may be underrepresented.

- **Exponential Decay Weighting** exponentially penalizes lower-ranked documents:

$$S(d) = w \cdot e^{-\lambda \cdot \text{rank}(d)}$$

where w is a weight factor and λ controls the decay rate. This method favours documents ranked highly by at least one retriever but can overlook relevant lower-ranked documents.

2.3 Fact-Checking

Fact-checking is the process of verifying the accuracy of claims, statements, or information to determine whether they are true, false, misleading, or lacking context. It is commonly applied in journalism, politics, social media, and scientific discourse to fight misinformation and disinformation [VR14].

Traditionally, fact-checking has long been a manual process, relying on experts (journalists, researchers, or fact-checking organizations) to verify claims by referencing them with credible sources such as official documents, or expert opinions. While this approach is thorough, it is also time-consuming and unable to keep up with the rapid spread of information. The delay in verification allows the spread of false narratives, shaping public opinion and creating opportunities for manipulation [SMY+25].

Fact-checkers typically follow a structured process consisting of identifying check-worthy claims, gathering evidence, determining the veracity of the claim, and providing explanations for their findings. The manual approach has been the standard for ensuring accuracy, but its lack of scalability has created a demand for automated solutions.

2.3.1 Automated Fact-Checking

With the growing volume of online misinformation, automated fact-checking has emerged as a promising solution to scale verification efforts supporting human fact-checkers by reducing the time and effort required for verification.

A survey on monolingual, multilingual and crosslingual research [PZ24] discusses the different subtasks of an automated fact-checking pipeline. They state that the pipeline is composed of five components [NCH+21, BBVEF+24]:

- **Claim Detection** identifies statements that are check-worthy, such as social media posts or news articles. This step filters out irrelevant content and focuses on statements that have factual implications.

- **Claim Prioritization** ranks the detected claims based on factors such as potential impact, speed of spread, potential harm, or public interest. This ensures that fact-checkers allocate resources efficiently to address the most impactful misinformation.
- **Retrieval of Evidence** searches for supporting evidence. This step provides the necessary context to assess the truthfulness of a claim.
- **Veracity Prediction** classifies claims based on the retrieved evidence, determining whether they are true, false, partially true, or require further context.
- **Generation of Explanation** produces a human-readable justification for the classification, explaining why the veracity is assigned to the claim. This can include summarizing retrieved evidence, highlighting contradictions, or providing contextual information.

The pipeline is shown in Figure 2.1.

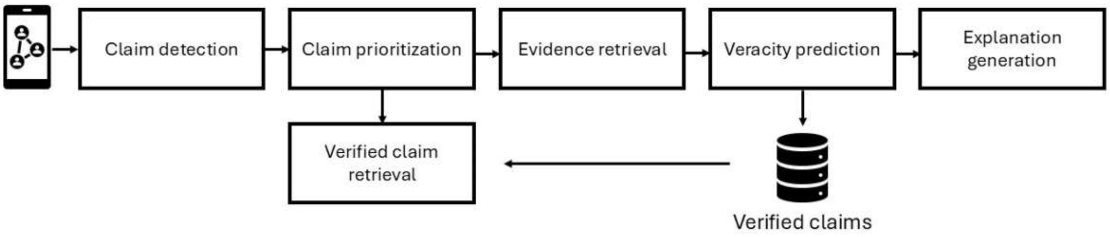

Figure 2.1: Fact-checking pipeline (created by author)

Apart from these five major components, there is an additional component in charge of retrieving previously fact-checked claims. This component identifies pairs of texts with similar claims to be addressed with the same fact check. Grouping similar claims across different languages can help prioritize efforts and expand their reach, combating misinformation more effectively.

This task is referred to as verified claim retrieval [BCEN$^+$20] or claim matching [KGGH21a]. Other names for this task are also used in the literature, such as previously fact-checked claim retrieval (PFCR) [PSM$^+$23] or fact-checked claim detection [SBDSMN20]. For consistency, we will refer to this task as PFCR throughout the thesis.

In this thesis, we are focusing on this component of the automated fact-checking pipeline.

One could argue that the limitation of this task is the assumption of the existence of a corresponding fact-checked article for a given claim. However, it is important to remember that PFCR is an additional component that aims to create a shortcut in the fact-checking pipeline in case the same, perhaps reformulated, claim reappears online after it has been verified. If such a claim does not exist, then, the pipeline would continue with the retrieval of evidence.

CHAPTER 3

Related Work

In this chapter, we give an overview of available fact-checking datasets and retrieval approaches that influenced this work.

3.1 Fact-Checking Datasets

Several datasets have been developed to support the task of PFCR, each different in methodology, language coverage, and data construction.

The CheckThat! 2020 [BEN$^+$20] and CheckThat! 2021 [NDSME$^+$21] datasets contain manually filtered pairs of English and Arabic tweets alongside fact-checks. CheckThat! 2021 extends this by incorporating manually constructed fact-checking data from political debates. Another dataset, CrowdChecked [HCK$^+$22], collects URLs of verified fact-checking articles shared on Twitter and retrieves all tweets referencing those URLs. However, due to its collection process, this dataset includes a high level of noise, requiring additional filtering to refine relevant matches.

Kazemi et al. (2021) [KGGH21b] created a dataset from several million chat messages in multiple languages (English, Bengali, Hindi, Malayalam, Tamil) and around 150,000 fact-checks. They released two annotated datasets — one for claim detection and another for claim similarity. The claim similarity dataset consists of 2,343 pairs of social media messages and fact-checks, categorized on a four-point similarity scale. However, only about 250 pairs were confirmed as positive, as only "Very Similar" pairs were considered valid matches.

Vo and Lee (2019) [VL19] created a dataset that combines images with text, consisting of tweet-reply pairs where fact-checkers responded to original tweets with fact-checked articles from Snopes[1] and PolitiFact[2]. This dataset is exclusively in English. The dataset was later refined in [VL20] by retaining only tweets that contained both text and images.

Table 3.1 provides a comparative overview of these datasets, illustrating their scope in terms of claim volume, fact-checks, and language coverage. While CrowdChecked and Vo and Lee (2019), contain a large number of claims, they are limited to English. CheckThat! 2021 and Kazemi et al. (2021) include multilingual data but cover few languages or contain a limited number of validated claim - fact-check pairs.

Despite these efforts, there was still a significant gap in datasets that support large-scale, multilingual, and crosslingual PFCR. To address this, the MultiClaim dataset [PSM+23] was developed. The new dataset contains 205,751 fact-checks in 39 languages and 28,092 social media posts in 27 languages. With the help of professional fact-checkers, 31,305 pairs of posts and corresponding fact-checks were gathered out of which 4,212 pairs are crosslingual, meaning that the languages of post and fact-check are different. Its linguistic diversity and large dataset size make it a valuable benchmark for evaluating both monolingual and crosslingual PFCR. The dataset is described in detail in Section 4.1.1.

	Input claims	**Fact-checked claims**	**Pairs**	**Languages**
MultiClaim	28,092	205,751	31,305	27/39
CheckThat! 2020	1,197	10,375	2,002	1
CheckThat! 2021	2,928	43,414	3,244	2
CrowdChecked	316,564	10,340	332,660	1
Kazemi et al. (2021)	NA	150,000	258	5
Vo and Lee (2019)	64,110	73,203	73,203	1

Table 3.1: Comparison of PFCR datasets

As shown in Table 3.1, MultiClaim stands out as the dataset with the highest number of verified fact-checked claims and the broadest multilingual coverage. While it has fewer input claims and claim - fact-check pairs than CrowdChecked and Vo and Lee (2019), those datasets are both limited to English. We use this dataset to evaluate our approaches, as it enables a robust assessment of both monolingual and crosslingual retrieval models.

3.2 Preprocessing for Lexical Retrieval

In this section, we highlight the key works that explore the impact of the different preprocessing steps on the effectiveness of lexical retrieval models, such as BM25.

Previous research has evaluated BM25's performance in monolingual settings. In [AOS24], the authors systematically assessed various preprocessing techniques to determine their impact on BM25 in Arabic information retrieval systems. Their findings indicated

[1] snopes.com

[2] politifact.com

that stemming significantly improved performance, normalization had a positive effect, while stop-word removal alone led to a decline in retrieval effectiveness. However, combining certain preprocessing methods, such as normalization with stemming or stop-word removal, yielded notable improvements, highlighting the importance of tailored preprocessing strategies.

In [HKMY20], they reviewed twelve studies on preprocessing for text classification using a bag-of-words representation and extended this research by evaluating all possible preprocessing combinations. This work systematically assessed various preprocessing combinations and found that stop-word removal was the only single technique that consistently improved accuracy across multiple datasets. For some datasets, combining preprocessing methods, such as lowercase conversion and spelling correction, yielded the best results. The study examined datasets primarily in English, but also in Czech and Turkish, reinforcing the importance of dataset-specific preprocessing choices.

Building on prior research, we conclude that enhancing the performance of lexical retrievers and constructing an effective retrieval pipeline requires a systematic evaluation of individual steps and their combinations. In contrast to previous work, our work investigates the impact of various preprocessing strategies on the performance of BM25 in monolingual claim retrieval across multiple languages, including English, German, French, Spanish, Portuguese, Arabic, Malay, and Thai.

3.3 Fact-Checked Claim Retrieval Approaches

In this section, we review key works and methodologies in the field of fact-checking, along with related approaches that contribute to our approach.

3.3.1 Evolution of Fact-Checking Approaches

Fact-checking as a task was first introduced in the "Fact Checking: Task definition and dataset construction" [VR14] paper in 2014. The authors define it as the assessment of the truthfulness of a claim, emphasizing its importance both in journalism, where it is a time-consuming process and for ordinary people to assess the truthfulness of the growing amount of data they consume.

The authors explore two baseline approaches to fact-checking. The first treats it as a classification task, where statements are labeled with verdicts and used to train supervised models. However, they argue this approach is unlikely to succeed, as statements often lack the necessary knowledge and temporal or spatial context for accurate classification. The second approach focuses on matching new statements to those already fact-checked by journalists, reframing the task as a semantic similarity problem. This method uses existing fact-checks to assess the truthfulness of new claims.

In 2012, SemEval introduced a pilot task on semantic textual similarity (STS) [ACDGA12], where word overlap was used as the baseline. Most participating teams improved the baseline by incorporating lemmatization and Part-of-Speech (PoS) tagging. In the following year, approaches had expanded to include parsing, word sense disambiguation, semantic role labeling, time and date resolution, lexical substitution, string similarity, and textual entailment [ACD+13]. Over the next few years, additional techniques such as Latent Semantic Analysis (LSA) [Fol96] and WordNet [Mil95] were introduced, often combined within ensemble models [ABC+14, ABC+15].

From 2016, STS methods began integrating deep learning with traditional NLP pipelines [ABC+16]. Long Short-Term Memory (LSTM) networks and Deep Structured Semantic Models (DSSMs) were combined with feature-engineered models like Random Forest (RF), Gradient Boosting (GB), and XGBoost (XGB), using n-gram overlap features to enhance performance [CDA+17].

Since 2018, PFCR has relied on traditional information retrieval methods like BM25 and embedding-based models. Reranking was used to improve retrieval performance by combining multiple retrieval methods while balancing computational efficiency [SBDSMN20].

Starting in 2020, PFCR became a key task in CLEF's CheckThat! challenge [BEN+20], where participating teams mostly used fine-tuned BERT variants and Support Vector Machines (SVMs) [NME+21a]. By 2022, systems started to employ a two-stage retrieval pipeline: first retrieving documents with BM25 or a similar sparse retrieval method, then reranking them using neural models such as sentence-BERT or T5-based rerankers [BCnEN+20, NMA+22]. Fine-tuned transformer models such as Sentence-BERT, ST5, and GPT-Neo were commonly used, with some systems also exploring data augmentation to further improve retrieval performance.

A more recent approach [VL20] has experimented with using images alongside text to enhance retrieval accuracy. Some papers improve retrieval by summarizing fact-checking articles or extracting key sentences. Bhatnagar et al. (2022) [BKC22] explored summarization of long fact-checks, while Sheng et al. (2021) [SCZ+21] focused on selecting the most relevant evidence sentences.

3.3.2 Our Approach: Hybrid Retrieval and Reranking for PFCR

Building upon the mentioned fact-checking approaches, we propose a hybrid retrieval framework that integrates lexical and semantic models with an ensembler-based retrieval-reranking pipeline.

Unlike prior studies that rely on either lexical or dense retrievers followed by a single reranker, our approach introduces:

- Retriever Ensembling – A combination of multiple retrievers to enhance recall and retrieval robustness.
- Reranker Ensembling – A set of rerankers that refine retrieved results by leveraging their strengths to enhance precision.

By integrating retriever and reranker ensembling, our framework balances lexical and semantic models, in a scalable and robust solution for multilingual and crosslingual PFCR.

3.4 SemEval 2025 Task 7

This thesis is inspired by our participation in SemEval-2025 Task 7: Multilingual and Crosslingual Fact-Checked Claim Retrieval, which addresses the challenge of efficiently identifying previously fact-checked claims across multiple languages.

The task is divided into two subtasks:

- Monolingual retrieval, where fact-checks are retrieved in the same language as the given claim.
- Crosslingual retrieval, where fact-checks may be in a different language than the claim.

We participated in both monolingual and crosslingual subtasks. We discuss our shared task results in Section 7.4.

Systems were evaluated using S@k metric, which measures the proportion of claims for which at least one relevant fact-check appears within the top-k retrieved results.

The use of any external data apart from the provided dataset to prepare the submission was not allowed in the shared task. However, using pre-trained language models and data augmentation of the dataset was allowed.

This thesis builds upon the research done within the shared task, further exploring hybrid retrieval strategies, ensembling, and reranking techniques to enhance fact-check retrieval efficiency and robustness.

CHAPTER 4

Experiment Setup

To address our research questions, we designed and evaluated a series of experiments, testing different configurations of retrieval models, rerankers, and ensemble strategies. Our goal was to determine the most effective approach for multilingual and crosslingual previously fact-checked claim retrieval. This section details the used datasets, evaluation measures for each research question and the implementation details.

4.1 Datasets

4.1.1 MultiClaim Dataset

The MultiClaim dataset [PSM+23] is a large-scale multilingual dataset designed for PFCR, addressing the need for multilingual and crosslingual fact-checking. It was created to overcome the lack of datasets that extend beyond English and support retrieval in multiple languages.

MultiClaim contains fact-checks and social media posts for monolingual and crosslingual retrieval. The dataset includes 205,751 fact-checks in 39 languages and 28,092 social media posts in 27 languages.

Fact-checks were collected from the Google Fact Check Explorer[1], with additional inputs from major fact-checking sources (e.g., Snopes[2]). In total, fact-checks were collected from 142 organisations, where each entry includes the claim, title, publication date, and URL. While full article texts are not included, each fact-check provides a one-sentence summary of the information being verified. Social media posts were gathered from Facebook[3],

[1] https://toolbox.google.com/factcheck/explorer

[2] https://www.snopes.com/

[3] https://www.facebook.com/

Instagram[4], and former Twitter[5], resulting in a total of 28,092 posts across 27 languages. The dataset includes 31,305 aligned post–fact-check pairs, where each post is linked to at least one relevant fact-check. 26,774 pairs are monolingual (post and fact-check in the same language), while 4,212 are crosslingual (post and fact-check in different languages) [PSM+23].

Languages and Distribution

MultiClaim provides monolingual data for ten languages: English, German, French, Spanish, Portuguese, Arabic, Malay, Thai, Polish, and Turkish, with the last two included only in the test set. While English accounts for the largest portion of the dataset with 85,814 fact-checks, the remaining languages contribute significantly to the multilingual diversity of the collection. Specifically, Portuguese includes 21,569 fact-checks, followed by Arabic (14,201), Spanish (14,082), Malay (8,424), Turkish (6,676), German (4,996), Polish (4,430), and French (4,355) [PSM+23]. Thai, while included in the monolingual evaluation, has a smaller volume of data with only 382 fact-checks.

Despite the dominance of English, MultiClaim ensures broad representation across both high- and low-resource languages, supporting robust multilingual and crosslingual retrieval research, making it one of the most comprehensive resources for fact-checking in a multilingual context.

Dataset Structure

For each fact-check, the dataset provides:

- **ID**: A unique identification of the fact-check.
- **Claim**: The summary statement being fact-checked, its English translation, identified languages, and their respective percentages.
- **Title**: The original title, English translation of the title, identified languages, and their respective percentages.
- **Publication Date & Source (instances)**: Metadata indicating when and where the fact-check was published, including timestamps and URLs.

An example of a fact-check entry from the dataset is provided in Table 4.1.

For each social media post, the dataset includes:

- **ID**: A unique identification of the post.

[4] https://www.instagram.com/
[5] https://www.twitter.com/

fact_check_id	18
claim	La filarmónica de París toca el Bolero de Ravel en una plataforma sobre el Sena por el levantamiento de la cuarentena, The Paris Philharmonic plays Ravel's Bolero on a platform over the Seine for the lifting of the quarantine, ('spa', 1.0)
instances	1594243500.0, https://factual.afp.com/el-video-de-una-orquesta-tocando-el-bolero-de-ravel-sobre-el-sena-en-paris-es-de-2017#367fc73ea7c0c5812887632bc66ff2f5
title	El video de una orquesta tocando el Bolero de Ravel sobre el Sena, en París, es de 2017, The video of an orchestra playing Ravel's Bolero on the Seine, in Paris, is from 2017, ('eng', 1.0)

Table 4.1: An example of a fact-check's data structure from the MultiClaim dataset

- **Text content**: The text written by the user, the English translation of the text, identified languages, and their respective percentages.

- **OCR** (optical character recognition): transcripts of images attached to the post (if any), their English translations, identified languages, and their respective percentages.

- **Publication Date & Source (instances)**: Metadata indicating the timestamp and social media platform where the post was published.

- **Verdict**: The conclusion regarding the veracity of the post, where the possible verdicts are:

  - False information: The claim made in the post is incorrect based on the verified sources.
  - Partly false information: The post contains a mix of accurate and inaccurate details, potentially misleading the readers.
  - Missing context: The post is not necessarily false, but it lacks the context for proper interpretation.
  - Altered photo/video: The post includes manipulated or edited visual content that misrepresents the truth.

- **Publication Date & Source (instances)**: Metadata indicating when and where it was published, including timestamps and URLs.

post_id	3057
instances	1654553532.0, fb
ocr	[]
verdicts	False Information
text	#2022 - 9€ - Kulturreisen? Die #Sylter waren jedenfalls begeistert von den Touristen! :) :), #2022 - 9€ - cultural trips? In any case, the #Sylter were enthusiastic about the tourists! :) :), ('deu', 1.0)

Table 4.2: An example of a post's data structure from the MultiClaim dataset

An example of a post entry from the dataset is provided in Table 4.2.

Mappings between fact-checks and posts are provided in a separate file, linking each post to one or more fact-checks that verify or refute the claim.

An example of a post - fact-check mapping entry from the dataset is provided in Table 4.3.

fact_check_id	968
post_id	27280

Table 4.3: An example of a post - fact-check mapping from the MultiClaim dataset

Dataset Preprocessing

The dataset preprocessing include the following steps:

- **Removing Noisy Fact-checks and Posts**: Fact-checks that had no claim or where the claim was shorter than 10 characters were removed, as well as the texts or OCR transcripts that were shorter than 25 characters or had more than 50% non-alphabetical characters.
- **Translation**: Fact-checks and posts were translated into English to enable translation-based retrieval alongside multilingual approaches.
- **Deduplication**: Redundant fact-checks and posts were filtered out.

The dataset is divided into three stages: training, development, and testing. Each stage includes three datasets: fact-checks, social media posts, and their corresponding mappings.

4.1.2 CheckThat! 2021 Dataset

To further evaluate our approach, we evaluate the pipeline on the CheckThat! 2021 dataset for subtask 2A. It consists of two main components: verified claims (referred to as vclaims) and social media posts or tweets (referred to as iclaims). The dataset includes 2,928 claims, 43,414 fact-checks and 3,244 mappings between them in English and Arabic. This dataset contains only monolingual mapping and will be used during the evaluation of the monolingual subtask.

Dataset Structure

Each vclaim represents a previously fact-checked claim and is used for verifying new input claims from social media. Verified claims are provided in the following format:

- **vclaim_id**: A unique identifier for the verified claim.
- **vclaim**: The text of the verified claim.
- **title**: The title of the article providing justification for the claim's veracity label.

An example of a fact-check claim entry from the dataset is provided in Table 4.4.

vclaim_id	vclaim-sno-mom-tear-gas-photo
vclaim	A photograph of a migrant mother and her children fleeing a tear gas attack near a border crossing was staged.
title	Was the 'Illegal Alien Mom with Kids' Photograph Staged?

Table 4.4: An example of a fact-check claim data structure from the CheckThat! 2021 dataset

Input claims represent the social media content that needs to be verified. In subtask 2A, input claims are tweets, provided in a tab-separated file with the following columns:

- **tweet_id** or **iclaim_id**: A unique identifier for each tweet.
- **tweet_text** or **iclaim**: The content of the tweet or input claim.

An example of a post entry from the dataset is provided in Table 4.5.

iclaim_id	tweet-sno-298
iclaim	#FakeNews Media all share same photo of "women & kids gassed". Perpetrators of invasion instantly become victims in new #FakeNews narrative. Congratulations, you've been #Hoaxed! pic.twitter.com/ISkExrlc2T — #WalkAway Mexican J.Lo. (@jetrotter) November 26, 2018

Table 4.5: An example of a post's data structure from the CheckThat! 2021 dataset

Mappings between input claims and verified claims are provided in a separate tab-separated file. Each row has the following structure:

iclaim_id 0 vclaim_id 1

- **iclaim_id**: Identifier of the input claim (tweet).
- **0**: A constant field to comply with the requested (TREC) format.
- **vclaim_id**: Identifier of the verified claim that supports or refutes the tweet.
- **relevance**: Indicates a relevant (positive) match between the input claim and the verified claim. Only relevant (relevance=1) mappings were listed in the file. All unlisted pairs are considered non-relevant by default.

During the preprocessing, we remove the "0" and "relevance" columns to match the format of mappings from the MultiClaim dataset.

An example of a post - fact-check mapping entry from the dataset is provided in Table 4.6.

iclaim_id	tweet-sno-298
vclaim_id	vclaim-sno-mom-tear-gas-photo

Table 4.6: An example of a post - fact-check mapping from the CheckThat! 2021 dataset

4.2 Evaluation Measures

To assess the effectiveness of retrieval models, we use standard retrieval evaluation metrics that measure how well the retrieved documents (fact-checks) align with the relevant ground-truth references. The following metrics are used in our evaluation:

4.2.1 Success-at-k (S@k)

Success-at-k (S@k) [Voo05] evaluates whether at least one relevant document appears within the top k retrieved results. It is a binary metric, assigning a value of 1 if any relevant fact-check is found within the top k, and 0 otherwise. The final score is the average over all the queries:

$$S@k = \frac{1}{N}\sum_{i=1}^{N} \nVdash(\exists d \in D_i^k : d \in R_i)$$

where:

- N is the total number of queries,
- D_i^k is the set of top-k retrieved documents for query i,
- R_i is the set of relevant documents for query i,

- $\nVdash(\cdot)$ is an indicator function that returns 1 if at least one relevant document is found within the top-k results, and 0 otherwise.

Throughout this research, we base the evaluation on the S@k metric, as it is practical in real-world contexts, such as PFCR. S@10 provides a binary indication of whether users are likely to find a relevant document (fact-check) within the top 10 results, without having to scan through many retrieved results. A retrieval is considered successful if any relevant document appears within the top 10 results. This is crucial when achieving a perfect ranking order is challenging, and the priority is simply retrieving relevant documents. S@10 is an important metric for determining whether relevant results are accessible, however, the addition of rank-sensitive metrics provides deeper insights into retrieval effectiveness.

4.2.2 Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) measures the rank position of the first relevant document in the retrieved list. If the first relevant document appears at rank r_i, the reciprocal rank is $\frac{1}{r_i}$, and MRR is the average over all queries:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i}$$

where:

- N is the total number of queries,
- r_i is the rank of the first relevant document for query i (if no relevant document is retrieved, $\frac{1}{r_i}$ is set to 0).

Higher MRR values indicate that relevant fact-checks tend to appear earlier in the retrieved list.

4.2.3 Precision-at-k (P@k)

Precision-at-k (P@k) measures the proportion of retrieved documents within the top k results that are actually relevant. It evaluates the system's ability to return relevant fact-checks while minimizing irrelevant ones:

$$P@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i^k \cap R_i|}{|D_i^k|}$$

where:

- N is the total number of queries,
- D_i^k is the set of top-k retrieved documents for query i,
- R_i is the set of relevant documents for query i.

A higher P@k score indicates that a greater proportion of the retrieved fact-checks are relevant.

4.2.4 Recall-at-k (R@k)

Recall-at-k (R@k) measures the fraction of relevant documents that appear within the top k retrieved results. It evaluates how comprehensive the retrieval system is in retrieving relevant documents:

$$R@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|D_i^k \cap R_i|}{|R_i|}$$

where:

- N is the total number of queries,
- D_i^k is the set of top-k retrieved documents for query i,
- R_i is the set of relevant documents for query i.

A higher R@k score indicates the retrieval system captures more relevant fact-checks within the top-k results.

4.2.5 Comparison of Measures

To assess the effectiveness of retrieval models in the PFCR task, we use evaluation measures, where each provides a different perspective on retrieval performance:

- S@k is important in PFCR, where retrieving at least one verified fact-check that matches the claim can be sufficient for fact-checking. However, it does not consider the ranking quality within the top k, which is not a problem when k is reasonably low (e.g. k=10). However, S@k also overlooks cases where multiple relevant fact-checks exist, making it a less precise measure of overall retrieval effectiveness. Despite this, it remains useful in practical fact-checking scenarios where finding any relevant fact-check is a priority.
- MRR emphasizes the quality of the ranking by rewarding models that rank relevant fact-checks higher. However, it only considers the first relevant fact-check, ignoring cases where multiple fact-checks may provide valuable information.

- P@k focuses on retrieval accuracy, ensuring the retrieved results are relevant. However, since the number of relevant fact-checks is often lower than the number of retrieved ones, P@k values tend to be low, making it less informative compared to other metrics. Additionally, P@k does not account for missed relevant claims, limiting its effectiveness in overall retrieval performance evaluation.

- R@k focuses on retrieval completeness, ensuring that as many relevant fact-checks as possible are retrieved. In PFCR, higher recall ensures that more relevant fact-checks are retrieved, helping to cover different aspects of the claim. However, R@k only considers whether relevant fact-checks appear within the top k results but does not account for their ranking order. A system could have high recall but still rank irrelevant fact-checks higher than relevant ones, reducing practical usefulness.

4.2.6 Evaluation Measures per RQs

To answer RQ1 (impact of preprocessing and retrieval errors), we evaluate performance using:

- P@10
- R@10
- S@10
- MRR

For RQ2 (efficiency comparison between multilingual and translation-based retrieval), we assess:

- S@10 to compare retrieval effectiveness.
- Execution time (seconds per query) to measure execution time.
- Number of model parameters indicating model size.
- Memory usage (MB) capturing RAM/VRAM consumption.

To address RQ3 (effectiveness of retriever-reranker configurations), we use:

- S@10 to evaluate the ranking quality across different retrieval pipelines.

4.3 Implementation Details

As a starting point for model selection and benchmarking, we referred to the Massive Text Embedding Benchmark (MTEB)[6], which provides an evaluation of text embedding models across diverse retrieval and ranking tasks.

We implemented the following retrieval models:

- BM25: A traditional sparse retrieval baseline using the BM25Okapi implementation from the rank-bm25 Python library[7]. This served as a reference point for comparing performance against neural models.
- Bi-encoder: A dense retrieval model where queries and documents are encoded independently. We used AutoModel from Hugging Face's transformers library[8] to load pre-trained models.
- Cross-encoder: A reranking model where the query-document pair is jointly encoded to compute relevance scores. This was implemented using the SentenceTransformer library[9].

4.4 Computation and Efficiency

Given the large size of the retriever and reranker models, we enabled mixed precision during inference to improve computational efficiency, reduce GPU memory consumption, and accelerate processing.

All the experiments were conducted on NVIDIA GeForce GTX 1080 Ti and NVIDIA TITAN RTX GPUs.

[6] https://huggingface.co/spaces/mteb/leaderboard
[7] https://pypi.org/project/rank-bm25/
[8] https://huggingface.co/transformers
[9] https://sbert.net/

CHAPTER 5

System Overview

This section presents the system architecture shown in Figure 5.1, outlining the key components of the pipeline:

(1) Data preprocessing module (Section 5.1),

(2) Retrieval-ensemble module (Section 5.2),

(3) Reranking-ensemble module (Section 5.3),

(4) Evaluation module (Section 5.4).

Given a collection of fact-checks, our system retrieves the top k most relevant fact-checks for a given claim. In this context, relevant fact-checks are the ones addressing the same statement as the given claim.

5.1 Data Preprocessing Module

The data preprocessing module prepares claims and fact-checks for downstream retrieval and reranking. For lexical models, preprocessing focuses on cleaning and normalizing the text, while for semantic models, the text is enriched with additional contextual descriptions.

5.1.1 Preprocessing for Lexical Retrieval

This component applies text cleaning and normalization to improve the effectiveness of lexical retrieval models. We systematically evaluate translation, stop-word removal, stemming, lemmatization, and spell correction to determine their impact on retrieval accuracy.

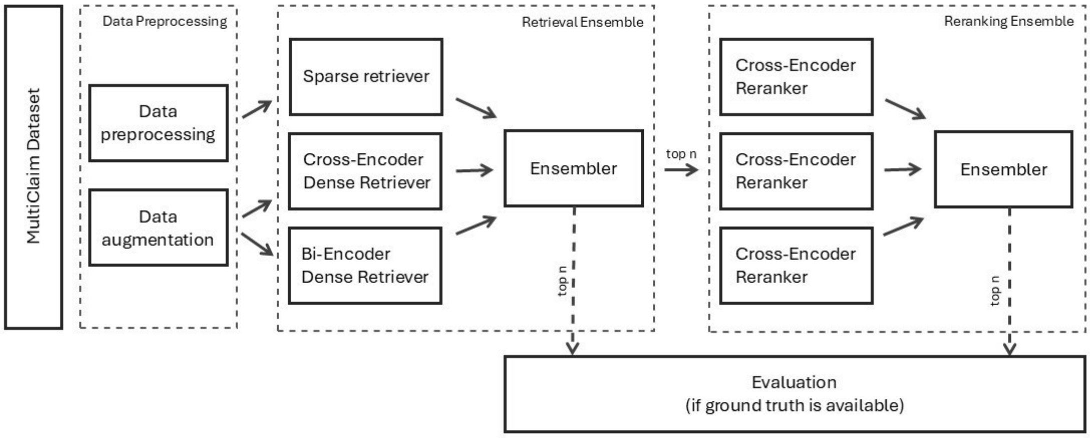

Figure 5.1: System architecture overview

In addition to optimizing retrieval performance, we analyze how preprocessing affects retrieval errors, such as false positives and false negatives. Our evaluation spans multiple languages (English, German, French, Spanish, Portuguese, Arabic, Malay, and Thai), offering insights into effective preprocessing strategies for multilingual fact-checking.

Baseline Evaluation

As previously noted, BM25 is widely adopted as a strong baseline in modern information retrieval research, making it a natural choice for our lexical retriever. We begin by evaluating BM25's retrieval performance using S@10 scores on both the original-language text and its English-translated version without applying any preprocessing. This comparison aims to assess whether translations can mitigate the linguistic noise often present in social media content, such as the use of multiple languages within a post, inconsistent spelling, and hashtags, which can hinder lexical retrieval.

The original text achieved an average S@10 of 0.5171, while the translated text performed better, reaching 0.5569. The difference between the performance of the English and the original version suggests that the translation can enhance retrieval performance due to improved linguistic consistency and alignment with fact-checking sources.

This initial analysis provides a baseline to measure the impact of individual preprocessing steps. We continue with the pipeline development based on the better-performing English baseline.

Preprocessing Pipeline Development

Building on the baseline results, we iteratively refined the preprocessing pipeline to reduce noise, improve normalization, and enhance retrieval robustness. Table 5.2 compares S@10 scores for different preprocessing steps using the English-translated text. We first assess the impact of text cleaning and lemmatization, as these steps improve text consistency and reduce vocabulary sparsity. The cleaned and lemmatized text then serves as the base for evaluating additional preprocessing techniques, including emoji handling and date normalization. This stepwise approach allows us to isolate the effect of each step and measure its influence on retrieval performance. Finally, we create the full preprocessing pipeline, combining all steps that demonstrated improvements.

Code	**Preprocessing Step**
C	Text Cleaning
L	Lemmatization
S	Stemming
TE	Translated Emoji
RE	Removed Emoji
D	Normalized Dates
N	Removed Digits
SC	Spelling Correction

Table 5.1: Legend of preprocessing components used in Table 5.2 and Table 5.3. The columns of the tables correspond to a cumulative combination of these steps.

The pipeline consists of:

1. Text Cleaning & Normalization
   - Case normalization, whitespace standardization, removal of URLs, HTML entities, stop-words and punctuation, Unicode normalization and reducing multiple consecutive characters minimized the noise and contributed to lexical matching.
2. Lemmatization
   - By converting words to their base lemma forms, we mitigate issues related to inflected word variations that would otherwise be treated as separate terms, leading to retrieval mismatches.
   - For English, we used WordNetLemmatizer[1], while for other languages, we employed Simplemma[2].
3. Emoji Handling

[1] https://www.nltk.org/_modules/nltk/stem/wordnet.html

[2] https://github.com/adbar/simplemma

Language	None	C	C+L	C+L+RE	C+L+D	Full (C+L+RE+D)
Arabic (ara)	0.7076	0.7091	0.7103	0.7212	0.7633	0.8388
German (deu)	0.4403	0.6300	0.6775	0.6892	0.6136	0.7494
English (eng)	0.4128	0.6125	0.6288	0.6300	0.6379	0.6906
French (fra)	0.5883	0.6464	0.6521	0.6862	0.7067	0.8152
Malay (msa)	0.5185	0.6725	0.6806	0.6862	0.6959	0.7173
Portuguese (por)	0.6070	0.7925	0.8086	0.8054	0.7312	0.8284
Spanish (spa)	0.5585	0.7974	0.8164	0.8222	0.7325	0.8177
Thai (tha)	0.2738	0.4351	0.7986	0.8021	0.8479	0.9163
Average	0.5133	0.6619	0.7216	0.7328	0.7161	**0.7967**

Table 5.2: S@10 scores across languages using BM25 on English-translated texts with varying preprocessing steps. Abbreviations refer to cumulative combinations of preprocessing components as defined in the legend.

- Given that BM25 does not effectively handle emojis, we experimented with two strategies:
  - Transcribing emojis into the original language.
  - Removing emojis entirely.
- The results in Table 5.2 show that removing emojis improved retrieval effectiveness, as emojis often introduced noise.

4. Date Normalization

- Since dates and numbers contribute little in lexical retrieval methods, we tested:
  - Normalizing dates to a standard format (month day, year e.g. February 12, 2025) .
  - Removing digits entirely to reduce sparsity.
- Results in Table 5.2 show that date normalization improved retrieval performance in comparison to the (C+L) version, in Arabic, English, French, Malay

Language	C+S	C+L+TE	C+L+N	Full+SC
Arabic (ara)	0.6272	0.7001	0.7633	0.8148
German (deu)	0.4943	0.6815	0.5925	0.7073
English (eng)	0.4755	0.6292	0.5574	0.6446
French (fra)	0.5919	0.6939	0.7049	0.7955
Malay (msa)	0.5305	0.6842	0.6589	0.6920
Portuguese (por)	0.6699	0.7913	0.7271	0.7958
Spanish (spa)	0.5730	0.8136	0.7304	0.7970
Thai (tha)	0.4688	0.7697	0.8479	0.8935
Average	0.5539	0.7204	0.6978	0.7676

Table 5.3: S@10 scores across languages using BM25 on English-translated texts with preprocessing steps that were evaluated but not included in the final pipeline. Abbreviations refer to cumulative combinations of preprocessing components as defined in the legend.

and Thai by making temporal references more comparable. Removing the digits decreased the performance of the retriever, as shown in Table 5.3.

5. Final Preprocessing Pipeline

   - After evaluating individual steps, we combined the most effective ones into a final preprocessing pipeline. We evaluated the pipeline with both the original and English-translated texts, with a stronger focus on the translated text as it showed stronger performance.
   - The final pipeline includes:
     - Translation
     - Text cleaning
     - Emoji removal
     - Date normalization
     - Lemmatization

In addition to the steps mentioned above, we explored several other promising preprocessing techniques. While they did not lead to performance improvements, we include them here for completeness. The results refer to Table 5.3.

- Stemming
  - We evaluated stemming, which reduces inflected words to their root form by removing prefixes and suffixes.
  - Stemming was implemented using NLTK's SnowballStemmer[3] and malaya[4] for Malay.
  - Compared to lemmatization, stemming led to a decrease in retrieval performance. This decline in performance may be due to overly aggressive reductions, producing words that are not actual dictionary terms, causing reduced lexical matching as some stemmed words no longer align well with fact-checked claims, negatively impacting retrieval effectiveness.
- Digit Removal
  - Removing digits decreased the performance in most languages, suggesting that numbers alone were not major sources of retrieval errors.
- Impact of Spelling Correction
  - We tested adding spelling correction before lemmatising on the English-translated version, as an additional preprocessing step.
  - Spelling correction was implemented using the TextBlob library[5], which applies a probabilistic word-level correction based on term frequency in large English corpora. It operates on each word independently without contextual understanding, therefore, the suggested replacements may not always be semantically accurate.
  - The performance decreased possibly due to:
    - Overcorrection of noisy text, causing unintended semantic shifts.
    - Word-level correction implementation, causing semantic shifts.
    - Modification of fact-check-related terms, making them harder to match.

Findings and Implications

We compare the performance of BM25 with and without preprocessing based on P@k, R@k, S@k, and MRR metrics, and show the results in Table 5.4.

Our experiments demonstrate that careful preprocessing can significantly enhance BM25's fact-check retrieval performance. The average S@10 score increased from 0.5171 to 0.7229 for the original-language training data and from 0.5569 to 0.7967 for the English-translated version after applying the full preprocessing pipeline. The results demonstrate the importance of task-specific preprocessing for lexical retrieval models in multilingual

[3] https://www.nltk.org/api/nltk.stem.SnowballStemmer.html

[4] https://malaya.readthedocs.io/en/stable/_modules/malaya/stem.html

[5] https://textblob.readthedocs.io/en/dev/

settings. Translation, in particular, helps reduce linguistic noise commonly found in social media content, such as the use of multiple languages within a single post, slang, irregular capitalization, non-standard abbreviations (e.g., "lol" for "laugh out loud"), and hashtags (e.g., #Throwback). Addressing these issues through translation and preprocessing allows for more accurate and reliable retrieval, overcoming barriers that typically complicate lexical searches.

Additionally, we compare our S@10 results with those reported in the MultiClaim dataset paper [PSM+23]. While their reported scores vary, they indicate S@10 values of 0.61 and 0.78 for the English-translated version, and 0.48 and 0.62 for the original-language version (see Tables 2 and 9 in their paper). Our approach achieves 0.7229 on the original-language data and 0.7967 on the English-translated data, both of which surpass the reported baselines, highlighting the effectiveness of our preprocessing and retrieval strategy.

Language	P@10	R@10	S@10	MRR
No Preprocessing				
Arabic (ara)	0.0630	0.6243	0.7376	0.4924
German (deu)	0.0340	0.2646	0.4403	0.2080
English (eng)	0.0274	0.2276	0.4128	0.1526
French (fra)	0.0404	0.3910	0.5883	0.3236
Malay (msa)	0.0244	0.2288	0.5185	0.1328
Portuguese (por)	0.0302	0.2474	0.6070	0.1799
Spanish (spa)	0.0285	0.2578	0.5585	0.1989
Thai (tha)	0.0422	0.4215	0.2738	0.3259
Average	**0.0363**	**0.3339**	**0.5171**	**0.2518**
Full Preprocessing				
Arabic (ara)	0.0846	0.8388	0.8388	0.6495
German (deu)	0.0895	0.7264	0.7494	0.5165
English (eng)	0.0807	0.6665	0.6906	0.4320
French (fra)	0.0843	0.8120	0.8152	0.6931
Malay (msa)	0.0764	0.7057	0.7173	0.4051
Portuguese (por)	0.1009	0.8035	0.8284	0.5660
Spanish (spa)	0.0891	0.8104	0.8177	0.6478
Thai (tha)	0.0916	0.9163	0.9163	0.7821
Average	**0.0871**	**0.7855**	**0.7967**	**0.5865**
MultiClaim Paper	NA	NA	0.61	0.78

Table 5.4: Evaluation metrics comparison for BM25 monolingual retrieval using English-translated text

Addressing RQ1: The preprocessing pipeline that most effectively enhanced the performance of BM25 for monolingual claim retrieval consists of translation, text cleaning (URLs, HTML entities, stop-words and punctuation removal, Unicode and case normalization, reducing multiple consecutive characters, whitespace standardization), emoji removal, date normalization and lemmatization.

Based on the data in Table 5.4, we observe a reduction in both false negatives (through a higher R@10) and false positives (in increased P@10) across all languages. Without any preprocessing, R@10 was 0.3339 while after the preprocessing it increased to 0.7855, proving that less relevant fact-checks were missed. R@10 is a very indicative measure as we aim to minimize the number of missed fact-checks. P@10 increased from 0.0363 to 0.0871, indicating that fewer irrelevant results (false positives) are included in the top-ranked documents. However, in PFCR, the number of false positives is not as indicative as the number of false negatives, as it measures the number of accurate predictions in the top 10 retrieved documents, while there are only one or two correct fact-checks per claim, therefore, the score is always low. MRR and S@10 increase significantly after preprocessing, as correct fact checks appear earlier in the ranked list and more often within the top 10 results, providing an evaluation measure most relevant for real-world PFCR settings.

5.1.2 Contextual Enrichment for Semantic Models

Unlike lexical retrieval, semantic models process natural language as-is. However, enriching claims and fact-checks with structured descriptions serves as context, helping the model better understand the underlying relationship between a claim and a fact-check.

We optimized the input formatting, finding that explicitly defining fact-check claims and posts improved retrieval accuracy, enhancing the model's ability to understand contextual relationships between elements. The original text of fact-checks and posts remained unchanged. We evaluated the following three formats:

1. Concatinating attributes without descriptions:

   - Posts:

     claim title claim

   - Fact Checks:

     ocr text instances verdict

2. Prefixing attributes:

   - Posts:

     claim: *claim* title: *title* instances: *claim*

   - Fact Checks:

     ocr: *ocr* text: *text* instances: *instances*

3. Contextual guidance:

- Posts:

  The following claim was posted: *OCR+text*, posted on *date*. The content is labeled as: *verdict*.

- Fact Checks:

  This is a fact-checked claim: *claim*, with the title *title* posted on *date*.

Table 5.5 compares the average S@100 of E5 and BGE models with the three input formats. Simpler formats, such as concatenating attributes without descriptions and prefixing attributes with their names only, both led to lower retrieval accuracy, likely due to the lack of contextual guidance.

Model/Setting	**No Descriptions**	**Prefixing**	**Contextual Guidance**
E5	0.9440	0.9555	0.9586
BGE	0.9471	0.9531	0.9537

Table 5.5: Average S@100 scores for E5 and BGE models using the different input formats on the English-translated version

5.2 Retrieval-Ensemble Module

The retrieval-ensemble module retrieves the top k relevant fact-checks for a given claim as an ensemble of lexical and semantic retrievers. We select a set of retrievers that complement each other's strengths which are then leveraged by the ensembler.

5.2.1 Retrievers

Our system utilizes a combination of lexical and dense retrieval models to retrieve relevant fact-checks efficiently and accurately. We employ:

- Lexical (Sparse) Retriever: BM25 term-based model that retrieves fact-checks based on word overlap with the claim.

- Bi-Encoder (Dense) Retriever: A neural model that independently encodes claims and fact-checks into a shared embedding space, allowing retrieval based on semantic similarity.

- Cross-Encoder (Dense) Retriever: A more complex model that jointly encodes claims and fact-checks, capturing deeper contextual relationships for improved retrieval.

Selection of Dense Retrievers

We selected the following pre-trained models as encoders in the retrieval stage based on their zero-shot S@k performance on the training set:

- $multilingual - E5 - large - instruct$ (E5) as a cross-encoder,
- $bge - multilingual - gemma2$ (BGE) as a bi-encoder

Table 5.6 compares all the evaluated models. Experiments were conducted using both the original-language texts and their English translations, with the original versions having consistently better results. The best-performing model, E5, achieves an average S@100 score of 0.9330 while the second-best model, BGE, achieves a similar performance with an S@100 score of 0.9293. We prioritize E5 and BGE for further tuning and evaluation.

For evaluation of the retriever models, we report S@100 scores ($k = 100$) rather than S@10, as the retrieval stage is responsible for producing a larger candidate set of fact-checks, which are then refined. Thus, the performance on a broader set is more indicative of the retrieval effectiveness at this stage.

Model	**Avg S@100**	**Model size (params)**
Multilingual-E5-Large-Instruct	0.9330	560M
BGE-Multilingual-Gemma2	0.9293	9.24B
NV-Embed-v2	0.9201	7.85B
GTR-T5-XL	0.9019	1.24B
BGE-M3	0.8731	568M
MiniLM-L6-v2	0.7947	22.7M
stella_en_1.5B_v5	0.5288	1.54B
XLM-RoBERTa-Large	0.1467	561M

Table 5.6: Retriever model comparison (S@100) without any preprocessing, using original language

5.2.2 Ensembler

The retriever ensembler aggregates retrieval results from multiple models. It is designed to balance the strengths of sparse and dense retrieval, ensuring that the lexical model provides high-precision results for explicit claim matches while semantic models capture implicit relationships and conceptual similarities.

The addition of an ensembler enhances retrieval robustness, particularly in multilingual retrieval, where different retrieval models may perform better depending on the language and dataset characteristics.

Aggregation Function

To effectively combine the results of multiple retrievers, we explored various aggregation strategies, such as majority voting, exponential decay weighting, and RRF, and evaluated them in both monolingual and crosslingual settings. Among these methods, RRF delivered the highest performance and was selected for our pipeline. As shown in Table 5.7, the ensembler using RRF achieved the best average S@100 score in both settings, highlighting its effectiveness in prioritizing relevant fact-checks. Its superior performance comes from its ability to effectively prioritize highly ranked fact-checks from diverse retrieval models while reducing the impact of lower-ranked ones.

Setting	**Aggregation Function**	**S@100**
Monolingual	Majority Voting	0.9649
Monolingual	Exponential Decay Weighting	0.9674
Monolingual	RRF	0.9720
Crosslingual	Majority Voting	0.8813
Crosslingual	Exponential Decay Weighting	0.8897
Crosslingual	RRF	0.8967

Table 5.7: Comparison of ensembler aggregation functions for monolingual and crosslingual settings

Retrieval Set Size

To determine the optimal number of fact-checks retrieved per claim in the first-stage retrieval, we evaluated the ensembler's S@k performance across different retrieval set sizes (k = 50, 100, 200, 300, 400) in the monolingual setting. We observed that increasing k initially improves the ensembler's performance until it stabilizes. This indicates that a larger retrieval set enhances performance up to a certain threshold, beyond which additional increases yield no further gains.

The robustness of the ensembler at higher k values can be attributed to the RRF aggregation method, which ensures that highly ranked fact-checks from any model remain prioritized, while lower-ranked ones have minimal impact.

Based on these findings, we set k=300 for our monolingual experiments, as increasing the retrieval set size beyond this point does not improve performance. Table 5.8 illustrates the ensembler's performance across different retrieval set sizes, showing that S@k plateaus at k=300.

Following the monolingual retrieval experiments, we assessed the impact of retrieval set size on the ensembler's S@100 performance in the crosslingual setting. As shown in Table 5.8, increasing k improved performance, but the gains diminished beyond k = 300, where S@100 stabilized (0.8967 at k=300 vs. 0.8970 at k=400).

This plateau confirms that while expanding the retrieval set increases the likelihood of retrieving relevant fact-checks, the marginal benefits become insignificant beyond a certain threshold. The ensembler's stability at larger k values further supports the effectiveness of the RRF aggregation method in maintaining ranking robustness.

Based on these findings, we set k=300 for our crosslingual retrieval experiments, as it maximizes performance without causing unnecessary computational overhead.

Setting	**Retrieval Set Size (k)**	**S@k**
Monolingual	50	0.9693
Monolingual	100	0.9679
Monolingual	200	0.9715
Monolingual	300	0.9720
Monolingual	400	0.9720
Crosslingual	50	0.8914
Crosslingual	100	0.8947
Crosslingual	200	0.8954
Crosslingual	300	0.8967
Crosslingual	400	0.8970

Table 5.8: Comparison of ensembler performance (S@k) across different retrieval set sizes (k) in monolingual and crosslingual settings

Ensemble Weighting

We explored ensemble weighting strategies to optimize retrieval performance. The results in Table 5.9 show that reducing BM25's weight while maintaining higher weights for semantic models in the monolingual setting leads to improved retrieval effectiveness. Specifically, assigning a weight of 0.5 to BM25 and 1.0 to both E5 and BGE achieves the highest average S@100 score of 0.9720, slightly outperforming the equal-weighted (1.0, 1.0, 1.0) ensemble, which scored 0.9718. This suggests that BM25 contributes positively but should not be weighted equally with the semantic models, which are better at capturing the relationships between queries and fact-checks.

Further reducing BM25's weight to 0.25 (0.25 BM25 + 1.0 E5 + 1.0 BGE) resulted in a slight performance drop with S@100 of 0.9711, indicating that BM25 still provides useful lexical matching and should not be completely minimized.

Interestingly, increasing the weight of E5 to 2.0 (0.5 BM25 + 2.0 E5 + 1.0 BGE) or BGE to 2.0 (0.5 BM25 + 1.0 E5 + 2.0 BGE) led to slightly lower performance (0.9701 and 0.9694, respectively), suggesting that the ensemble benefits from the strengths of each semantic model, rather than favouring one over the other.

These results confirm that semantic retrieval methods have a dominant role in improving retrieval effectiveness, while BM25 remains beneficial but requires weighting to achieve optimal performance.

BM25	E5	BGE	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
1.0	1.0	1.0	0.9726	0.9742	0.9497	0.9731	0.9805	0.9632	0.9651	0.9962	0.9718
0.5	1.0	1.0	0.9708	0.9742	0.9509	0.9722	0.9825	0.9632	0.9662	0.9962	**0.9720**
0.25	1.0	1.0	0.9691	0.9742	0.9513	0.9722	0.9805	0.9608	0.9641	0.9962	0.9711
0.5	2.0	1.0	0.9657	0.9649	0.9521	0.9722	0.9825	0.9616	0.9655	0.9962	0.9701
0.5	1.0	2.0	0.9657	0.9719	0.9477	0.9749	0.9766	0.9592	0.9631	0.9962	0.9694

Table 5.9: Ensembler performance comparison on S@100 based on different ensemble weighting schemes in the monolingual setting

We evaluated ensemble weighting strategies for crosslingual retrieval. As shown in Table 5.10, reducing BM25's weight while maintaining higher weights for semantic models improved performance, consistent with monolingual results. The best configuration (S@100 = 0.8967) assigns BM25 a weight of 0.5, with E5 and BGE set to 1.0, outperforming the equal-weighted (1.0, 1.0, 1.0) setup.

Further reducing BM25's weight to 0.25 led to a slight drop in performance (S@100 = 0.8940), confirming its role in capturing lexical matches. As in the monolingual setting, increasing the weight of E5 or BGE does not improve performance, suggesting that a balanced contribution from both semantic models is crucial.

We adopted the same (0.5 BM25 + 1.0 E5 + 1.0 BGE) configuration for crosslingual retrieval, as it offered the best balance between lexical and semantic retrieval.

BM25	E5	BGE	S@100
1.0	1.0	1.0	0.8947
0.5	1.0	1.0	0.8967
0.25	1.0	1.0	0.8940
0.5	2.0	1.0	0.8856
0.5	1.0	2.0	0.8947

Table 5.10: Comparison of ensemble weighting schemes on S@100 performance in the crosslingual setting

5.3 Reranking-Ensemble Module

In this stage, each reranker within the reranking-ensemble module independently reranks the fact-checks retrieved by the retrieval-ensemble module. The outputs of the rerankers are then passed to the ensembler, which aggregates these reranked results to determine the final top 10 fact-checks per claim.

5.3.1 Rerankers

The reranking module refines the initial retrieval results by reordering the top k fact-checks to identify the most relevant subset.

For reranking, we use pre-trained cross-encoder models, which jointly encode both the query and the candidate documents (fact-checks), allowing for more context-aware relevance scoring. These models can capture more fine-grained interactions and, therefore, improve the ranking accuracy. However, cross-encoders are computationally more expensive and feasible when we have a smaller pool of candidates.

Selection of Rerankers

The selection of rerankers was guided by their performance in zero-shot reranking tasks, specifically on the retriever-ensembler's top 100 retrieved fact-checks. To inform our decision, we used the MTEB (Massive Text Embedding Benchmark)[6] as a starting reference point and evaluated the leading reranker models on this benchmark. We then narrowed down our choices based on models that offered strong performance on the MTEB while also considering our GPU resource constraints.

The following three were selected for their combination of high performance on the MTEB leaderboard and their computational feasibility under our setup:

- $gte-Qwen2-7B-instruct$ (QWEN2)
- $NV-Embed-v2$ (NV)
- $GritLM-7B$ (GRITLM)

5.3.2 Ensembler

The ensembler combines the outputs of multiple rerankers to produce a more accurate and reliable final ranking. Combining models with different strengths mitigates individual biases and enhances the overall ranking robustness.

Depending on the model's fine-tuning setting, some rerankers are optimized for multilingual retrieval, while others perform better in English settings. Certain models excel at detecting negation, identifying inconsistencies or handling specific topics. We use the ensembler to integrate these complementary capabilities and improve the system's overall effectiveness.

Aggregation Function

As with the retrievers, we evaluated different aggregation functions for reranker ensembling. Table 5.11 shows the performance of majority voting, exponential decay weighting, and

[6] https://huggingface.co/spaces/mteb/leaderboard

RRF. Among them, RRF again performed best, achieving an average S@10 of 0.9202 in the monolingual setting. This confirms that RRF is well-suited for combining reranker outputs, as it effectively prioritizes highly ranked documents while minimizing the influence of poorly ranked ones.

Aggregation Function	Avg S@10
Majority Voting	0.9075
Exponential Decay Weighting	0.9066
RRF	0.9202

Table 5.11: Ensembler performance comparison based on the different aggregation functions in the monolingual setting (S@100)

Ensemble Weighting

For the rerankers, we adopt an equal weighting scheme, as all three models demonstrated competitive and complementary performance during evaluation. For the individual reranker performance scores, refer to Table 6.3 in the results chapter. To prevent overfitting to specific domains or language patterns, we refrain from assigning differential weights. Instead, we assign equal weight to each reranker in the ensemble, ensuring robustness across diverse claim types and languages while effectively leveraging the strengths of each model.

Pipeline Overview

In the first stage, the retriever ensembler aggregates the top 300 results from multiple retrievers (BM25, E5, BGE) into a top 100 candidate set.

In the second stage, the reranking ensemble module refines the selection using cross-encoder rerankers (NV, GRITLM, QWEN). Each reranker processes the top 100 fact-checks and returns its top 50 reranked results, which are then aggregated by a final ensembler to produce the top 10 ranked fact-checks.

The described pipeline is depicted in Figure 5.1.

5.4 Evaluation Module

To assess the effectiveness of the retrieval pipeline, the evaluation module measures how well retrieved fact-checks align with ground-truth references using the described retrieval evaluation metrics.

CHAPTER 6

Monolingual Results and Analysis

We present an evaluation of the retrieval and reranking components of the proposed fact-checking pipeline in a monolingual setting. We explore the effectiveness of retrieval ensembles and assess reranking performance and their contribution to overall pipeline effectiveness.

6.1 Retrieval-Ensemble Analysis

Model size vs. Retrieval performance

We analyze the relationship between model sizes and retrieval performances of the chosen retrievers from Table 5.6.

The best-performing model, E5, achieves an average S@100 score of 0.9330 while being relatively small in size (560M parameters) compared to other competitive models. In contrast, the second-best model, BGE, achieves a similar performance with an S@100 score of 0.9293, but at a substantially larger scale, with 9.24B parameters — over 16 times the size of E5. This shows that larger models do not necessarily guarantee better retrieval performance, especially in zero-shot settings.

Other models further confirm this observation. NV-Embed-v2 performs well with an S@100 of 0.9201 but is also a large model (7.85B parameters). Meanwhile, GTR-T5-XL achieves a high score of 0.9019 with 1.24B parameters, showing a balance between size and performance. BGE-M3, a smaller variant with 568M parameters, lags behind its larger counterpart (BGE-Gemma2) with an S@100 of 0.8731, despite being of comparable size to the E5 model.

These results show that model architecture, training objectives, and multilingual capabilities likely play a more significant role in retrieval effectiveness than model size alone. The efficiency and performance balance of E5 suggests that, for this task, well-optimized mid-sized models can outperform or perform equally well as far larger models, making them suitable for practical applications, especially in resource-constrained environments.

6.1.1 Ensembler

In the retrieval-ensemble module, we first retrieve a set of top 300 candidate fact checks using each retriever model (E5, BGE, and BM25), and then an ensembler aggregates those predictions into top 100.

Ensemble Performance

Table 6.1 compares the performances of individual retrievers and ensemble configurations. The results show that dense retrievers (E5, BGE) consistently outperform the lexical BM25 in all languages, highlighting the effectiveness of semantic models. However, the ensemble methods that combine BM25 with dense retrievers (e.g., E5 + BM25, E5 + BGE + BM25) show further performance gains, achieving the highest average S@100 scores of 0.9720. Combining these approaches makes the most out of their complementary strengths, leading to more robust and accurate retrieval.

Model	k	**ARA**	**DEU**	**ENG**	**FRA**	**MSA**	**POR**	**SPA**	**THA**	**AVG**
E5	300	0.9691	0.9672	0.9580	0.9758	0.9844	0.9763	0.9719	1.0000	0.9752
BM25	300	0.9451	0.8946	0.8768	0.9345	0.9142	0.9289	0.9334	0.9886	0.9270
BGE	300	0.9657	0.9742	0.9481	0.9776	0.9649	0.9534	0.9627	1.0000	0.9683
E5 + BM25	100	0.9657	0.9344	0.9386	0.9677	0.9766	0.9575	0.9600	0.9924	0.9616
BGE + BM25	100	0.9537	0.9625	0.9358	0.9686	0.9571	0.9551	0.9558	0.9962	0.9606
E5 + BGE	100	0.9639	0.9719	0.9488	0.9722	0.9805	0.9583	0.9634	0.9962	0.9694
E5 + BM25 + BGE	100	0.9691	0.9742	0.9509	0.9722	0.9844	0.9633	0.9658	0.9962	0.9720

Table 6.1: Retrieval performance (S@k) using original-language text across models and ensembles on the training set

Retriever Ensemble Performance

We assess the module's effectiveness as a standalone component instead of as an intermediate step in the pipeline using S@10. Table 6.2 presents a performance comparison between our retrieval-ensemble module and the best-performing baseline model (GTR-T5-Large) from the Multiclaim dataset paper [PSM$^+$23], which achieves an average S@10 of 0.82. Our retrieval-ensemble outperforms the baseline across all languages, reaching an average S@10 of 0.9237.

6.2 Retriever-Reranker Analysis

We evaluated the retriever-reranker pipeline, a two-stage retrieval approach that aims to refine the initial retrieval results for improved fact-check retrieval.

Model	k	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
Baseline Model (GTR-T5-Large)	10	0.86	0.69	0.77	0.86	0.82	0.80	0.84	0.90	0.82
Retriever Ensembler (E5 + BM25 + BGE)	10	0.9280	0.8923	0.8784	0.9408	0.9279	0.9158	0.9296	0.9772	0.9237

Table 6.2: Retrieval-ensemble performance (S@10) using original-language text on the training set

Model	k	ARA	DEU	ENG	FRA	MSA	POR	SPA	THA	AVG
NV	50	0.9503	0.9438	0.9386	0.9596	0.9591	0.9379	0.9472	0.9734	0.9512
NV (EN)	50	**0.9588**	0.9297	0.9386	0.9596	0.9493	0.9297	0.9444	**0.9886**	0.9498
NV (TRANSL)	50	0.9451	0.9438	0.9386	0.9614	**0.9669**	0.9395	0.9317	0.9544	0.9477
GRITLM	50	0.9468	**0.9485**	**0.9394**	0.9650	0.9630	0.9379	**0.9548**	0.9772	**0.9541**
GRITLM (EN)	50	0.9571	0.9204	0.9394	**0.9659**	0.9571	**0.9453**	0.9517	**0.9886**	0.9532
GRITLM (TRANSL)	50	0.9485	0.9321	0.9394	0.9650	0.9649	0.9404	0.9527	0.9316	0.9468
QWEN	50	0.9451	**0.9485**	0.9235	0.9534	0.9630	0.9436	0.9448	0.9734	0.9494
QWEN (EN)	50	0.9503	0.9110	0.9235	0.9453	0.9376	0.9093	0.9247	0.9810	0.9353
QWEN (TRANSL)	50	0.9430	0.9321	0.9235	0.9459	0.9643	0.9411	0.9421	0.9710	0.9454
Baseline Model (GTR-T5-Large)	10	0.86	0.69	0.77	0.86	0.82	0.80	0.84	0.90	0.82
QWEN + GRITLM	10	0.9177	0.8478	0.8792	0.9318	**0.9181**	0.9101	0.9196	0.9316	0.9070
QWEN + NV	10	0.9262	0.8501	0.8803	0.9309	0.9045	0.9003	0.9058	0.9316	0.9037
NV + GRITLM	10	0.9091	0.8618	**0.8942**	0.9363	0.9103	0.9028	0.9247	0.9087	0.9060
QWEN + GRITLM + NV	10	**0.9297**	**0.8735**	0.8938	**0.9444**	**0.9181**	**0.9142**	**0.9261**	**0.9620**	**0.9202**

Table 6.3: Reranking performance on the training set. EN refers to experiments using the English version of the data, while TRANSL refers to the original language with injected task instructions and descriptions translated into the language of origin. The best scores per language are in bold.

Evaluation Setup

We evaluated models in three settings to examine the impact of language representation and instruction translation:

1. Using the original language.
2. Using the provided English translation (EN).
3. Using the original version, with also translating the injected task descriptions and instructions into the respective language (TRANSL).

The idea behind the third setting was to investigate whether injecting task descriptions and instructions translated to the original language can improve the quality of the retrieval. In the first setting, instructions and task descriptions are injected in English regardless of the language.

Reranker Performance

Table 6.3 presents the S@50 and S@10 scores for individual rerankers and their ensembles across languages.

GRITLM achieves the highest average S@50 score (0.9541) when using original-language inputs, closely followed by NV (0.9512) and QWEN (0.9494).

Across individual language performances, Thai consistently achieves the highest scores (0.9886 with NV EN), likely due to the smaller size of its fact-check corpus. German and English, on the other hand, show greater variability in performance, which can be due to larger datasets and more diverse linguistic patterns.

Performance in English-translated versions (EN) shows mixed results in all languages. Arabic (ARA) and Thai (THA) benefit the most from translation, as they achieve their best results using the English translation with all three models. Both achieved their highest S@50 scores (0.9588 and 0.9886) with NV EN. Translation into English can normalize morphologically rich languages, improving retrieval for models trained on English-heavy corpora.

German (DEU) and Spanish (SPA) experience performance drops in the EN setting, as translating into English can also remove important linguistic features or introduce translation errors. This highlights that for high-resource languages, preserving the original text often leads to more reliable fact-check retrieval.

Injecting task instructions in the original language (TRANSL) shows mixed results. Malay (MSA) shows a noticeable improvement (0.9669 NV TRANSL) compared to the original (0.9591 NV), while other languages, such as Thai (THA), show performance drops. This suggests that translation quality and the nature of the underlying language structure may impact the effectiveness of this approach.

To ensure consistency, we used the on-average best-performing setup across languages. However, language-specific tuning could further enhance retrieval, as languages respond differently to translation and formatting.

Reranker Ensemble Performance

Ensembling multiple rerankers enhances their individual performance, leading to more effective rankings. The ensemble incorporating all three models (QWEN + GRITLM + NV) achieves the highest average S@10 score of 0.9202, consistently surpassing pairwise combinations (e.g., QWEN + GRITLM: 0.9070) indicating that diverse reranker combinations result in more robust rankings. Furthermore, the ensemble significantly outperforms the baseline (0.9202 vs. 0.82). The results are presented in Table 6.3.

Effectiveness of Reranking

Despite the expected advantages of reranking, our results indicate that the retriever-reranker pipeline delivers only marginal improvements in Arabic, English, and French, failing to consistently outperform retriever-ensemble approach. Our results demonstrate that hybrid retrieval strategies — combining lexical and dense models — are both more effective and computationally efficient than stacking increasingly complex neural architectures. The limited improvements from reranking suggest that retrieval bottlenecks cannot always be resolved through additional processing, reinforcing the importance of well-designed ensembling over the reliance on increasingly complex models. This highlights that retrieval performance can be optimized efficiently without excessive computational overhead.

Model	**ARA**	**DEU**	**ENG**	**FRA**	**MSA**	**POR**	**SPA**	**THA**	**AVG**
Retriever Ensembler (E5 + BM25 + BGE)	0.9280	0.8923	0.8784	0.9408	0.9279	0.9158	0.9296	0.9772	**0.9237**
Reranker Ensembler (QWEN + GRITLM + NV)	0.9297	0.8735	0.8938	0.9444	0.9181	0.9142	0.9261	0.9620	0.9202

Table 6.4: Comparison of the retrieval-ensemble and the full retriever-reranker pipelines (S@10)

Table 6.4 compares the performances of the retrieval-ensemble and the full retriever-reranker pipeline.

Addressing RQ3: Among all retriever-reranker configurations, the most consistent performance is achieved with a pipeline that combines E5, BM25, and BGE as retrievers with QWEN, GRITLM, and NV as rerankers achieving an average S@10 of 0.9202 while maintaining high and stable performance without significant drops in any of the languages.

Interestingly, however, the retrieval-ensemble setup that integrates E5, BM25, and BGE without rerankers yields the highest monolingual retrieval performance, reaching an average S@10 of 0.9237.

6.3 Pipeline Evaluation on CheckThat! 2021 Dataset

To assess the effectiveness of the constructed monolingual PFCR pipeline, we conduct an evaluation using the CheckThat! 2021 Task 2A dataset. This benchmark allows us to measure how well the pipeline retrieves relevant fact-checks for social media claims in English and Arabic. We compare the performance of two configurations (retriever-ensembler and retriever-reranker) against the official results reported on the CheckThat! 2021 Task 2A leaderboard.

6.3.1 Performance on English Data

Table 6.5 compares our retriever ensembler and retriever reranker setups with the top 3 submissions on the English development set.

Team Aschern used TF-IDF, fine-tuned pre-trained sentence-level BERT, and the re-ranking LambdaMART model. Team DIPS used Sentence-BERT embeddings for all claims and then computed the cosine similarity for each input tweet - verified claim pair.

Team	MAP@5	MAP@1	MAP@3	MAP@10	MAP_All	MRR	P@3	P@5	P@10	S@10
Aschern	**0.941**	**0.932**	**0.941**	**0.941**	**0.943**	**0.940**	**0.318**	0.191	0.095	NA
DIPS	0.936	0.927	0.935	0.937	0.937	0.935	0.315	0.190	0.096	NA
TIB-VA	0.902	0.857	0.900	0.902	0.903	0.901	**0.318**	**0.192**	**0.096**	NA
Retriever Ensembler	0.754	0.686	0.738	0.763	0.766	0.766	0.2674	0.175	0.094	0.936
Retriever Reranker	0.900	0.870	0.896	0.902	0.902	0.902	0.309	0.189	**0.096**	0.956

Table 6.5: Our results vs. top 3 submissions on the CheckThat! 2021 Task 2A leaderboard on the development set using English data. Best values per column are shown in bold.

The prediction was made by passing a sorted list of cosine similarities to a neural network [NME+21b]. As for Team TIB-VA, no system description paper has been published, so the details of their implementation remain unavailable.

Our retriever reranker pipeline delivers competitive results in MAP@k and MRR but falls slightly behind the top teams, while our retriever ensembler demonstrates strong retrieval effectiveness (S@10=0.936), ensuring that relevant fact-checks are retrieved. However, its lower MAP and MRR scores indicate that it does not always rank them optimally.

The primary difference between our approach and the top-performing systems is that our models are used in their pre-trained state, whereas the leading solutions are fine-tuned for PFCR. While fine-tuning provides a performance advantage, it also demands additional computational resources and development overhead. In contrast, our results demonstrate that pre-trained models can still achieve strong retrieval performance, offering a more scalable solution for real-world fact-checking applications.

6.3.2 Performance on Arabic Data

Team	MAP@5	MAP@1	MAP@3	MAP@10	MAP_All	MRR	P@3	P@5	P@10	S@10
TIB-VA	0.815	0.722	0.812	0.816	0.821	0.833	**0.341**	**0.207**	**0.105**	NA
bigIR	0.819	0.710	0.816	0.821	0.822	0.832	0.345	0.209	0.106	NA
Retriever Ensembler	0.795	0.713	0.786	0.802	0.804	0.804	0.2917	0.183	0.097	0.969
Retriever Reranker	**0.885**	**0.803**	**0.881**	**0.886**	**0.886**	**0.886**	0.323	0.197	0.099	**0.992**

Table 6.6: Our results vs. top 2 submissions on the CheckThat! 2021 Task 2A leaderboard on the development set using Arabic data. Best values per column are shown in bold.

Table 6.6 compares our retriever ensembler and retriever reranker setups with the top 2 submissions on the Arabic development set.

Team bigIR fine-tuned AraBERT [ABH20] by adding two neural network layers on top of it to predict the relevance for a given tweet–VerClaim pair. The fine-tuned model was used to re-rank the candidate claims based on the predicted relevance scores [NME+21b].

Our retriever reranker pipeline achieves the highest performance in MAP@k and MRR, indicating that our system is more precise at ranking relevant fact-checks than the other approaches. However, it does not achieve the best performance in P@k. This suggests that while our method is effective at ranking relevant documents, it retrieves fewer top-k relevant documents compared to other teams. Even though our retriever ensembler shows a high S@10 score of 0.969, it performs worse in comparison to the other approaches. This could be because the retriever ensembler prioritizes recall over precision, ensuring that relevant fact-checks are included in the retrieved set but not necessarily ranked optimally. This trade-off highlights the difference between retriever-ensembling and retriever-reranking - the retriever-ensembler increases the likelihood of retrieving the relevant fact-checks, while the reranker ensures that the most relevant results appear at the top.

6.3.3 Performance Comparison between the Languages

Our retriever reranker pipeline outperforms the top-ranked systems in Arabic, while fine-tuned solutions achieve higher performance in English.

This is likely due to the pre-training bias toward English. Many state-of-the-art retrieval and reranking models are trained on large-scale English corpora, benefiting fine-tuned English models. In contrast, fewer Arabic resources are available, making fine-tuned models less effective. Our approach uses multilingual pre-trained models, which were pre-trained on Arabic data, explaining why our retriever reranker pipeline achieves state-of-the-art results in Arabic, even without fine-tuning.

6.4 Error Case Study Analysis

To conduct an error analysis and identify key failure points of our framework for the monolingual retrieval, we examined two types of errors;

- Cases where individual retrievers failed but the ensembler successfully retrieved the correct fact-check, and
- Cases where both the individual retrievers and the ensembler failed to retrieve the correct fact-check.

We give an example of the first-case error with the following claim:

The following claim was posted: This is what real leadership looks like. The Ukrainian President on the ground. The US offered to evacuate him, but he chose to stay in Ukraine, posted on 2022-07-11.
The content is labeled as: False information.

The correct fact-check is:

This is a fact-checked claim: Ukrainian President Volodymyr Zelensky on battlefield in 2022, with the title Photo of Ukrainian President Zelensky in military gear was taken in 2021 — before Russian invasion posted on 2022-03-03.

None of the individual retrieval models included this fact-check in their top 10 predictions. Instead, they retrieved fact-checks covering related but distinct narratives, such as

Zelensky's whereabouts, misleading battlefield images, and claims about him fleeing Kyiv. Even though these fact-checks are contextually relevant, they do not directly address the specific claim. However, the ensembler successfully retrieved the correct fact-check using the strengths of multiple retrieval methods. This suggests that while individual models recognized its relevance, they did not prioritize it effectively. This highlights the advantage of hybrid approaches that combine lexical precision with semantic understanding, ensuring that fact-checks capturing both keyword overlap and conceptual similarity are ranked more effectively.

For the second type of error, where the ensembler also fails, we consider the following claim:

The following claim was posted: 5 10 Amy Cutshall-Benson 4 hrs Like • .. I have tried to share this 3 times and Facebook won't let me... if anybody can see this pls comment 4G st Comment Write a comment... 65% 4:30 AM : 3 Comments Send posted on 2020-11-06. The content is labeled as: Missing context.

The true fact-checks are:

This is a fact-checked claim: Map shows 2020 election results, with the title Election results map spreading on social media is from 2016, not 2020 posted on 10-11-2020.

This is a fact-checked claim: Old US election map misleads on voting trend in 2020 election, does not show Electoral College, with the title Old US election map misleads on voting trend in 2020 election posted on 2020-11-11.

The retrieved fact-checks are about Facebook allegedly banning or limiting post sharing, including claims about censorship, suppression of conservative news, and restrictions on sharing election-related content. Some retrieved fact-checks do mention an electoral map but do not match the intended verification. The challenge arises because the claim is labeled "Missing context" and does not explicitly reference election results, making it difficult for retrieval models to establish the correct connection.

These examples show key limitations in retrieval; related but incorrect fact-checks may be prioritized when claims are ambiguous, and implicit connections between claims and fact-checks can be difficult to capture. Even as ensembling improves performance, retrieval effectiveness remains sensitive to claim formulation and ambiguity.

6.5 Monolingual Results Summary

This chapter evaluated the effectiveness of a fact-checking pipeline in a monolingual setting. The retrieval-ensemble analysis shows that smaller, optimized models like E5 (560M parameters) outperform larger models, with E5 + BM25 + BGE achieving the highest performance. The retriever-reranker pipeline, tested on both original and translated tasks, shows that a hybrid approach with lexical and dense models is not only more efficient but also more effective than complex reranking architectures. While reranking enhances retrieval performance for languages like Arabic, English, and French, retrieval-ensemble methods generally outperform the retriever-reranker pipeline, suggesting that well-designed ensembling is more efficient than relying on increasingly complex models. On the CheckThat! 2021 dataset, the retriever-reranker setup surpassed the retrieval ensemble in English and Arabic, though both approaches were outperformed by fine-tuned models. In conclusion, combining mid-sized models like E5 with retrieval ensembles delivers strong results for fact-check retrieval tasks, with room for further enhancement through fine-tuning.

CHAPTER 7

Crosslingual Results and Analysis

In this chapter, we evaluate the retrieval performance of the crosslingual subtask by comparing the retrieval-ensemble and retriever-reranker configurations. We analyze the effectiveness of individual models, ensemble methods, and the two configurations, considering both translation-based and multilingual approaches.

7.1 Crosslingual Data Example

In this section, we illustrate an example of crosslingual data, where a claim made in one language is linked to a fact-check in another. To maintain consistency, we apply the same input formatting described earlier for crosslingual cases as well.

For example, the following claim was originally posted in Arabic:

The following claim was posted: چنوٹ مىں تىن مرىض قرنطنہ سي بہاگ گئي پہر جو هوا آپكي سامني هي *posted on 2020-12-04. The content is labeled as: False information.*

Its English translation is as follows:

The following claim was posted: Three patients escaped from the quarantine in Cheniot, then what happened is in front of you posted on 2020-12-04. The content is labeled as: False information.

The claim is linked to the following fact-checked statement, which was originally published in English:

This is a fact-checked claim: Pakistani forces apprehend COVID-19 quarantine escapees with the title This video has circulated in reports about Pakistani police and security forces conducting a training drill at a quarantine centre posted on 2020-16-04.

This example illustrates a scenario where the claim is posted in Arabic, while the fact-check is available in English. This highlights the challenges of crosslingual retrieval, as models must bridge the linguistic gap to correctly associate claims with fact-checks in different languages.

7.2 Retrieval-Ensemble Analysis

We evaluated the performance of individual retrievers and their ensemble for crosslingual retrieval, considering both original-language and English-translated versions. In Table 7.1, we report S@300 scores of the retrievers as the ensembler operates over the top 300 retrieved candidates.

Consistent with monolingual findings, dense retrievers (E5, BGE) outperform BM25, demonstrating the advantages of semantic retrieval. BGE (original) achieves the highest individual performance (S@300 = 0.9257), while E5 benefits from translation, improving from 0.8623 (original) to 0.8963 (English). BM25 is evaluated only in English due to its reliance on lexical overlap.

Based on our findings above, we aggregate the top 300 predictions from BM25, E5, and BGE using RRF with a weighting scheme of (0.5 BM25 + 1.0 E5 + 1.0 BGE). The ensembler then selects the top 100 candidates for further reranking.

Our best ensembler configuration (BM25 english + E5 english + BGE original), composed of the models in their best-performing settings, achieves S@100 of 0.9094 and S@10 of 0.7572. The S@100 score reflects the performance of the intermediate step within the pipeline, which is then passed to the rerankers. In contrast, the S@10 score highlights the module's effectiveness as a standalone component. The combination of E5 (english) and BGE (original) balances translated and original representations, enhancing crosslingual retrieval. These results confirm that combining lexical and semantic retrieval enhances crosslingual PFCR and demonstrates that a well-weighted ensemble improves retrieval effectiveness.

Table 7.1 summarizes the results of retrievers and the ensemble.

Model	**Language Version**	k	**S@k**
BM25	English	300	0.7884
E5	Original	300	0.8623
	English	300	0.8963
BGE	Original	300	0.9257
	English	300	0.9054
Ensembler	BM25 (orig) + E5 (eng) + BGE (orig)	100	0.9094
		10	0.7572

Table 7.1: Performance comparison of retrievers for crosslingual retrieval (S@k)

7.3 Retriever-Reranker Analysis

This section compares the performance of selected rerankers for the crosslingual retrieval task, assessing both the original-language and English-translated versions. We evaluated reranking effectiveness using S@50 as the ensembler performs aggregation on the reranker's top 50 candidates. Finally, we show the performance of the full pipeline using S@10. Table 7.2 presents the results.

Model	**Language Version**	k	**S@k**
NV	Original	50	0.8290
	English	50	0.8749
QWEN	Original	50	0.8471
	English	50	0.8612
GRITLM	Original	50	0.8682
	English	50	0.8799
Ensembler	NV (eng) + QWEN (eng) + GRITLM (orig)	10	0.7951

Table 7.2: Performance comparison of individual rerankers and ensembler configurations (S@k)

English-translated versions generally improve performance, with all three rerankers showing higher S@50 scores compared to their original-language counterparts, indicating that translation reduces linguistic variability and makes ranking more consistent across different languages.

Among individual rerankers, GRITLM achieved the highest scores across both original and translated versions, outperforming NV and QWEN. This suggests that GRITLM is better at capturing fine-grained semantic relationships between claims and fact-checks. NV performed well with the English version, achieving S@50 of 0.8749, while QWEN is slightly behind in both settings, with S@50 of 0.8612 and 0.8471 in English and original versions, respectively.

After evaluating all model and version combinations, the best performance was achieved with the ensembler configuration that combined NV (English), QWEN (English) and GRITLM (original), reaching an S@10 of 0.7951. This demonstrates the effectiveness of hybrid ensembling, showing that combining diverse rerankers enhances ranking robustness beyond what individual models can achieve. Interestingly, the inclusion of an original-language model suggests that translation is not always the optimal solution: language-specific nuances provide valuable ranking signals that contribute to improved retrieval accuracy.

Effectiveness of Reranking

In contrast to our findings in monolingual retrieval, our results for crosslingual retrieval indicate that reranking provides an improvement over the retriever-ensemble approach. As shown in Table 7.3, the reranker ensembler (QWEN + GRITLM + NV) achieves a significantly higher S@10 score (0.7951) compared to the retriever ensembler (E5 + BM25 + BGE) at 0.7572. This suggests that in crosslingual settings, retrieval bottlenecks are more effectively addressed through reranking, likely due to the complexities of language transfer and semantic alignment across different languages. The effectiveness of the reranking step highlights its role in refining retrieved candidates, making it a crucial component in improving performance for crosslingual retrieval tasks.

Model	**S@10**
Retriever Ensembler (E5 + BM25 + BGE)	0.7572
Reranker Ensembler (QWEN + GRITLM + NV)	0.7951

Table 7.3: Comparison of retriever ensembler and the full retriever-reranker pipelines (S@10)

Execution Time and Memory Usage

To address RQ2, we compare the S@10 scores, execution time, and memory usage of models used for crosslingual retrieval, evaluating both multilingual and translation-based (English) approaches. This comparison allows us to assess whether mapping all content into a single language embedding space (via translation) enhances retrieval performance while losing nuances of the original language, or whether using a shared multilingual embedding space yields better results.

Evaluation was conducted on the whole corpus using both the retriever and reranker models in isolation, enabling a controlled comparison of their effectiveness and efficiency in each setting.

We define execution time as the average time required to retrieve the top 10 documents for a query, while memory usage is measured in terms of RAM/VRAM consumption and model size (number of parameters). Table 7.4 presents a comparative analysis of these factors. Since the same underlying models process both English and multilingual inputs, their execution time and memory footprint remain consistent across both settings.

Addressing RQ2:

Since both approaches (translation-based and multilingual) rely on the same underlying models, execution time and memory usage remain unchanged. This means that the choice between them should be based on their retrieval performance, measured by S@10.

In terms of efficiency, the E5 model offers the fastest execution time (0.08s), making it well-suited for real-time retrieval, whereas cross-encoders such as NV and GRITLM are significantly slower, exceeding 1 second per query.

Model	Time (s)	N Parameters	Memory (MB)	S@10 (English)	S@10 (Original)
E5	0.08	560M	3964	0.6918	0.5962
BGE	0.39	9.24B	19566	0.7267	0.7409
NV	1.28	7.85B	36104	0.7582	0.6311
QWEN	0.40	7.61B	31871	0.6925	0.6673
GRITLM	1.23	7.24B	31920	0.7653	0.7304

Table 7.4: Comparison of S@10, execution time, model size, and memory usage for retrieval and reranking models.

Memory usage varies, with BGE maintaining relative efficiency despite its 9.24B parameters, while models like NV and QWEN consume over 30GB of memory, making them computationally expensive.

Regarding retrieval effectiveness, GRITLM achieves the highest S@10 scores, 0.7653 in English, 0.7304 in the original language. BGE also performs well (0.7267 in English, 0.7409 in the original language) while being considerably faster, making it a balanced choice for both speed and accuracy. NV, despite achieving a strong performance in English (0.7582), shows a drop in the original language (0.6311), suggesting potential language biases during training.

These results suggest that translation-based approaches are preferred when using bi-encoders like BGE, which provide a balance between efficiency and accuracy, while GRITLM and NV offer higher retrieval precision but with a substantial computational cost.

While in our setting, translation-based retrieval improved ranking accuracy without increasing computational costs, it is important to consider the real-world feasibility of translation. In this work, translations were provided within the dataset, but in real-world applications, translation introduces an additional processing step that could affect efficiency. This highlights the need to weigh the accuracy of the retrieval against the potential overhead when choosing between multilingual and translation-based retrieval strategies.

7.4 SemEval 2025 Task 7 Submission Results

We present the results achieved during our participation in the SemEval 2025 Task 7.

During the shared task, the organizers first released the "train" and "dev" dataset to tune the models, but the final evaluation was done on the "test" set.

Table 7.5 compares our test set performance with the organizer's best-performing model used as a baseline and the best-performing model on the leaderboard. Our approach outperformed the organizer's baseline in both monolingual (S@10: 0.93 vs. 0.84) and crosslingual retrieval (S@10: 0.75 vs. 0.59). The top leaderboard model achieved 0.96 and 0.86, respectively.

For the monolingual submission, we selected the best-performing setup per language, choosing between retrieval-ensemble and retriever-reranker configurations. We used the retriever- ensemble for Arabic, Malay, German, Thai and Turkish, and the full retriever-reranker pipeline for English, French, Spanish, Portuguese and Polish. For the crosslingual subtask, we used the retrieval-ensemble setup instead of the retrieval-reranker setup due to insufficient time for further evaluation.

Task	**Baseline**	**Best**	**Our Score**
Monolingual	0.84 ± 0.01	0.96	0.93
Crosslingual	0.59 ± 0.05	0.86	0.75

Table 7.5: Test set performance comparison (S@10)

CHAPTER 8

Conclusion

We summarize the key contributions and limitations and outline potential directions for future research.

8.1 Contributions

In this thesis, we explored the task of multilingual and crosslingual previously fact-checked claim retrieval, addressing the challenge of efficiently retrieving relevant fact-checks across different languages.

To answer RQ1, we designed a pipeline above that enhanced the performance of BM25 for monolingual claim retrieval, and confirmed that a well-designed preprocessing pipeline enhances lexical retrieval by reducing both false negatives and false positives. The average S@10 score increased to 0.7229 for the original-language training data and to 0.7967 for the English-translated version.

To address RQ2, we compared retrieval performances for original-language and English-translated versions for crosslingual retrieval. Translated versions often achieved higher S@10 scores, however, as both approaches use the same models, the execution time and memory footprint remained unchanged. In real-world deployment, one would need to consider translation overhead.

For RQ3, we examined the system's robustness across multiple languages. The retriever-reranker setup, with E5, BM25, and BGE as retrievers and QWEN, GRITLM and NV as rerankers, delivered the most consistent monolingual retrieval performance achieving an average S@10 of 0.9202. However, the retrieval-ensemble setup that integrates E5, BM25, and BGE without rerankers yielded the highest monolingual retrieval performance, reaching an average S@10 of 0.9237.

The main conclusions of this work were that in the monolingual setting of PFCR, reranking provides only marginal improvements over hybrid retriever ensembling, suggesting that ensemble-based retrieval alone is effective in balancing accuracy and efficiency. This highlights the strength of combining lexical and dense retrievers rather than relying on additional reranking steps. On the other hand, for the crosslingual setting, the retriever-reranker configuration proved to be the most effective approach. The additional reranking step played a more significant role in refining retrieval, likely due to the increased variability in language structure and translation-induced noise.

Our work was conducted within the scope of the SemEval-2025 Shared Task 7 Lab, where we submitted a paper titled: "ipezoTU at SemEval-2025 Task 7: Hybrid Ensemble Retrieval for Multilingual Fact-Checking: Balancing Efficiency and Accuracy" [PHS25], which is currently under review.

The complete codebase developed for this thesis is publicly available on GitHub[1].

8.2 Limitations

Although our approach demonstrates strong retrieval effectiveness, several limitations must be acknowledged.

MultiClaim Dataset Creation Limitations. The authors of the MultiClaim dataset state that the dataset was processed using third-party AI services for machine translation to English and language detection. Both of those introduce additional sources of noise, potentially impacting retrieval accuracy and crosslingual alignment. Additionally, the presence of non-textual claims, which rely on visual information (e.g., images, videos, or memes) to convey misinformation, remains a significant challenge for retrieval performance as optical character recognition (OCR) tools often introduce noise and errors.

Computational Constraints. Our retrieval-reranking pipeline employs large-scale neural models (e.g., BGE, GRITLM, and QWEN), which demand significant GPU resources for inference. While these models show high retrieval accuracy, further improvements could be achieved by utilizing higher-precision computations or processing larger batches. However, these enhancements were constrained by resource limitations.

Existance of Previously Fact-Checked Claim. The developed pipeline relies on the availability of previously fact-checked claims. However, this approach faces limitations when such claims are not available and relies heavily on manual fact-checkers for validation.

Real World Application. Due to the reliance on pre-trained models, retrieval effectiveness may vary across unseen domains and evolving misinformation trends. Additionally, real-time fact-checking requires efficient query processing, yet some models have high

[1] https://github.com/ivapezo/Multilingual-and-Crosslingual-Fact-Checked-Claim-Retrieval/

computational costs, making large-scale deployment challenging. Finally, language coverage remains a constraint, as retrieval effectiveness may decline in low-resource languages with fewer resources.

Trust in AI. Public trust in AI-driven fact-checking remains a challenge, as users may be sceptical of automated systems, particularly when explanations are lacking or errors occur. LLMs can introduce biases, hallucinate, or struggle with nuanced claims, raising concerns about their reliability in fact verification. Additionally, growing opposition to fact-checking efforts, due to political polarization and distrust in media, has led to resistance against automated fact-checking systems. These trends highlight the need for transparency, interpretability, and validation methods to build user confidence in AI-assisted verification [LWV24].

8.3 Future Work

Potential directions for future research include the following:

1. Enhancing retrieval with k-shot learning and domain-specific fine-tuning. Integrating k-shot retrieval could improve generalization to unseen claims with limited labeled examples, enabling the system to adapt with minimal supervision. Additionally, fine-tuning retrieval models on fact-checking datasets could optimize performance, ensuring stronger alignment with the requirements of fact verification.
2. Adapting retrieval strategies based on language properties and claim complexity. Since languages differ in morphology, syntax, and availability of resources, retrieval strategies could be tailored accordingly. Additionally, claims vary in complexity — some are straightforward paraphrases, while others require deeper contextual understanding. Dynamic retrieval approaches that adjust based on language structure and claim complexity could enhance both efficiency and accuracy, particularly in low-resource settings.
3. Improving ranking quality through claim entailment verification. Incorporating an entailment verification step into the ranking process could help determine whether retrieved fact-checks support or contradict a claim. This refinement would mitigate irrelevant results, ensuring that retrieved evidence is not only topically relevant but also semantically aligned with the claim.
4. Integrating newer architectures and evolving large language models. As LLMs continue to advance, integrating state-of-the-art models with improved contextual reasoning and multilingual capabilities could enhance retrieval and reranking. Additionally, exploring alternative retrieval architectures such as retrieval-augmented generation (RAG) could further optimize performance.
5. Evaluation of the framework on additional datasets to further validate its generalizability.

By addressing these aspects, future work can further improve fact-check retrieval, making it more efficient and adaptable across diverse languages and claim types.

Overview of Generative AI Tools Used

In this thesis, Grammarly [Gra25] and ChatGPT [Ope25] (versions 3 and 4), were only used to help with grammatical structure and improve sentence clarity.

Generally, the following prompt template was employed to refine sentences that required improvement:

```
Please rewrite this sentence to improve its clarity: {sentence}.
```

Once the chatbot suggested a refinement, the sentence was manually adjusted to fit the context. Furthermore, ChatGPT was used to assist with refining equations and formatting tables in LaTeX.

Additionally, DeepL [Tra25] was used for assistance with translating the abstract to German.

List of Figures

List of Tables

Bibliography

[ABC$^+$14] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 task 10: Multilingual semantic textual similarity. In Preslav Nakov and Torsten Zesch, editors, *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics.

[ABC$^+$15] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In Preslav Nakov, Torsten Zesch, Daniel Cer, and David Jurgens, editors, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics.

[ABC$^+$16] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In Steven Bethard, Marine Carpuat, Daniel Cer, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. Association for Computational Linguistics.

[ABH20] Wissam Antoun, Fady Baly, and Hazem Hajj. AraBERT: Transformer-based model for Arabic language understanding. In Hend Al-Khalifa, Walid Magdy, Kareem Darwish, Tamer Elsayed, and Hamdy Mubarak, editors, *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France, May 2020. European Language Resource Association.

[ACD+13] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In Mona Diab, Tim Baldwin, and Marco Baroni, editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[ACDGA12] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In Eneko Agirre, Johan Bos, Mona Diab, Suresh Manandhar, Yuval Marton, and Deniz Yuret, editors, **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[AOS24] Abdelkrim Aarab, Ahmed Oussous, and Mohammed Saddoune. Optimizing arabic information retrieval: A comprehensive evaluation of preprocessing techniques. In *2024 IEEE 12th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pages 1–4, 2024.

[BBVEF+24] Oana Balalau, Pablo Bertaud-Velten, Younes El Fraihi, Garima Gaur, Oana Goga, Samuel Guimaraes, Ioana Manolescu, and Brahim Saadi. FactCheckBureau: Build Your Own Fact-Check Analysis Pipeline. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pages 5185–5189, New York, NY, USA, October 2024. Association for Computing Machinery.

[BCEN+20] Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. Overview of CheckThat! 2020: Automatic Identification and Verification of Claims in Social Media. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 215–236, Cham, 2020. Springer International Publishing.

[BCnEN+20] Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings*, page 215–236, Berlin, Heidelberg, 2020. Springer-Verlag.

[BEN$^+$20] Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, Fatima Haouari, Nikolay Babulkov, Bayan Hamdan, Alex Nikolov, Shaden Shaar, and Zien Sheikh Ali. Overview of checkthat 2020: Automatic identification and verification of claims in social media. *CoRR*, abs/2007.07997, 2020.

[BGJM16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[BKC22] Varad Bhatnagar, Diptesh Kanojia, and Kameswari Chebrolu. Harnessing abstractive summarization for fact-checked claim detection. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2934–2945, Gyeongju, Republic of Korea, October 2022. International Committee on 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. *CoRR*, abs/2005.14165, 2020.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117, 1998. Proceedings of the Seventh International World Wide Web Conference.

[CCB09] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 758–759, New York, NY, USA, 2009. Association for Computing Machinery.

[CDA+17] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel Cer, and David Jurgens, editors, *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[CFG+21] Yinqiong Cai, Yixing Fan, Jiafeng Guo, Fei Sun, Ruqing Zhang, and Xueqi Cheng. Semantic models for the first-stage retrieval: A comprehensive review. *CoRR*, abs/2103.04831, 2021.

[CLMS23] Tanmoy Chakraborty, Valerio La Gatta, Vincenzo Moscato, and Giancarlo Sperlì. Information retrieval algorithms and neural ranking models to detect previously fact-checked information. *Neurocomputing*, 557:126680, 2023.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[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.

[Fol96] Peter W Foltz. Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, & Computers*, 28:197–202, 1996.

[Gra25] Inc. Grammarly. Grammarly. https://www.grammarly.com, 2025. Accessed: 2025-03-17.

[HBMC24] Scott A. Hale, Adriano Belisario, Ahmed Mostafa, and Chico Camargo. Analyzing Misinformation Claims During the 2022 Brazilian General Election on WhatsApp, Twitter, and Kwai, January 2024. arXiv:2401.02395.

[HCK+22] Momchil Hardalov, Anton Chernyavskiy, Ivan Koychev, Dmitry Ilvovsky, and Preslav Nakov. CrowdChecked: Detecting previously fact-checked claims in social media. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 266–285, Online only, November 2022. Association for Computational Linguistics.

[HKMY20] Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, 15(5):1–22, 05 2020.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[Jon94] Karen Sparck Jones. *Natural Language Processing: A Historical Review*, pages 3–16. Springer Netherlands, Dordrecht, 1994.

[KGGH21a] Ashkan Kazemi, Kiran Garimella, Devin Gaffney, and Scott Hale. Claim Matching Beyond English to Scale Global Fact-Checking. 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 4504–4517, Online, August 2021. Association for Computational Linguistics.

[KGGH21b] Ashkan Kazemi, Kiran Garimella, Devin Gaffney, and Scott Hale. Claim matching beyond English to scale global fact-checking. 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 4504–4517, Online, August 2021. Association for Computational Linguistics.

[KLPRM21] Ashkan Kazemi, Zehua Li, Verónica Pérez-Rosas, and Rada Mihalcea. Extractive and Abstractive Explanations for Fact-Checking and Evaluation of News. In Anna Feldman, Giovanni Da San Martino, Chris Leberknight, and Preslav Nakov, editors, *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 45–50, Online, June 2021. Association for Computational Linguistics.

[KZL$^+$20] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: A hybrid approach. *CoRR*, abs/2010.01195, 2020.

[LMR$^+$24] Jurek Leonhardt, Henrik Müller, Koustav Rudra, Megha Khosla, Abhijit Anand, and Avishek Anand. Efficient neural ranking using forward indexes and lightweight encoders. *ACM Trans. Inf. Syst.*, 42(5), April 2024.

[LPS16] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6, 2016.

[LWV24] Q. Vera Liao and Jennifer Wortman Vaughan. AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap. *Harvard Data Science Review*, (Special Issue 5), may 31 2024. https://hdsr.mitpress.mit.edu/pub/aelql9qy.

[MCCD13] Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, 01 2013.

[Mil95] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[Moo52] Calvin Mooers. Information retrieval viewed as temporal signaling. In *Proceedings of the International Congress of Mathematicians*, 1952.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[NCH^{+}21] Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. Automated fact-checking for assisting human fact-checkers. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4551–4558. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[NDSME^{+}21] Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, Bayan Hamdan, Zien Sheikh Ali, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, Thomas Mandl, Mucahid Kutlu, and Yavuz Selim Kartal. Overview of the clef–2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings*, page 264–291, Berlin, Heidelberg, 2021. Springer-Verlag.

[NMA^{+}22] Preslav Nakov, Giovanni Da San Martino, Firoj Alam, Shaden Shaar, Hamdy Mubarak, and Nikolay Babulkov. Overview of the clef-2022 checkthat! lab task 2 on detecting previously fact-checked claims. In *Conference and Labs of the Evaluation Forum*, 2022.

[NME^{+}21a] Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, Bayan Hamdan, Zien Sheikh Ali, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, Thomas Mandl, Mücahid Kutlu, and Yavuz Selim Kartal. Overview of the CLEF-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. *CoRR*, abs/2109.12987, 2021.

[NME^{+}21b] Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeño, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Watheq Mansour, Bayan Hamdan, Zien Sheikh Ali, Nikolay Babulkov, Alex Nikolov, Gautam Kishore Shahi, Julia Maria Struß, Thomas Mandl, Mucahid Kutlu, and Yavuz Selim Kartal. Overview of the clef–2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news, 2021.

[Ope25] OpenAI. Chatgpt. https://chat.openai.com, 2025. Accessed: 2025-03-17.

[PHS25] Iva Pezo, Allan Hanbury, and Moritz Staudinger. ipezotu at semeval-2025 task 7: Hybrid ensemble retrieval for multilingual fact-checking: Balancing efficiency and accuracy. In *Proceedings of the 19th International Workshop on Semantic Evaluation*, SemEval 2025, Vienna, Austria, July 2025.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[PSM^{+}23] Matúš Pikuliak, Ivan Srba, Robert Moro, Timo Hromadka, Timotej Smoleň, Martin Melišek, Ivan Vykopal, Jakub Simko, Juraj Podroužek, and Maria Bielikova. Multilingual Previously Fact-Checked Claim Retrieval. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16477–16500, Singapore, December 2023. Association for Computational Linguistics.

[PZ24] Rrubaa Panchendrarajan and Arkaitz Zubiaga. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066, June 2024.

[RSR^{+}19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[RWJ^{+}94] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at trec-3. In Donna K. Harman, editor, *TREC*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST), 1994.

[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.

[SCZ+21] Qiang Sheng, Juan Cao, Xueyao Zhang, Xirong Li, and Lei Zhong. Article reranking by memory-enhanced key sentence matching for detecting previously fact-checked claims. 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 5468–5481, Online, August 2021. Association for Computational Linguistics.

[She20] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 03 2020.

[SJ88] Karen Sparck Jones. *A statistical interpretation of term specificity and its application in retrieval*, page 132–142. Taylor Graham Publishing, GBR, 1988.

[SL65] G. Salton and M. E. Lesk. The smart automatic document retrieval systems—an illustration. *Commun. ACM*, 8(6):391–398, June 1965.

[SLG+24] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15933–15946, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[SMY+25] Premtim Sahitaj, Iffat Maab, Junichi Yamagishi, Jawan Kolanowski, Sebastian Möller, and Vera Schmitt. Towards automated fact-checking of real-world claims: Exploring task formulation and assessment with llms, 02 2025.

[Tra25] DeepL Translator. Deepl. https://www.deepl.com/en/translator, 2025. Accessed: 2025-03-23.

[TRR+21] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663, 2021.

[TUR50] A. M. TURING. I.—computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[VL19] Nguyen Vo and Kyumin Lee. Learning from fact-checkers: Analysis and generation of fact-checking language. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 335–344, New York, NY, USA, 2019. Association for Computing Machinery.

[VL20] Nguyen Vo and Kyumin Lee. Where are the facts? searching for fact-checked information to alleviate the spread of fake news. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7717–7731, Online, November 2020. Association for Computational Linguistics.

[Voo05] Ellen Voorhees. Overview of trec 2004, 2005-08-01 2005.

[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.

[WC03] Steven Walczak and Narciso Cerpa. Artificial neural networks. In Robert A. Meyers, editor, *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 631–645. Academic Press, New York, third edition edition, 2003.

[YNL21] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: BERT and beyond. In Greg Kondrak, Kalina Bontcheva, and Dan Gillick, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online, June 2021. Association for Computational Linguistics.