République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département de Génie Industriel

End-of-study project dissertation

for obtaining the State Engineer's degree in Industrial engineering

Option : Data Science and Artificial Intelligence (DSIA)

FactCheckBureau: Build Your Own Fact-Check Analysis Pipeline

SAADI Brahim & ELFRAIHI Mohammed Younes

Under the supervision of	Mr. ZOUAGHI Iskander	ENP
	Mrs. IOANA Manolescu	LIX
	Mrs. OANA Balalau	LIX

Presented and publicly defended on (19/12/2024)

Composition of the jury:

President:	Mr. ABBACI Ayoub	ENP
Examiner:	Mrs. BELDJOUDI Samia	ENP
Supervisor:	Mr. ZOUAGHI Iskander	ENP

ENP 2024

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département de Génie Industriel

End-of-study project dissertation

for obtaining the State Engineer's degree in Industrial engineering

Option : Data Science and Artificial Intelligence (DSIA)

FactCheckBureau: Build Your Own Fact-Check Analysis Pipeline

SAADI Brahim & ELFRAIHI Mohammed Younes

Under the supervision of	Mr. ZOUAGHI Iskander	ENP
	Mrs. IOANA Manolescu	LIX
	Mrs. OANA Balalau	LIX

Presented and publicly defended on (19/12/2024)

Composition of the jury:

President:	Mr. ABBACI Ayoub	ENP
Examiner:	Mrs. BELDJOUDI Samia	ENP
Supervisor:	Mr. ZOUAGHI Iskander	ENP

ENP 2024

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Département de Génie Industriel

Mémoire de Fin d'Études

Présenté pour l'obtention du diplôme d'Ingénieur d'État en Génie Industriel

Option : Data science et Intelligence Artificielle (DSIA)

FactCheckBureau : Construisez votre propre pipeline d'analyse de vérification des faits

---

SAADI Brahim & ELFRAIHI Mohammed Younes

Sous la supervision de	M. ZOUAGHI Iskander	ENP
	Mme IOANA Manolescu	LIX
	Mme OANA Balalau	LIX

Présenté et soutenu publiquement le (19/12/2024)

Composition du jury :

Président :	M. ABBACI Ayoub	ENP
Examinateur :	Mme BELDJOUDI Samia	ENP
Directeur :	M. ZOUAGHI Iskander	ENP

ENP 2024

Acknowledgments

We would like to begin by expressing our deepest gratitude to our academic supervisor, Mr. ***Iskander Zouaghi****, whose unwavering support and guidance made this internship possible in the first place. His extensive experience and thoughtful advice helped us strive to produce our best work throughout this project. No matter how much we thank him, we feel that we cannot give him the full merit he deserves for his constant encouragement and expert feedback.*

A very special thanks to Mrs. ***Ioana Manolescu*** *and Mrs.* ***Oana Balalau****, our internship supervisors from the LIX Laboratory. The expertise and attention to detail that Ioana brought to the project as a senior reasearcher were invaluable. Her meticulous approach and deep insights into data management and systems shaped our work and pushed us to aim higher. Oana, with her profound knowledge in Natural Language Processing, was not only an extraordinary mentor but also one of the nicest and most down-to-earth people we've ever had the pleasure to meet. Her discipline, work ethic, and unwavering support have made a lasting impact on us, and she remains a model of professionalism to follow.*

We are also immensely grateful to Mrs. ***Garima Gaur****, a post-doctoral researcher, whose dedication and involvement with us made her feel like a third supervisor. She always made time to discuss our ideas, encouraged us to explore new directions, and reassured us that she was always available for any assistance. Her feedback was always insightful, and she never missed any of our meetings, making her a key contributor to the progress of our work.*

Our utmost respect and thanks go to Mr. ***Théo Galizzi****, the most competent engineer we have worked with so far. His technical skills, along with his friendly and supportive nature, helped us integrate seamlessly into the project. He was always willing to offer advice and assistance, which made a significant difference in our work.*

We would also like to thank Mr. ***Pablo Bertaud-Velten****, a previous intern we had the chance to work with briefly. Pablo was responsible for data collection before us, and he made sure everything was perfectly clear and well-organized before his departure, ensuring that our transition into the project was smooth.*

Special thanks to Mr. ***Samuel da Silva Guimarães****, Mr.* ***Salim Chouaki****, Mrs.* ***Asmaa Elfraihi****, and Mr.* ***Muhammad Ghufran Khan****, PhD students at the LIX Laboratory, whose camaraderie and helpful discussions were invaluable. Their advice, suggestions, and friendship greatly enhanced our time at the laboratory. We are also grateful to Mr.* ***Zhang Kun*** *for his input and assistance with NLP-related tasks, always being available to help despite his busy schedule.*

We extend our sincere thanks to all the other members of the ***CEDAR team*** *for their kindness and for welcoming us into the group. We also appreciate the support of all the other members and staff of the* ***LIX Laboratory*** *and* ***École Polytechnique****, who contributed to making our internship such a positive and enriching experience.*

We are also deeply grateful to the jury members Mrs. ***Beldjoudi Samia*** *and Mr.*

***Ayoub Abbaci**, whose time, expertise, and valuable feedback on our work are greatly appreciated. Their contribution to improving the quality of our project cannot be overstated, and we are honored by the time and effort they have dedicated to evaluating our work.*

*We would also like to express our deep gratitude to the faculty of the **Department of Industrial Engineering at the École Nationale Polytechnique**, for the high-quality education and professional guidance they provided us throughout our academic journey. Their expertise, dedication, and commitment to our success have been instrumental in shaping us as engineers.*

Finally, we would like to thank our classmates, both from the three years "en spécialité" and the two years "en classes préparatoires". The unforgettable memories, the mutual support, and the friendship we have shared over the years have helped shape us into who we are today. We are grateful for every moment and for the bonds that will last a lifetime.

Dedication

“

To my parents, whose unconditional love, sacrifices, and wisdom have allowed me to complete this project. Your unwavering support has always been my greatest source of strength and inspiration.,

To my sister, for your love, your unfailing support, and your confidence in me throughout this journey.,

To my partner, Younes, for your collaboration, support, and commitment throughout this project. Thank you for your teamwork and determination,

To my friends, for your support, encouragement, and the shared moments that made this journey more enjoyable,

To all those dear to me, to all of you, thank you.

”

- Brahim

“

To my parents, for your boundless love, unwavering support, and countless sacrifices. Your wisdom and guidance have been my foundation throughout this journey.,

To all those who have crossed my path and had an impact on my life, no matter how big or small. Your influence has helped shape me into who I am today.,

To my partner, Brahim, for your collaboration, dedication, and support throughout this project. Your hard work and shared commitment have made all the difference.,

To my friends, for the laughter, encouragement, and shared experiences that made this journey more enjoyable and fulfilling.,

To each and every one of you, thank you.

”

- Younes

ملخص

تستكشف هذه الأطروحة تطوير وتقييم أنظمة التحقق الآلي من الحقائق، مع التركيز على مطابقة الادعاءات والتغريدات مع مقالات التحقق من الحقائق. نقوم بتقييم طرق الاسترجاع وإعادة الترتيب، مثل خوارزمية BM25 ونموذج SBERT.

تشمل المساهمات الرئيسية:

- **التشابه على مستوى الجمل:** نهج جديد لإعادة الترتيب باستخدام SBERT يعزز دقة مطابقة التغريدات مع المقالات.
- **تحليل خاص باللغة:** يسلط التحليل المقارن للادعاءات والتغريدات باللغة الإنجليزية والفرنسية الضوء على الحاجة إلى نماذج خاصة بكل لغة.
- **منصة تطبيق ويب FactCheckBureau** : مصمم لمساعدة الباحثين والصحفيين على تطوير أنظمة دقيقة لمطابقة الادعاءات والتحقق من الحقائق.

تكشف تجاربنا عن نقاط القوة والضعف في الطرق المختلفة. بينما يشكل BM25 أساسًا قويًا، يعزز SBERT مع دقة على مستوى الجمل من تحسين الدقة. كما نستكشف تقنيات إثراء التغريدات مثل OCR وإضافة تسميات الصور لتحسين تمثيل التغريدات.

تساهم هذه الدراسة في تقدم تقنيات التحقق الآلي من الحقائق، حيث توفر أدوات ورؤى لمكافحة المعلومات المضللة. تُمكّن منصةFactCheckBureau من التحقق الفعّال من الادعاءات، مما يعزز نشر المعلومات الدقيقة عبر الإنترنت.

الكلمات المفتاحية: التحقق من الحقائق، المعلومات المضللة، الأنظمة الآلية، مطابقة التغريدات مع المقالات، استرجاع المعلومات، BM25، SBERT، التشابه على مستوى الجمل، إثراء التغريدات، الإنجليزية والفرنسية، FactCheckBureau.

Résumé

Cette thèse explore le développement et l'évaluation des systèmes de vérification automatique des faits, en se concentrant sur la correspondance entre les affirmations et les tweets avec les articles de vérification des faits. Nous évaluons des méthodes de récupération et de re-classement, telles que l'algorithme BM25 et le modèle SBERT.

Les principales contributions incluent :

- **Similarité au niveau des phrases :** Une approche novatrice pour le re-classement avec SBERT améliore la précision de la correspondance tweet-article.
- **Analyse spécifique à la langue :** Une analyse comparative des affirmations et des tweets en anglais et en français souligne la nécessité de modèles spécifiques à chaque langue.
- **Plateforme FactCheckBureau :** Une application web conçue pour aider les chercheurs et les journalistes à développer des systèmes précis de correspondance entre les affirmations et la vérification des faits.

Nos expériences révèlent les forces et les limites de différentes méthodes. Bien que BM25 serve de référence robuste, SBERT avec une granularité au niveau des phrases améliore la précision. Nous explorons également des techniques d'enrichissement des tweets, telles que l'OCR et la génération de légendes d'images, pour améliorer la représentation des tweets.

Cette recherche fait progresser la vérification automatique des faits en offrant des outils et des perspectives pour lutter contre la désinformation. La plateforme FactCheckBureau permet une vérification efficace des affirmations, promouvant une information en ligne plus précise.

Mots-clés : vérification des faits, désinformation, systèmes automatisés, correspondance tweet-article, récupération d'information, BM25, SBERT, similarité au niveau des phrases, enrichissement des tweets, anglais et français, FactCheckBureau.

Abstract

This thesis explores the development and evaluation of automated fact-checking systems, focusing on matching claims and tweets to fact-checking articles. We assess retrieval and re-ranking methods, such as the BM25 algorithm and SBERT model.

Key contributions include:

- **Sentence-Level Similarity:** A novel approach for SBERT re-ranking improves accuracy in tweet-article matching.
- **Language-Specific Analysis:** Comparative analysis of English and French claims highlights the need for language-specific models.
- **FactCheckBureau Platform:** A web application designed to help researchers and journalists develop accurate claim-fact check matching systems.

Our experiments reveal the strengths and limitations of various methods. While BM25 serves as a robust baseline, SBERT with sentence-level granularity enhances precision. We also explore tweet enrichment techniques like OCR and image captioning to improve tweet representation.

This research advances automated fact-checking, offering tools and insights to combat misinformation. The FactCheckBureau platform enables effective claim verification, promoting accurate information online.

Keywords :fact-checking, misinformation, automated systems, tweet-article matching, information retrieval, BM25, SBERT, sentence-level similarity, tweet enrichment, English and French, FactCheckBureau.

Contents

List of Figures

List of Tables

Abbreviations & Acronyms

ALBERT	*A Lite BERT*
API	*Application Programming Interface*
BERT	*Bidirectional Encoder Representations from Transformers*
BM25	*Best Matching 25*
BoW	*Bag-of-Words*
CamemBERT	*French BERT-based Language Model*
CNN	*Convolutional Neural Network*
CPU	*Central Processing Unit*
demoji	*Emoji Handling Library*
DistillBERT	*Distilled BERT*
duckdb	*In-Process SQL OLAP Database Management System*
easyocr	*Optical Character Recognition Library*
FAISS	*Facebook AI Similarity Search*
GloVe	*Global Vectors for Word Representation*
GPT	*Generative Pre-trained Transformer*
GPT-2	*Generative Pre-trained Transformer 2*
GPT-3	*Generative Pre-trained Transformer 3*

HTML	*HyperText Markup Language*
Hugging Face	*Hugging Face Transformers Library*
IFCN	*International Fact-Checking Network*
IVF	*Inverted File*
IVF-PQ	*Inverted File with Product Quantization*
LangDetect	*Language Detection Tool*
Longformer	*Transformer Model for Long Document Processing*
LSTM	*Long Short-Term Memory*
MAP	*Mean Average Precision*
MLM	*Masked Language Model*
MRR	*Mean Reciprocal Rank*
NDCG	*Normalized Discounted Cumulative Gain*
NER	*Named Entity Recognition*
NLI	*Natural Language Inference*
NLTK	*Natural Language Toolkit*
NSP	*Next Sentence Prediction*
num2words	*Number-to-Words Conversion Library*
OCR	*Optical Character Recognition*
OLAP	*Online Analytical Processing*
PIL	*Python Imaging Library*
plotly	*Interactive Visualization Library*

PQ	*Product Quantization*
rank-bm25	*BM25 Implementation Library*
RNN	*Recurrent Neural Network*
RoBERTa	*A Robustly Optimized BERT Pretraining Approach*
SBERT	*Sentence-BERT*
SQL	*Structured Query Language*
T5	*Text-to-Text Transfer Transformer*
TF-IDF	*Term Frequency-Inverse Document Frequency*
unicodedata	*Unicode Character Normalization Tool*
Word2Vec	*Word to Vector*
XLM-R	*Cross-lingual Language Model - RoBERTa*

Introduction

The Problem of Misinformation

In today's digital age, information spreads rapidly through online platforms and social media, connecting people like never before. However, this easy access to information has also led to the spread of misinformation—false or misleading information that can be shared intentionally or unintentionally.
Misinformation can take many forms, such as fake news, rumors, conspiracy theories, and edited media. It can come from various sources, including individuals, groups, or even state-sponsored actors, and it spreads through numerous channels like social media, news websites, blogs, and messaging apps.
The effects of misinformation are serious and widespread. It can distort how people perceive events, damage trust in institutions, deepen political divides, and even incite violence or discrimination. Misinformation has also been linked to negative health outcomes, such as reluctance to get vaccinated and the spread of harmful health practices.
Addressing misinformation is challenging. The sheer volume of online information makes it hard to identify and verify all misleading claims. Moreover, the speed at which misinformation spreads, often boosted by social media algorithms, makes it difficult to control and correct. Not only false news resurface, but they also spread six times faster than correct news[21]. False beliefs can persist even after being debunked, worsening the problem.
Fighting misinformation requires a variety of strategies, including media literacy education, critical thinking skills, and strong fact-checking initiatives. Automated fact-checking systems that use natural language processing and machine learning are emerging as helpful tools to support human fact-checkers. These systems can potentially expand and speed up fact-checking efforts, providing timely and accurate information to the public.

Fact-Checking as a Solution

Fact-checking is the thorough process of verifying claims to ensure they are accurate and truthful. It has become a crucial defense against the spread of misinformation. Fact-checking organizations, whether independent or linked to news outlets, play a key role in this effort. By carefully investigating claims, consulting credible sources, and evaluating evidence, fact-checkers help provide the public with reliable information and expose false or misleading narratives.

Well-known fact-checking organizations like Agence France-Presse (AFP), Snopes, PolitiFact, and Full Fact have dedicated teams of researchers and journalists. These teams specialize in verifying claims across a wide range of topics, including politics, health, science, and social issues. They follow strict journalistic standards and methodologies to ensure their fact-checking process is transparent and rigorous. They often publish their findings in detailed reports, articles, or social media posts, making the information accessible and easy to understand.

However, traditional fact-checking is manual and has its limitations. The sheer amount of information online and the speed at which misinformation spreads make it almost impossible for human fact-checkers to keep up with every claim in a timely manner. The resources needed for thorough investigations, including research, interviews, and analysis, can be substantial, making it hard to scale manual fact-checking efforts.

To overcome these challenges and extend the reach and impact of fact-checking, there's growing interest in developing automated fact-checking systems. These systems use natural language processing (NLP) and machine learning techniques to automate parts of the fact-checking process, such as detecting claims, retrieving evidence, and even verifying claims to some extent. By automating repetitive and time-consuming tasks, these systems can greatly enhance the capabilities of human fact-checkers, allowing them to focus on more complex and nuanced investigations.

In this context, our research is centered on developing an automated system for matching claims with facts, a critical step in the fact-checking process. By using advanced NLP techniques like BM25 and SBERT, our system aims to quickly and accurately identify relevant fact-checking articles for any given claim, thus speeding up the fact-checking workflow and helping to combat misinformation.

Automated Fact-Checking

Given the limitations of manual fact-checking and the growing threat of misinformation, automated fact-checking systems have started to emerge. These systems use advanced natural language processing (NLP) and machine learning to make the verification process faster and more efficient. By automating various fact-checking tasks, these systems aim to support human fact-checkers and improve their ability to fight misinformation.

Key Tasks in Automated Fact-Checking:

- **Claim Detection:** This involves automatically spotting potential claims in large amounts of text. Machine learning models are trained to recognize linguistic patterns and context clues that signal a statement might need further investigation.
- **Evidence Retrieval:** This task focuses on finding relevant evidence that supports or disproves a claim. Automated systems can search through huge collections of text, including news articles, research papers, and fact-checking websites, to find pertinent information.
- **Claim Verification:** This is the toughest part of fact-checking. It not only involves finding relevant evidence but also assessing its credibility and determining its

impact on the truthfulness of the claim. While fully automating this process is complex, machine learning models can help human fact-checkers by identifying potential evidence sources, highlighting inconsistencies, and suggesting possible verdicts.

Challenges in Automated Fact-Checking[5][13]:

- **Language Ambiguity:** Natural language can be ambiguous, with words and phrases often having multiple meanings depending on the context. Automated systems must accurately interpret the intended meaning of claims and evidence to avoid mistakes.
- **Diversity of Sources:** Misinformation can come from various sources, each with its own style, format, and credibility. Automated systems need to handle this diversity and adapt to different types of text data.
- **Evolving Nature of Misinformation:** Misinformation tactics are always changing, with new forms and techniques appearing regularly. Automated systems must be flexible and able to learn from new patterns of deception.
- **Ethical Considerations:** The use of automated fact-checking brings up ethical issues, such as potential biases in algorithms, the risk of relying too much on automated systems, and the need for transparency and accountability in developing and using these tools.

Despite these challenges, automated fact-checking holds great promise for improving the efficiency and scale of fact-checking efforts. By automating time-consuming tasks, these systems can free up human fact-checkers to tackle more complex investigations, leading to a more thorough and timely response to the spread of misinformation.

Our Approach and Contributions

To tackle the challenge of claim-fact matching, we've taken a fresh approach that combines the strengths of two powerful tools: BM25, a well-known method for quickly finding potentially relevant documents, and SBERT, a cutting-edge model that understands the nuances of language and meaning. This two-pronged strategy allows us to efficiently sift through vast amounts of information while also ensuring that the articles we find are truly relevant to the claim at hand.

We're excited about the potential of this research, as it could make a real difference in the fight against misinformation. Here's what we believe our work brings to the table:

1. **A New Way to Match Claims and Facts:** Our combined BM25 and SBERT approach is a novel solution that goes beyond simple keyword matching. By understanding the meaning behind the words, we can find relevant fact checks even when they don't use the exact same language as the claim.
2. **Rigorous Testing:** We've put our system through its paces, testing it on a wide variety of claims and fact-checking articles from reliable sources. We've even included both English and French texts to see how well it works across different languages.

3. **Measuring Success:** We didn't just rely on one way to measure how well our system works. We've used multiple metrics to give a complete picture of its performance, including not just accuracy, but also how well it ranks the most relevant articles.

4. **Real-World Potential:** While not the focus of this thesis, we envision this technology eventually becoming part of a user-friendly fact-checking platform. This could empower not just professional fact-checkers, but everyday people as well, to quickly verify claims they encounter online.

In the following sections, we'll walk you through exactly how our system works, the data we used to test it, and the exciting results we achieved. We'll also discuss what we learned along the way and where we see this research heading in the future.

Chapter 1

Foundational Technologies

Introduction

In the development of automated fact-checking systems, foundational technologies play a critical role in enabling accurate, efficient, and scalable solutions. This chapter explores the key technologies that underlie modern natural language processing (NLP) and information retrieval models, which form the backbone of our system for claim-fact matching.

We delve into transformer-based models, such as BERT and SBERT, which have revolutionized the way machines understand language context and sentence similarity. The chapter also covers innovations like FAISS for efficient similarity search in large-scale datasets, and Longformers, designed to handle long documents, which are particularly useful in fact-checking. Additionally, we discuss the importance of asymmetric models and their ability to enhance document relevance understanding through datasets like MS MARCO.

By examining these foundational technologies, we highlight their contributions to improving the performance and scalability of automated fact-checking systems, setting the stage for further exploration of advanced models and methods.

1.1 Information Retrieval (IR) Fundamentals:

In the digital age, the vast amount of information available online presents both opportunities and challenges. While we have unprecedented access to knowledge, finding relevant and trustworthy information amidst the noise can be a daunting task. This is where Information Retrieval (IR) comes in. IR is a field of computer science dedicated to the science of searching for information within a document collection. It provides the tools and techniques necessary to efficiently locate and retrieve relevant documents based on a user's query or information need.

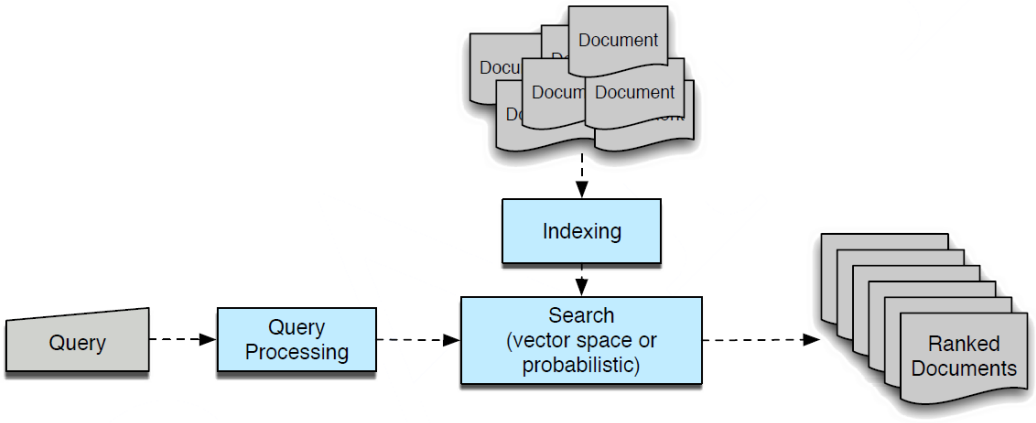

Figure 1.1: Overview of the Information Retrieval (IR) System [4]

1.1.1 Relevance in Information Retrieval

At the heart of IR lies the concept of relevance. A document is considered relevant if it satisfies the user's information need. However, defining and measuring relevance is not always straightforward, as it can be subjective and context-dependent. IR systems aim to automate this process by employing various algorithms and techniques to estimate the relevance of documents based on their content and the user's query.[9]

1.1.2 Document Representation

Before we can delve into retrieval models, it's crucial to understand how documents are represented in a way that computers can understand and process. This involves transforming unstructured text into a structured, numerical format that captures the essential information for retrieval.

1. **Bag-of-Words (BoW)**:
    - In the BoW model, a document is represented as an unordered collection of its unique words, disregarding grammar and word order. Essentially, it creates a "bag" of words where the order doesn't matter. Each document is then represented as a vector, where each element corresponds to a word in the vocabulary (the set of all unique words in the corpus), and the value of each element is the frequency of that word in the document.
    - While BoW is simple and computationally efficient, it ignores the context and relationships between words, which can limit its effectiveness in capturing the meaning of the text.
2. **Term Frequency-Inverse Document Frequency (TF-IDF):**
    - TF-IDF is an extension of the BoW model that addresses some of its limitations. It assigns weights to words based on their importance in a document and across the corpus. The weight of a term is directly proportional to the number of times it appears in the document but inversely proportional to the frequency of the word in the corpus. This means that words that are frequent in a document but rare across the corpus are given higher weights, as they are considered more informative.
    - While TF-IDF improves upon BoW by considering term importance, it still ignores word order and semantic relationships.

1.1.3 Term Frequency-Inverse Document Frequency (TF-IDF):

The Fundamentals of TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is an information retrieval algorithm that uses statistical methods to measure the importance of a keyword within a document in relation to a collection of documents.

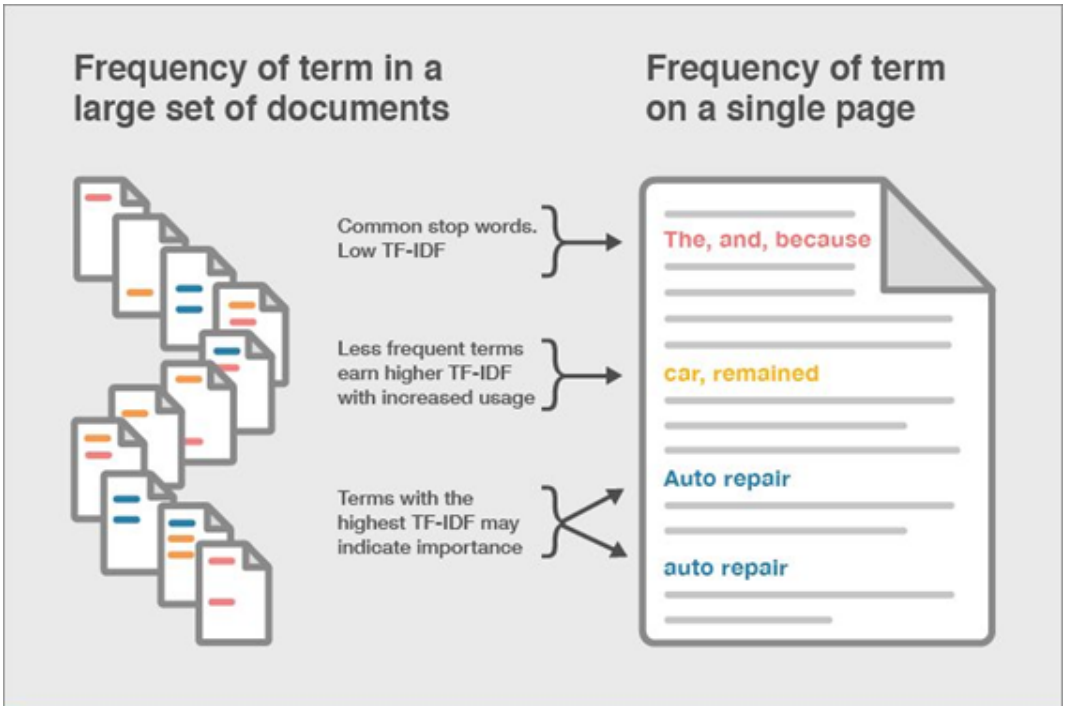

Figure 1.2: TF-IDF overview [10]

TF-IDF is composed of two parts: **Term Frequency (TF)** and **Inverse Document Frequency (IDF)**. These components measure different aspects, but they are multiplied together to obtain the final score that estimates the relevance of a word within a document.

- Term Frequency (TF): This component measures the occurrence of a specific keyword in a document. The more instances of the keyword in a document, the higher the TF value.

- Inverse Document Frequency (IDF): This component measures the proportion of documents in a collection that contain the keyword. The more frequently the keyword appears across documents, the lower its IDF score. This penalizes common words such as 'a', 'an', and 'is' that appear in many documents.

After calculating the TF and IDF components, the final TF-IDF score is obtained by multiplying them. This score captures the importance of a keyword by assigning it a higher value if it appears frequently in one document but rarely in others.

Search engines use TF-IDF scores to determine the relevance of a document to a user's keyword or query, ranking the documents and presenting the most relevant ones to the user.

Problems with TF-IDF

TF-IDF has two main issues that can be improved:

1. **Document Length Bias:**

   - Consider a query for the word “rabbit” in two documents. In Document A (1000 words), ”rabbit” appears ten times. In Document B (10 words), ”rabbit” appears once.
   - Traditional TF-IDF would give Document A a higher TF score (10) compared to Document B (1). However, Document B might be more relevant due to the higher concentration of the keyword.
   - To address this, the normalized variant of TF-IDF divides TF by the document length, providing a more balanced relevance score.

2. **Keyword Saturation:**

   - The linear relationship in TF-IDF suggests that more occurrences of a keyword continuously increase relevance. For example, if ”rabbit” appears 400 times in a document, it is not necessarily twice as relevant as a document with 200 occurrences.
   - The score increase from 2 to 4 occurrences should have a greater impact than the increase from 200 to 400 occurrences. This diminishing return effect is not captured by the traditional TF-IDF formula.

1.1.4 Best Matching 25 (BM25)

Best Matching 25 (BM25) is an algorithm designed to improve upon traditional TF-IDF by addressing the problems mentioned earlier.

Keyword Saturation

In traditional TF-IDF, the TF part grows linearly with the number of keyword occurrences. BM25 modifies this by introducing a new parameter in the TF equation:

$$\mathrm{TF}_{\mathrm{BM25}} = \frac{(\mathrm{TF}) \cdot (k+1)}{\mathrm{TF} + k} \tag{1.1}$$

The parameter k controls the contribution of each incremental keyword occurrence to the TF score. The impact of the first few occurrences is significant, but as the keyword appears more frequently, its contribution diminishes. Higher k values slow the growth of the TF score, addressing the keyword saturation issue.[2]

Document Length Normalization

BM25 also takes document length into account, enhancing the relevance of shorter documents with concentrated keywords. The term $|D|$ represents document length, and $\mathrm{avg}(D)$ is the average document length in the corpus:

$$\mathrm{TF}_{\mathrm{BM25}} = \frac{\mathrm{TF} \cdot (k+1)}{\mathrm{TF} + k \cdot \left(1 - b + b \cdot \frac{|D|}{\mathrm{avg}(D)}\right)} \tag{1.2}$$

The parameter b controls the importance of document length. Setting b to 0 ignores document length, while $b = 1$ gives it full importance. This normalization ensures that shorter documents reach the saturation point faster than longer ones.

IDF Component

The IDF part of BM25 is slightly different from TF-IDF:

$$\text{IDF}_{\text{BM25}} = \log\left(\frac{N - \text{DF} + 0.5}{\text{DF} + 0.5}\right) \tag{1.3}$$

Where N is the total number of documents, and DF is the number of documents containing the keyword. This adjustment prevents the IDF value from becoming negative if the keyword appears in more than half of the documents. To avoid negative values entirely, a scalar of 1 is often added:[2]

$$\text{IDF}_{\text{BM25}} = \log\left(\frac{N - \text{DF} + 0.5}{\text{DF} + 0.5} + 1\right) \tag{1.4}$$

Final BM25 Equation

The final BM25 equation for scoring a keyword in a document is:

$$\text{BM25} = \sum_{i=1}^{n} \text{IDF}_i \cdot \frac{\text{TF}_i \cdot (k+1)}{\text{TF}_i + k \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avg}(D)}\right)} \tag{1.5}$$

Parameters of BM25

BM25 has two tunable parameters: k and b. These values can be adjusted to fit specific use cases:

- - **k**: Typically ranges from 0.5 to 2. A higher k is suitable for longer documents where keywords may appear frequently without necessarily being relevant.
- - **b**: Typically ranges from 0.3 to 0.9. A lower b is better for corpora where document length doesn't affect keyword relevance, while a higher b penalizes keyword spamming.

Practical Values

In practice, $k = 1.2$ and $b = 0.75$ often yield good results across various corpora. However, it's essential to experiment with these values to find the best fit for your specific use case, following the "no free lunch" theory, which implies no universally optimal parameter settings.

- For collections of long documents, a higher k value prevents reaching the saturation point too quickly. - For collections where document length is significant, adjust b to reflect the relevance of longer or shorter documents. For scientific documents, a lower b might be ideal, while for subjective content, a higher b could help manage keyword spamming.

Conclusion

BM25 addresses the limitations of traditional TF-IDF by incorporating document length normalization and a non-linear TF component, resulting in a more accurate and flexible relevance scoring system for information retrieval.

1.2 Deep Learning for Natural Language Processing (NLP): Extracting Meaning from Words

Deep learning, a subfield of machine learning, has revolutionized the field of Natural Language Processing (NLP). This approach utilizes artificial neural networks with multiple layers (hence ”deep”) to learn representations of data, allowing computers to understand and process human language in ways previously unattainable.

1.2.1 Overview of Deep Learning and Its Applications in NLP

Deep learning models are inspired by the structure and function of the human brain, using interconnected nodes (neurons) organized in layers to learn complex patterns from data.

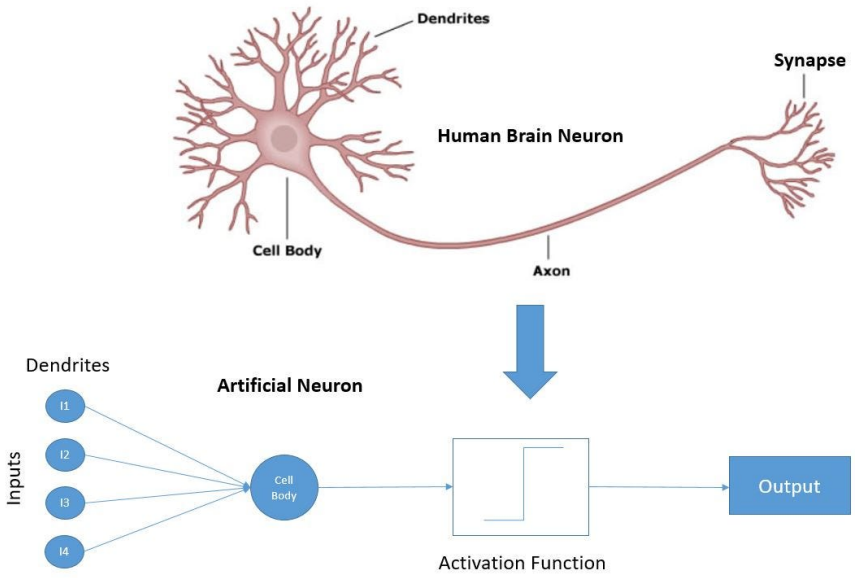

Figure 1.3: Artificial and biological neuron analogy [16]

This ability to extract meaning from text has opened up a myriad of applications in NLP:

- **Machine Translation:** Deep learning models have significantly improved the quality of machine translation, enabling more accurate and fluent translations between languages.
- **Sentiment Analysis:** Deep learning can be used to determine the emotional tone of a piece of text, whether it’s positive, negative, or neutral. This is valuable for analyzing customer reviews, social media posts, and other forms of user-generated content.

- **Question Answering:** Deep learning models can be trained to understand questions posed in natural language and provide accurate answers based on information from a knowledge base or document corpus.

- **Text Summarization:** These models can condense long articles or documents into concise summaries, saving time and effort for readers.

- **Chatbots and Conversational AI:** Deep learning is used to power chatbots and virtual assistants, enabling them to understand and respond to user queries in a natural and engaging way.

In the context of claim-fact matching, deep learning models can be used to understand the semantic meaning of claims and fact-check articles, enabling more accurate and efficient matching than traditional methods based on keyword matching or rule-based systems.

1.2.2 Word Embeddings: The Building Blocks of Meaning

A fundamental concept in deep learning for NLP is word embeddings. These are dense vector representations of words that capture their meanings and relationships in a continuous vector space. Similar words have similar vectors, allowing models to understand semantic relationships such as synonyms, antonyms, and analogies.

Two popular methods for generating word embeddings are:

- **Word2Vec:** This model learns word embeddings by training a neural network to predict a word based on its surrounding context (or vice versa).

- **GloVe(Global Vectors for Word Representation)**: Developed by Stanford, GloVe constructs word vectors by factoring in the global word-word co-occurrence matrix, which captures how frequently words co-occur in a corpus. It combines the advantages of global matrix factorization and local context window methods.

Word embeddings play a crucial role in many NLP tasks, including claim-fact matching. They provide a way to represent the meaning of words in a way that can be easily processed by machine learning algorithms.

1.2.3 Neural Networks for NLP: RNNs, LSTMs, and CNNs

Several types of neural networks have been used in NLP:

- **Recurrent Neural Networks (RNNs):** These networks are designed to process sequential data, like text, by maintaining a hidden state that captures information from previous time steps. However, RNNs can struggle to capture long-range dependencies due to the vanishing gradient problem.[12]

- **Long Short-Term Memory (LSTM) Networks:** LSTMs are a type of RNN that addresses the vanishing gradient problem through the use of gates that control the flow of information. This enables them to capture long-range dependencies more effectively.[12]
- **Convolutional Neural Networks (CNNs):** While primarily used for image processing, CNNs have also been applied to NLP tasks. They use filters to extract local features from text, which can be combined to form higher-level representations.[12]

While these traditional neural network architectures have achieved significant success in NLP, they have limitations in capturing complex linguistic phenomena and long-range dependencies. These limitations paved the way for the development of the Transformer architecture, which we will discuss in the next section.

1.3 The Transformer Architecture

1.3.1 Overview of the Architecture and Its Significance

The Transformer architecture, introduced by Vaswani et al. in the paper "Attention Is All You Need"[19], has revolutionized the field of Natural Language Processing (NLP). Unlike previous architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs), Transformers do not rely on sequential data processing, which allows for more efficient parallelization during training. This makes Transformers highly effective for a wide range of NLP tasks, including translation, summarization, and question answering.

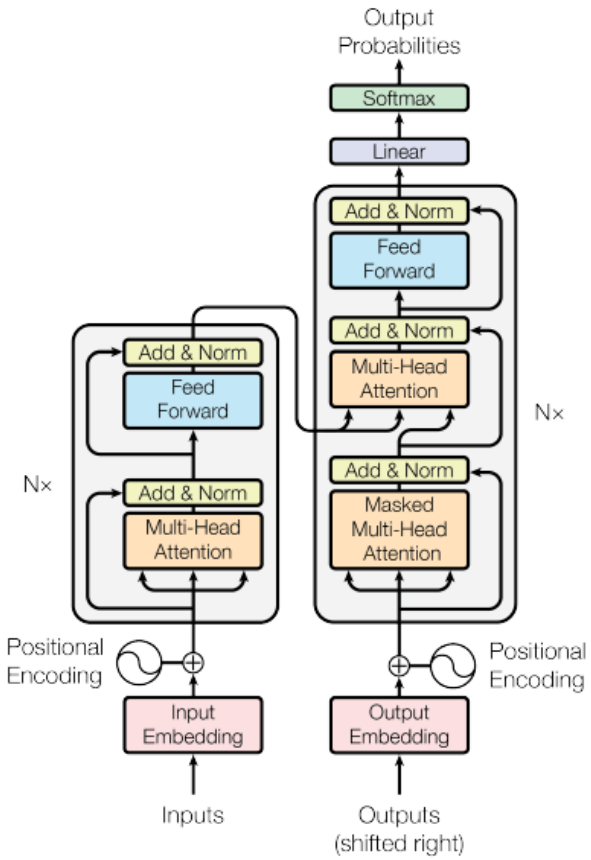

Figure 1.4: High-Level Transformer Architecture [19]

1.3.2 Self-Attention Mechanism Explained in Detail

At the core of the Transformer architecture is the self-attention mechanism, which allows the model to weigh the importance of different words in a sentence relative to each other. This mechanism enables the model to capture dependencies regardless of their distance in the sequence.[19]

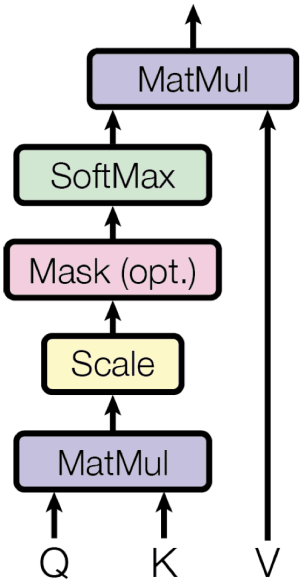

Figure 1.5: Self-Attention Mechanism [19]

The self-attention mechanism operates as follows:

1. **Input Representation:** Each input token is transformed into three vectors: Query (Q), Key (K), and Value (V).

2. **Attention Scores Calculation:** Attention scores are computed by taking the dot product of the Query vector with all Key vectors. These scores determine how much focus the model should place on other parts of the input when encoding a particular part.

$$\text{Attention Score}(Q, K) = Q \cdot K^T \tag{1.6}$$

3. **Scaling:** The attention scores are scaled by the square root of the dimension of the Key vectors ($\sqrt{d_k}$). This prevents the scores from growing too large and ensures more stable gradients.

$$\text{Scaled Attention Score}(Q, K) = \frac{Q \cdot K^T}{\sqrt{d_k}} \tag{1.7}$$

4. **Softmax Application:** A softmax function is applied to the scaled scores to convert them into probabilities. This normalizes the scores so that they sum to one, indicating the weight of each word.

$$\text{Attention Probability}(Q, K) = \text{softmax}(\text{Scaled Attention Score}(Q, K)) \tag{1.8}$$

5. **Weighted Sum:** The final output is obtained by computing a weighted sum of the Value vectors, using the attention probabilities as weights.

$$\text{Attention Output}(Q, K, V) = \text{Attention Probability}(Q, K) \cdot V \tag{1.9}$$

The self-attention mechanism is mathematically defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.10}$$

1.3.3 Multi-Head Attention and Positional Encoding

Multi-Head Attention:

To enhance the model's ability to focus on different positions, the Transformer employs multi-head attention. Instead of performing a single attention function, the model uses multiple attention heads, each with its own set of Query, Key, and Value weight matrices. The outputs of these attention heads are then concatenated and linearly transformed to form the final output (Figure 1.6).

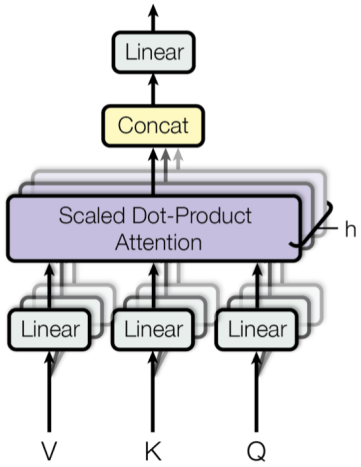

Figure 1.6: Multi-Head Attention Mechanism
[19]

This approach allows the model to capture various aspects of relationships between words, providing a richer representation of the input data.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{1.11}$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$.

Positional Encoding:

Since Transformers do not inherently process sequences in order, positional encodings are added to the input embeddings to provide information about the position of each word in the sequence. These encodings are vectors of the same dimension as the input embeddings, and they are added to the input embeddings at the bottom of the encoder and decoder stacks.

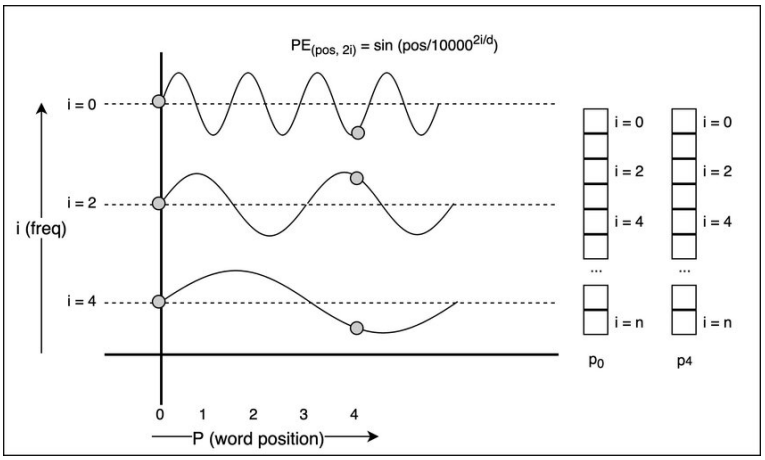

Figure 1.7: Working of positional encoding in Transformer Neural Networks. [6]

The positional encoding vectors are defined using sine and cosine functions of different frequencies:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right) \tag{1.12}$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\mathrm{model}}}}\right) \tag{1.13}$$

where pos is the position and i is the dimension.

1.3.4 Encoder-Decoder Structure

The Transformer architecture consists of an encoder and a decoder, each composed of multiple layers (Figure 1.8 and Figure 1.9).

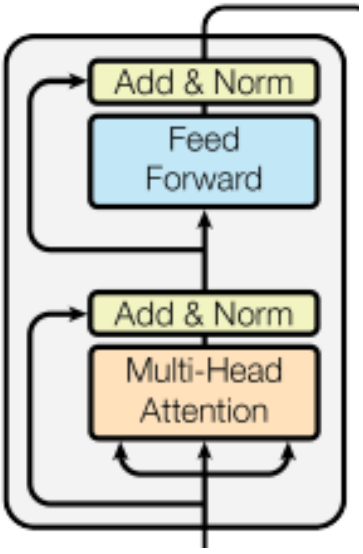

Figure 1.8: Encoder Structure
[19]

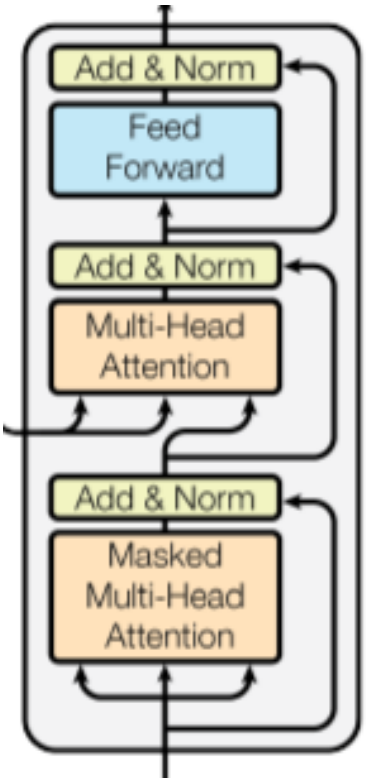

Figure 1.9: Decoder Structure
[19]

Encoder:

- Each encoder layer has two main components: a multi-head self-attention mechanism and a position-wise fully connected feed-forward network.
- Layer normalization and residual connections are employed to stabilize and enhance the training process.

$$\text{EncoderLayer}(x) = \text{LayerNorm}(x+\text{FeedForward}(\text{LayerNorm}(x+\text{MultiHeadAttention}(x)))) \tag{1.14}$$

Decoder:

- The decoder layers also consist of a multi-head self-attention mechanism and a feed-forward network, but they include a third sub-layer for multi-head attention over the encoder's output.
- Similar to the encoder, the decoder uses layer normalization and residual connections.

$$\begin{aligned}\text{DecoderLayer}(x) = \text{LayerNorm}(x+ \\ \text{FeedForward}(\text{LayerNorm}(x+ \\ \text{MultiHeadAttention}(\text{LayerNorm}(x+ \\ \text{MaskedMultiHeadAttention}(x))))))\end{aligned} \tag{1.15}$$

Advantages of Transformers

- **Parallelization:** Unlike RNNs and LSTMs, Transformers allow for parallel processing of data, leading to significantly faster training times.
- **Long-Range Dependencies:** Self-attention mechanisms enable Transformers to capture long-range dependencies more effectively than RNN-based models.
- **Scalability:** Transformers can be scaled up easily, which has been demonstrated by large models like BERT, GPT-3, and T5.
- **Versatility:** Transformers have proven to be highly effective across various NLP tasks, setting new benchmarks in many applications.

1.4 Transformer-Based Models

1.4.1 Encoder Models (BERT)

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking transformer-based model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers [3] [7]. This approach enables BERT to develop a rich understanding of language, leading to state-of-the-art performance on a wide variety of natural language processing (NLP) tasks.

Architecture:

- **Bidirectional Training:** BERT is trained on unlabeled text by jointly conditioning on both left and right context in all layers. This allows it to understand the context of a word based on its entire surrounding text, rather than just the words before or after it.
- **Layers:** BERT is composed of multiple layers of Transformer blocks. The base model consists of 12 layers, each with 768 hidden units and 12 self-attention heads.
- **Input:** BERT accepts a pair of sentences as input. Special tokens are used to mark the beginning (‘[CLS]‘) and end (‘[SEP]‘) of each sentence. The ‘[CLS]‘ token is used for classification tasks, as it represents the aggregated representation of the input sentence pair.

Training Objectives:

BERT is pre-trained on two unsupervised tasks:

- **Masked Language Modeling (MLM):** A percentage of input tokens are randomly masked. The model is trained to predict the original (masked) words based only on their context. This helps BERT learn deep bidirectional representations of words and phrases.
- **Next Sentence Prediction (NSP):** Given two sentences, BERT predicts whether the second sentence is the actual next sentence in the original text. This helps BERT learn relationships between sentences, which is useful for tasks like question answering and natural language inference.

Strengths:

- **Contextual Understanding:** BERT's bidirectional training and masked language modeling enable it to capture deep contextual relationships between words, leading to better representations and performance.
- **Transfer Learning:** The pre-trained BERT model can be easily fine-tuned on a wide range of downstream tasks with smaller task-specific datasets, saving time and resources.

Limitations:

- **Resource Intensive:** Training and fine-tuning BERT, especially the larger versions, requires significant computational resources and time.
- **Inference Time:** Larger BERT models can be slow for real-time applications, where quick responses are required.
- **Bias:** Like other language models, BERT can inherit biases present in its training data, potentially leading to biased or unfair predictions.

1.4.2 Sentence Embedding Models (SBERT)

SBERT (Sentence-BERT) is an adaptation of BERT that provides dense vector representations for sentences, making it effective for tasks requiring sentence or text similarity computations.[15]

Architecture:[15]

- **Sentence Transformers:** SBERT uses BERT to generate embeddings and then fine-tunes the network with a siamese or triplet network structure to derive fixed-size embeddings.

- **Pooling Layer:** Outputs are averaged or max-pooled to obtain a fixed-size sentence vector.

Training:

Triplet Loss: Fine-tunes BERT on sentence pairs with a triplet loss objective to bring similar sentences closer in vector space while pushing dissimilar ones apart.[15]

NLI Data: Often trained on Natural Language Inference (NLI) datasets to learn semantic similarity.

Strengths:

Efficient Similarity Computation: Produces sentence embeddings that can be quickly compared using cosine similarity or other metrics.

Versatile: Useful for a wide range of tasks like semantic textual similarity, clustering, and paraphrase mining.

Limitations:

- **Pre-training Dependency:** Depends heavily on the pre-trained BERT model, inheriting its computational resource requirements.
- **Sentence Length:** Performance can degrade with very long sentences.

1.4.3 Other Relevant Models

ALBERT (A Lite BERT)

ALBERT is a more efficient version of BERT designed to reduce memory consumption and increase the training speed.[11]

Architecture:

- **Parameter Sharing:** Shares parameters across layers to reduce the model size.
- **Factorized Embedding Parameterization:** Reduces the number of parameters in the embedding layer.

Training:

- **Similar to BERT:** Uses MLM and sentence order prediction tasks for pre-training.
- **Enhanced Training Techniques:** Incorporates techniques like inter-sentence coherence loss.

Strengths:

- **Efficiency:** Smaller model size and faster training time compared to BERT.
- **Performance:** Comparable performance to BERT on many NLP tasks.

Limitations:

- **Complexity in Tuning:** The parameter-sharing approach can make the model more complex to fine-tune for specific tasks.

DistilBERT

DistilBERT is a smaller, faster, and lighter version of BERT, obtained through the process of knowledge distillation.[17]

Architecture:

- **Compressed Model:** Retains 97% of BERT's language understanding while being 60% faster and 40% smaller.[17]
- **Simplified:** Removes the token-type embeddings and pooler.

Training:

Knowledge Distillation: Trained to reproduce the behavior of a larger BERT model by mimicking its logits.

Strengths:

- **Speed:** Faster inference due to reduced size.
- **Resource Efficient:** Less computational resources required for training and deployment.

Limitations:

- Reduced Capacity: May not perform as well as BERT on complex tasks requiring deep understanding.

Decoder Models

Decoder models are Transformer architectures that focus on generating text sequences. They are often used in tasks like machine translation, text generation, and summarization.

GPT (Generative Pre-trained Transformer)

GPT is a Transformer-based decoder model designed for generating coherent and contextually relevant text.

Architecture:

- **Unidirectional Training:** GPT is trained in a left-to-right fashion, meaning each word is generated based on the previous words in the sequence.[23]
- **Layers:** GPT-2 and GPT-3 use a large number of Transformer blocks, with GPT-3 having up to 175 billion parameters.[23]
- **Input:** GPT uses a standard Transformer decoder architecture with masked self-attention to prevent the model from seeing future tokens.[23]

Training:

- **Language Modeling:** Trained on a large corpus of text using a language modeling objective to predict the next word in a sequence.
- **Contextual Generation:** Fine-tuned on various datasets to improve its ability to generate contextually appropriate responses.

Strengths:

- **Natural Text Generation:** Capable of generating highly fluent and coherent text.
- **Versatile Applications:** Effective for a wide range of tasks including translation, summarization, and question answering.

Limitations:

- **Resource Intensive**: Requires substantial computational resources for training and inference.
- **Sensitivity to Input**: Performance can be highly dependent on the quality and specificity of input prompts.
- T5 (Text-To-Text Transfer Transformer).
- T5 treats all NLP tasks as a text-to-text problem, where both inputs and outputs are text strings.

1.4.4 Other Decoder Models

GPT-3

GPT-3 is an extension of the original GPT models, significantly increasing the number of parameters to 175 billion. It offers advanced language generation capabilities and can perform a variety of tasks with few-shot or zero-shot learning.

Strengths:

High Performance: Delivers superior performance on many NLP tasks with minimal task-specific training data. Few-Shot Learning: Can generalize to new tasks with very few examples.

Limitations:

Extremely Resource Intensive: Requires massive computational resources for training and deployment. Accessibility: Due to its size, access is often limited to API-based usage.

1.5 Asymmetric Models (MS MARCO)

Asymmetric models in information retrieval employ distinct encoders to learn separate representations for queries and documents, enabling a more nuanced understanding of user intent and document relevance. The MS MARCO dataset, comprising millions of real user queries and passages, has been instrumental in driving advancements in this area. These models typically use cross-encoder architectures, where the query and document representations are jointly processed to produce a relevance score.

While asymmetric models can effectively handle complex queries and capture semantic relationships, their increased computational complexity compared to symmetric models like BM25 is a potential limitation. In the context of our fact-checking system, exploring asymmetric models trained on MS MARCO could offer an alternative or complementary approach to BM25 for candidate retrieval, potentially improving the identification of relevant fact-checking articles, especially for complex claims.

1.6 Longformers

Longformers address the limitations of traditional Transformer models in handling long sequences by introducing sparse attention mechanisms. This allows them to efficiently process longer documents while preserving the ability to capture contextual information crucial for understanding the broader context. Longformers achieve this by strategically combining sparse attention with global attention mechanisms.[1]

The ability of Longformers to efficiently handle long documents and capture long-range dependencies could be particularly beneficial for our fact-checking system, especially when

dealing with lengthy fact-checking articles. However, careful hyperparameter tuning and consideration of the trade-off between efficiency and capturing long-range context are necessary when incorporating Longformers.[1]

1.7 FAISS for Efficient Similarity Search

1.7.1 Introduction

In the realm of large-scale information retrieval and machine learning applications, efficient similarity search is crucial for tasks such as finding nearest neighbors, clustering, and recommendation systems. FAISS (Facebook AI Similarity Search) is a library specifically designed to address this challenge. It provides highly optimized implementations of various indexing structures and search algorithms, enabling efficient similarity search and clustering of dense vectors, even in massive datasets.[8]

1.7.2 Key Concepts and Techniques

- **Indexing Structures:** FAISS employs a variety of indexing structures, such as Inverted File (IVF) and Product Quantization (PQ), to efficiently store and organize high-dimensional vectors. These structures enable faster search times by partitioning the vector space and reducing the number of distance computations required.
-
- **Approximate Nearest Neighbor Search:** FAISS implements several approximate nearest neighbor search algorithms, which trade off some accuracy for significant speedups in large-scale datasets. These algorithms, such as IVF-PQ, allow for fast retrieval of the most similar vectors to a given query vector, even when dealing with millions or billions of vectors.

1.7.3 Advantages and Limitations

Advantages:

- **Efficiency:** FAISS is highly optimized for performance, making it suitable for handling large-scale datasets with millions or even billions of vectors.
- **Flexibility:** It supports various distance metrics (e.g., cosine similarity, Euclidean distance) and offers different indexing and search algorithms to cater to different use cases.
- **Hardware Acceleration:** FAISS can leverage both CPU and GPU resources, further enhancing its efficiency and scalability.

Limitations:

- Approximate Search: While FAISS excels at approximate nearest neighbor search, it may not always guarantee the retrieval of the absolute nearest neighbors, especially when using highly compressed indexing structures.
- Index Building Time: Building the initial index can be time-consuming, especially for very large datasets.

Conclusion

In this chapter, we explored the fundamental technologies that underpin modern automated fact-checking systems, laying the groundwork for the development of accurate, efficient, and scalable solutions. We began by introducing the essential concepts of Information Retrieval (IR), focusing on foundational models like Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and advanced techniques such as BM25. These methods provide the tools to locate and rank relevant documents based on their content and user queries, addressing critical challenges in document representation and relevance estimation.

We then delved into the transformative impact of deep learning in Natural Language Processing (NLP), examining how advancements in neural network architectures—particularly the rise of transformer-based models—have revolutionized language understanding. Technologies like word embeddings, recurrent networks, and Transformers, including BERT and Longformers, have dramatically enhanced the ability to capture semantic relationships, process long documents, and perform complex tasks like claim-fact matching.

The chapter also highlighted innovative approaches like FAISS for efficient similarity search and the role of asymmetric models in improving document relevance understanding. Together, these technologies create a robust foundation for building sophisticated fact-checking systems capable of navigating the challenges of misinformation in today's information-rich environment.

By bridging traditional IR methods with cutting-edge NLP advancements, this chapter sets the stage for deeper exploration of how these technologies integrate to form state-of-the-art automated fact-checking systems, which we will examine in subsequent chapters.

Chapter 2

Existing Approaches to Claim-Fact Matching

Introduction

The rise of misinformation across digital platforms has highlighted the critical need for efficient and scalable claim-fact matching systems. These systems aim to identify and retrieve fact-checked information corresponding to new claims, thereby curbing the spread of false information. This chapter examines state-of-the-art approaches to claim-fact matching, focusing on methodologies that address key challenges such as paraphrase detection, multimodal content retrieval, and multilingual misinformation.

We explore three notable approaches: a multimodal retrieval system that integrates text and image data, a system designed to improve retrieval accuracy by leveraging projection layers for cross-modal matching, and a multilingual framework that facilitates cross-lingual retrieval of fact-checked claims. Each method is analyzed in terms of its methodology, key contributions, strengths, and limitations, providing a comprehensive overview of the current advancements in the field. This analysis serves as a foundation for identifying gaps and opportunities for future innovations in claim-fact matching.

2.1 Key Papers and their Contributions

2.1.1 Paper 1: That is a Known Lie: Detecting Previously Fact-Checked Claims

Introduction:

The paper focuses on the challenge of **automatically detecting claims that have already been fact-checked** to combat misinformation more efficiently. The primary problem the authors aim to solve is finding fact-checked claims that match a new claim, even when the claim is rephrased or paraphrased. This ability is critical for fact-checking organizations, which often encounter previously fact-checked claims in new guises.[18]

To address this, the authors propose a two-step retrieval and ranking system that leverages both textual and visual information, enabling the system to handle a broader variety of claims, including those accompanied by images.[18]

Methodology:

The paper's approach is divided into two main steps:

1. BM25 Retrieval:

    - The BM25 algorithm is a standard and well-established information retrieval model used in this paper to perform an initial retrieval of fact-checked claims. BM25 scores documents based on their relevance to a query, where the query in this case is a new claim. It operates on simple textual features like term frequency and document length.

- BM25 serves as the first retrieval layer because of its effectiveness in quickly retrieving documents that may be relevant to the claim. It narrows down the number of candidate fact-checks from a large pool, focusing on the most likely matches based on keywords.
- However, BM25 alone cannot account for paraphrased claims or the use of images, which is where the second step of the approach comes in.

2. Re-ranking with a Multimodal Model:

- The second layer involves re-ranking the candidates retrieved by BM25 using a neural network model that can process both textual and visual information. The model takes in both the text of the claim and associated images (if available) and uses them to re-rank the fact-checks.
- Textual Matching: The neural model processes the claim and the fact-check text using word embeddings, which represent words as dense vectors. These embeddings allow the model to detect semantic similarity between the claim and the fact-check even when they are paraphrased or written differently.
- Visual Matching: The system can also handle image-based claims, which is crucial when verifying claims on social media where images play a significant role in spreading misinformation. For example, if a new claim comes with an image, the model will look for fact-checks that involve similar visual elements.
- The output of this model is a ranking of the fact-checks, prioritizing those that are most likely to match the new claim, based on both text and image similarities.

2.1.2 Key Contributions:

- **Multimodal Approach:** One of the primary innovations of this paper is its multimodal retrieval system, which integrates both text and images to retrieve fact-checked claims. This is especially relevant given the increasing use of images and memes in misinformation.
- **Paraphrase Detection:** By using a neural re-ranking model with word embeddings, the system can detect paraphrased claims, which traditional text-based methods like BM25 struggle with.
- **Scalable Framework:** The use of BM25 in the first stage ensures that the system is scalable, as it allows the neural re-ranking model to focus only on a small set of relevant candidates rather than the entire fact-check database.

2.1.3 Strengths:

- **Multimodal Retrieval:** By incorporating both text and images, this paper addresses an important gap in fact-checking research. Many fact-checking systems

focus solely on textual data, but misinformation often spreads via images, especially on social media. This system's ability to process visual data improves its versatility.

- **Efficient Two-Step Approach:** The two-step retrieval and ranking approach allows the system to be both fast and accurate. BM25 quickly narrows down the candidate set, and the neural re-ranking model refines this further by incorporating deeper semantic and multimodal information. This makes it scalable for use with large fact-check databases.
- **Handling Paraphrased Claims:** The use of word embeddings to represent text enables the system to match paraphrased claims with fact-checks. This is a critical improvement over traditional retrieval methods, which rely solely on exact keyword matches.
- **Potential Real-World Application:** The authors emphasize that this framework could be used in real-world fact-checking platforms where users submit claims, and the system can quickly verify whether the claim has already been checked. This offers direct utility in reducing the workload of human fact-checkers.

Weaknesses:

- Dataset Limitations: The effectiveness of the system heavily depends on the quality and size of the dataset used for training. In cases where there is limited data (especially in terms of multimodal content), the performance of the model might be suboptimal. For example, if only a few fact-checks contain images, the model's ability to handle image-based claims may be limited.
- Computational Resources: While the BM25 stage is computationally efficient, the neural re-ranking model requires more significant computational resources, especially when dealing with large datasets or real-time claim verification. This could be a bottleneck in scaling the system to a global fact-checking infrastructure.
- Generalization to Complex Claims: The paper primarily focuses on claims that are fact-checkable via simple text or image matches. However, more complex claims that require deeper reasoning or knowledge of context (such as nuanced political claims) might not be as easily handled by the system.

2.1.4 Paper 2: Where Are the Facts? Searching for Fact-Checked Information to Alleviate the Spread of Fake News

Introduction:

This paper addresses the growing problem of fake news by proposing a multimodal retrieval system designed to efficiently retrieve fact-checked information. The system matches claims with fact-checked content, utilizing both textual and visual data (images), making it suitable for combating misinformation, especially in multimedia-rich

environments like social media. The main goal is to enhance the retrieval process by leveraging multimodal information and deep learning techniques for accurate claim-fact matching[20].

Methodology:

The system is based on a two-stage retrieval and ranking process, similar to the first paper but with significant enhancements, particularly in the handling of multimodal (text + image) data. Here's a breakdown of the key components[20]:

1. **BM25 for Initial Textual Retrieval:**

   - Like many claim-fact matching systems, this paper uses BM25 as the first step in the retrieval pipeline. BM25 scores documents based on their relevance to a query (the claim) by considering keyword frequency and document length.
   - BM25 operates purely on textual information and serves as a fast filtering layer, reducing the number of potential fact-checks that need deeper processing.

2. **Multimodal Matching System:**

   - The real strength of the system lies in the second stage, where it uses multimodal data (text + image) to more accurately match claims to fact-checks. This second stage consists of the following key components:
   - **Projection Layer:** This layer projects both textual and visual features into a shared embedding space. The idea is to align textual and visual information such that both types of data can be compared and matched effectively. Claims and fact-checks, even if they involve different modalities (e.g., text in one and images in another), are projected into this common space to facilitate matching.
     - **Textual Embeddings:** The text from the claim and fact-checks are transformed into dense vector representations using word embeddings (such as Word2Vec or BERT). These embeddings allow the system to capture the semantic meaning of words and sentences, making it possible to detect paraphrased claims.
     - **Visual Embeddings:** Similarly, images associated with claims or fact-checks are processed using convolutional neural networks (CNNs), which extract visual features from the images. These features are also projected into the shared embedding space, allowing the system to compare images with both text and other images.
   - **Multimodal Matching Layer:** This layer is responsible for matching the projected features (text and image) from the claim and fact-checks. The system compares the textual and visual features of the claim against the fact-check and produces a similarity score. Fact-checks with higher similarity scores are considered more relevant to the claim.

– If a claim contains both text and images, the system can match it to fact-checks containing either text, images, or both, making it highly versatile for real-world misinformation detection.

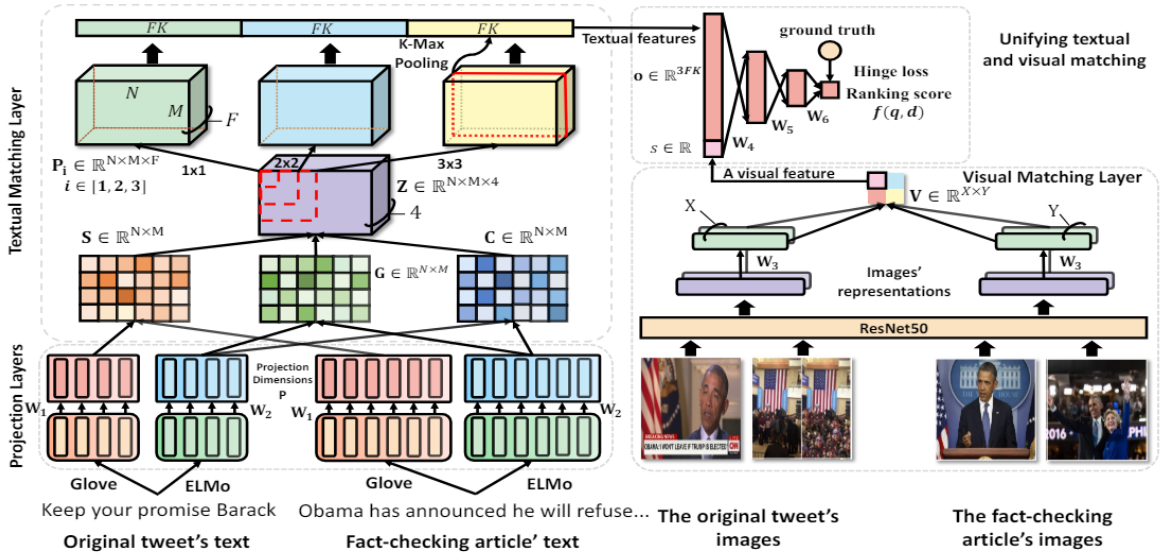

Figure 2.1: MAN system architecture [20]

Key Contributions:

1. Multimodal Retrieval:

   - A major innovation of this paper is its ability to process and match both text and images, recognizing that fake news often involves not just misleading text but also images (e.g., memes or manipulated photos). The use of multimodal embeddings makes it possible to match claims and fact-checks across different types of content.

2. Projection Layer for Cross-Modal Matching:

   - The use of a projection layer that aligns both textual and visual data into a common space is a critical contribution. It ensures that the system can compare and match claims and fact-checks even when they involve different modalities, such as matching a text-only claim with a fact-check that contains both text and images.

3. Improved Retrieval Accuracy:

   - By incorporating visual features alongside text, the system significantly improves retrieval accuracy, particularly for claims that involve images. This is important for fact-checking fake news in environments like social media, where images are often used to mislead.

Strengths:

1. Multimodal Fusion:
    - The system's ability to fuse both textual and visual information sets it apart from traditional text-only retrieval systems. This is especially useful for detecting fake news that includes misleading or manipulated images, which is increasingly common on platforms like Twitter, Facebook, and Instagram.
2. Flexible Matching Layer:
    - The multimodal matching layer allows the system to match claims and fact-checks even when they differ in modality. For instance, it can match a text-based claim with a fact-check that contains both text and an image, providing flexibility that is crucial in real-world fact-checking scenarios.
3. High Performance:
    - The paper demonstrates that the system outperforms text-only models in retrieval tasks, especially for fake news involving images. The multimodal approach leads to more accurate matches and a better overall performance in identifying the correct fact-check.

Weaknesses:

1. **Dataset Limitations:**
    - The system's performance is highly dependent on the availability of high-quality, multimodal datasets. While the authors build a dataset that includes both text and images, such datasets are still relatively rare in the fact-checking domain. The system may struggle when there is an imbalance in the availability of textual and visual data.
2. **Complexity and Scalability:**
    - The deep learning models used for processing images and text, as well as the projection layer, introduce significant computational complexity. This could pose challenges in terms of scalability, especially for real-time fact-checking applications. Running deep neural networks at scale, particularly when handling multimodal data, requires substantial computational resources, which might be a limiting factor for large-scale deployment.
3. **Image Relevance:**
    - The system assumes that images associated with claims or fact-checks are always relevant to the claim. However, in many cases, images can be used as attention-grabbing elements without directly contributing to the meaning of the claim. The system may not always account for such irrelevant images, leading to potential mismatches.

2.1.5 Paper 3: Multilingual Previously Fact-Checked Claim Retrieval

This paper tackles one of the major challenges in claim-fact matching: the multilingual nature of misinformation. Misinformation spreads globally across languages, making it crucial for fact-checking systems to retrieve previously fact-checked claims in a multilingual context. This paper introduces a system designed to retrieve fact-checked claims across multiple languages, addressing the growing need for cross-lingual retrieval systems in combating misinformation.[14]

Methodology:

The core contribution of this paper is a multilingual claim matching system that uses cross-lingual embeddings and machine translation techniques to match claims in one language with fact-checks in another.[14]

1. **Multilingual Dataset:**
    - The authors construct a large multilingual dataset of fact-checked claims to train and evaluate the system. This dataset includes claims and fact-checks in multiple languages (such as English, Spanish, French, etc.). By building a dataset that covers multiple languages, they ensure that the system can generalize across different linguistic contexts.
2. **Cross-Lingual Embeddings:**
    - The system relies on cross-lingual embeddings, which project text from different languages into a shared embedding space. This allows the system to compare claims in one language with fact-checks in another language.
        - For example, a claim made in Spanish can be matched with a fact-check written in English because both the claim and fact-check are mapped to a common embedding space.
        - The embeddings are typically pre-trained on large multilingual corpora using models like mBERT (Multilingual BERT) or XLM-R (Cross-lingual Language Model-Robust), which are capable of representing text from multiple languages in a unified space.
3. **Machine Translation:**
    - The system incorporates machine translation to facilitate direct comparisons between claims and fact-checks in different languages. If a direct match cannot be found through the cross-lingual embeddings, the system translates the claim or fact-check into a common language (typically English) and then performs a comparison.
        - This two-step process (cross-lingual embeddings + machine translation) enhances the system's ability to match claims and fact-checks across diverse languages, even when they have different structures or vocabularies.

4. **Retrieval Process:**

   - Similar to the previous papers, the system employs a two-stage retrieval process[14]:
     (1) BM25-based Retrieval: As with most retrieval systems, the first step is a BM25-based search over a large collection of fact-checks. BM25 retrieves fact-checks that are lexically similar to the claim, serving as an initial filtering layer.
     (2) Re-ranking with Cross-Lingual Embeddings: After the BM25 step, the retrieved fact-checks are re-ranked based on their similarity to the claim in the cross-lingual embedding space. Fact-checks that are semantically similar, even across different languages, are ranked higher.

5. **Multilingual Re-ranking:**

   - The final ranking of fact-checks takes into account the semantic similarity between claims and fact-checks in different languages. The system ensures that claims that have been paraphrased or rephrased in another language are still matched to the correct fact-check.[14]

Key Contributions:

1. **Multilingual Fact-Check Retrieval:**

   - The main contribution of the paper is a system capable of retrieving fact-checked claims in multiple languages. This is a crucial advancement because most fact-checking systems are monolingual, limiting their effectiveness in global contexts where misinformation is often shared in multiple languages.

2. **Cross-Lingual Embeddings:**

   - By using cross-lingual embeddings, the system is able to map claims and fact-checks from different languages into a shared space, making it possible to perform cross-lingual retrieval. This significantly improves the flexibility and utility of the system in multilingual environments.

3. **Large Multilingual Dataset:**

   - The authors create and use a large multilingual dataset of fact-checked claims, which is a valuable resource for the community. The dataset not only serves to train and evaluate the system but can also be used by other researchers working on cross-lingual misinformation detection.

Strengths:

1. **Multilingual Capabilities:**
    - The system's ability to retrieve fact-checked claims across multiple languages makes it a powerful tool for global fact-checking. As misinformation spreads internationally, having a system that can handle multiple languages is essential.
2. **Cross-Lingual Embedding Space:**
    - The use of cross-lingual embeddings allows the system to overcome the language barrier, enabling it to compare claims and fact-checks from different languages effectively. This is a significant improvement over monolingual systems that would require fact-checks in each specific language.
3. **Robust Retrieval Process:**
    - The system combines traditional BM25 retrieval with advanced re-ranking using cross-lingual embeddings, ensuring that fact-checks are retrieved efficiently and ranked accurately, even in a multilingual setting.

Weaknesses:

1. **Reliance on Pre-trained Embeddings:**
    - The system heavily relies on pre-trained cross-lingual embeddings like mBERT or XLM-R. While these models perform well, they may not capture the full nuance of certain languages or dialects. For instance, low-resource languages or regional dialects may not be well-represented in the embedding space, leading to suboptimal performance in those cases.
2. **Quality of Machine Translation:**
    - The quality of machine translation can vary significantly across languages. While the system uses machine translation to handle cases where the cross-lingual embeddings don't perform well, the quality of the translation can impact the accuracy of the claim-fact matching. Poor translations could lead to mismatches or false positives.
3. **Dataset Coverage:**
    - Although the authors create a large multilingual dataset, it may not cover all languages equally. Low-resource languages might not have enough fact-checks available, which could limit the system's effectiveness in those regions. In cases where the system encounters languages with limited fact-checking data, it may struggle to retrieve accurate results.

Conclusion

This chapter presented an in-depth analysis of existing approaches to claim-fact matching, highlighting their methodologies, innovations, and areas for improvement. The multimodal retrieval systems demonstrated the ability to process text and image data effectively, addressing a significant gap in traditional text-only fact-checking methods. The incorporation of projection layers and shared embedding spaces further enhanced the accuracy of cross-modal retrieval systems.

The review of multilingual claim-fact matching emphasized the importance of cross-lingual embeddings and machine translation in tackling misinformation on a global scale. While these advancements have significantly improved the scalability and adaptability of fact-checking systems, challenges remain, particularly in addressing dataset limitations, computational resource demands, and the generalization of systems to handle complex claims.

The insights gained from these approaches underscore the progress made in the field and the challenges that still need to be overcome. These findings provide a valuable foundation for the subsequent exploration of novel methodologies aimed at enhancing the efficiency, scalability, and accuracy of claim-fact matching systems in combating misinformation.

Chapter 3

Methodology

Introduction

This chapter outlines the methodology employed to develop and evaluate the proposed claim-fact matching system. The methodology combines innovative retrieval techniques with robust evaluation metrics to ensure accuracy, scalability, and relevance in the matching process. A clear understanding of these methods is essential for assessing the system's ability to address challenges like paraphrased claims, multimodal data, and multilingual content.

The chapter introduces key components, including the retrieval pipeline, modeling techniques, and evaluation metrics. Specific attention is given to metrics like Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG), which are integral to measuring the system's performance in ranking relevant fact-check articles. Practical examples are provided to clarify the application of these metrics and to highlight their importance in assessing ranking accuracy and relevance. By detailing both the technical foundation and evaluation framework, this chapter establishes the basis for analyzing the efficacy of the proposed system in the subsequent chapters.

3.1 Research Objective

The primary objective of this research is to develop an automated system for claim-fact matching that effectively addresses the challenges posed by the proliferation of misinformation online. We aim to create a system that can accurately and efficiently identify relevant fact-checking articles for a given claim, enabling users to quickly verify the veracity of information they encounter.

3.2 Overall Approach

To achieve this objective, we propose a two-stage approach that leverages the complementary strengths of the BM25 algorithm and the SBERT model. In the first stage, BM25 is employed to efficiently retrieve a set of candidate fact-checking articles that are potentially relevant to the given claim. This initial retrieval step helps to narrow down the search space, enabling the subsequent stage to focus on a smaller and more manageable set of articles.

In the second stage, we utilize SBERT to re-rank the candidate articles based on their semantic similarity to the claim. By capturing the nuanced meaning and context of both the claim and the articles, SBERT ensures that the most relevant and informative fact-checks are presented for further analysis. This two-stage approach combines the efficiency of BM25's keyword-based retrieval with the semantic understanding of SBERT, resulting in a system that is both effective and accurate in identifying relevant fact-checking information.

3.3 Text Preprocessing

Text preprocessing is a crucial step in natural language processing (NLP) that prepares raw text data for analysis. It involves transforming the text into a cleaner and more structured format, which enhances the effectiveness of subsequent algorithms and models in understanding and interpreting the text. In our fact-checking system, we initially explored two preprocessing setups: a base setup and an extra setup with additional normalization techniques.

3.3.1 Base Setup

The base setup focused on essential cleaning and normalization techniques:

- **Special Character Escaping:** We escaped control and special characters, like zero-width characters, to prevent them from disrupting the tokenization process.
- **Link and Emoji Removal:** We removed URLs and emojis from the text using regular expressions and the ‘demoji.replace()‘ function, respectively, as they often do not contribute to the core meaning of the claim or article.
- **Punctuation and Number Removal:** We removed punctuation marks and individual numerical digits, as they rarely provide meaningful semantic information in this context. We used a translation table generated from the ‘string.punctuation‘ module for punctuation removal and regular expressions for number removal
- **Hashtag Stripping:** We removed the ”” symbol from hashtags while preserving the rest of the word, allowing us to capture the semantic content of hashtags without the noise of the symbol itself.
- **Stemming:** We applied stemming to reduce words to their base or root forms using the SnowballStemmer algorithm. This helps to standardize variations of the same word and improve the matching of semantically similar terms. We supported both English and French stemming using ‘STEMMER[”english”]‘ and ‘STEMMER[”french”]‘, respectively.
- **Stop Word Removal:** We eliminated common stop words (e.g., ”the,” ”a,” ”and”) that occur frequently but contribute little to the overall meaning of the text. We utilized NLTK’s stopword lists for both English and French (‘STOPWORDS[”english”]‘ and ‘STOPWORDS[”french”]‘).
-

3.3.2 Additional Preprocessing Techniques Explored

In addition to the base setup, we also experimented with an extra setup that incorporated further normalization and enrichment techniques:

- **Number Normalization:** We converted numbers in numerical format to their word equivalents (e.g., 131 to "one hundred thirty-one") to potentially capture their semantic meaning more effectively.
- **Date Normalization:** We transformed complete dates into a standardized format ("YYYY-MM-DD") to ensure consistency in date representation, which could be useful for time-sensitive claims.
- **Entity Recognition:** We employed named entity recognition (NER) to identify and tag entities such as persons, locations, organizations, and events. This aimed to provide additional context and improve semantic understanding.
- **Metadata Inclusion for BM25:** For the BM25 retrieval stage, we experimented with including the date of the claim and its source as additional tokens, hypothesizing that this metadata could enhance retrieval relevance.
- **Claim Formatting for Similarity Models:** For the SBERT re-ranking stage, we formatted claims to include metadata (source and date) in a structured manner, potentially aiding the model in capturing contextual information.

Rationale for Focusing on Base Setup

While we explored these additional preprocessing techniques, our empirical evaluation indicated that they did not lead to significant improvements in overall system performance. The base setup, being simpler and more computationally efficient, provided a good balance between effectiveness and complexity. Therefore, we opted to focus on the base setup for our final results.

3.3.3 Tokenization

* We segmented the text into individual words or subwords using an MPNetTokenizerFast tokenizer. This tokenizer is a subword tokenizer that can handle out-of-vocabulary words effectively, ensuring that all words in the text are represented in a meaningful way.

3.3.4 Alphabetic Filtering

* We kept only tokens consisting entirely of alphabetic characters to filter out any remaining non-word tokens that might have slipped through the previous preprocessing steps.

3.3.5 Rationale

The rationale behind this base preprocessing pipeline is to clean and normalize the text while preserving essential information for effective claim-fact matching. By removing noise (links, punctuation, emojis, numbers), reducing words to their base forms, and filtering out stop words, we aimed to create a more meaningful representation of the text that

would facilitate accurate semantic similarity calculations in the subsequent stages of our system.

3.3.6 Libraries and Tools:

We utilized the following libraries and tools:

- NLTK: For stop word lists, stemming, and tokenization.
- Transformers: For the MPNetTokenizerFast tokenizer.
- demoji: For emoji removal.
- re: For regular expression-based link and number removal.
- string: For generating the punctuation translation table.

3.4 Stage 1: Candidate Retrieval with BM25

The initial stage of our fact-checking system focuses on efficiently retrieving a set of candidate fact-checking articles that are potentially relevant to a given claim. To achieve this, we utilize the BM25 (Best Matching 25) algorithm, a probabilistic information retrieval model that has proven effective in ranking documents based on their relevance to a query.

3.4.1 BM25 Algorithm:

BM25 is a bag-of-words model that calculates a relevance score for each document based on the query terms. The score takes into account the following factors:

1. **Term Frequency (TF):** The frequency of query terms within the document. More frequent terms are generally considered more relevant.
2. **Inverse Document Frequency (IDF):** The rarity of query terms across the entire document collection. Rarer terms are typically more informative and receive higher weights.
3. **Document Length Normalization:** The length of the document relative to the average document length in the collection. Longer documents are penalized as they may contain more irrelevant information.

3.4.2 Implementation:

We used the rank_bm25 library in Python to implement the BM25 algorithm. Specifically, we created a BM25Okapi object (BM25) and indexed our corpus of preprocessed English fact-checking articles (en_articles_tokens). The bm25 function performs the following steps:

1. Query Preprocessing: If the query is not already tokenized, it preprocesses it using the preprocess_text function described in the previous section.
2. Candidate Retrieval: It uses the BM25.get_top_n method to retrieve the top n documents (articles) based on their BM25 scores calculated against the query. By default, we set n to 10 to retrieve the top 10 most relevant articles.

3.4.3 Rationale:

We chose BM25 for candidate retrieval due to its several advantages:

- **Efficiency:** BM25 is computationally efficient, making it suitable for large document collections. It uses an inverted index data structure to quickly identify documents containing the query terms.
- **Effectiveness:** BM25 has consistently demonstrated strong performance in various information retrieval tasks, including ad hoc retrieval and question answering.
- **Simplicity:** BM25 is relatively easy to implement and understand, making it a practical choice for our fact-checking system.

By employing BM25 in the first stage, we efficiently narrow down the search space, enabling the subsequent SBERT-based re-ranking stage to focus on a smaller set of potentially relevant articles and refine the results based on semantic similarity.

3.5 Stage 2: Re-ranking with SBERT

After retrieving a set of candidate fact-checking articles using BM25, the second stage of our system focuses on re-ranking these candidates based on their semantic similarity to the claim. For this purpose, we leverage Sentence-BERT (SBERT), a state-of-the-art sentence embedding model that excels at capturing the semantic meaning of sentences.

3.5.1 SBERT Models:

We utilized two pre-trained SBERT models depending on the language of the claim:

- **English Claims:** For English claims, we used the **sentence-transformers / all-mpnet-base-v2** model. This model is trained on a massive amount of English text data and has demonstrated strong performance on various semantic textual similarity tasks. Its ability to understand nuances in the English language makes it well suited for comparing English claims with fact-checking articles.
- **French Claims:** For French claims, we used the **sentence-transformers / paramagnet-multilingual-mpnet-base-v2** model. This multilingual model is specifically designed to handle semantic similarity between different languages. Its ability to understand both English and French makes it ideal for comparing French claims with potentially multilingual fact-checking articles.

3.5.2 Model Training:

While the SBERT models we used were pre-trained on large datasets, we did not perform any additional fine-tuning for this specific task. We relied on their pre-trained knowledge to generate meaningful sentence embeddings for both claims and candidate articles.

3.5.3 Similarity Calculation:

We calculated the semantic similarity between a claim and each candidate article by comparing their corresponding sentence embeddings. Specifically, we used the cosine similarity metric, which measures the cosine of the angle between two vectors. Cosine similarity ranges from -1 (completely dissimilar) to 1 (identical), with higher values indicating greater similarity.

3.5.4 Implementation:

We used the Sentence Transformers library in Python to load the pre-trained SBERT models and compute sentence embeddings. We then calculated the cosine similarity between the claim embedding and each candidate article embedding, and sorted the articles in descending order of similarity. The articles with the highest similarity scores were considered the most relevant to the claim.

3.5.5 Rationale:

We chose SBERT for re-ranking due to its several advantages:

- **Semantic Understanding:** SBERT models excel at capturing the semantic meaning of sentences, allowing for more accurate similarity comparisons than traditional keyword-based methods.
- **Transfer Learning:** Pre-trained SBERT models leverage knowledge learned from large-scale text corpora, enabling effective generalization to new domains and tasks.
- **Efficiency:** SBERT models can efficiently generate sentence embeddings, making them suitable for real-time fact-checking applications.

By combining the initial candidate retrieval with BM25 and the subsequent re-ranking with SBERT, our fact-checking system achieves both efficiency and accuracy in identifying relevant fact-checking articles for a given claim. This two-stage approach leverages the strengths of both techniques, resulting in a robust and effective fact-checking system.

3.6 Ranking Metrics:

In addition to top-k accuracy, we employed three ranking metrics to evaluate the overall performance of our system:

3.6.1 Mean Reciprocal Rank (MRR)

Definition: The average of the reciprocal ranks of the first relevant item in a set of queries[22].

Calculation Equation:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}$$

where:

- **N** is the total number of claims.
- **rank_i** is the rank (position) of the first relevant article for claim i.

Example: For 3 claims with the first relevant article found at ranks 1, 3, and 5, the MRR is:

$$\text{MRR} = \left(\frac{1}{3}\right) * \left(\frac{1}{1} + \frac{1}{3} + \frac{1}{5}\right) = 0.51$$

3.6.2 Mean Average Precision (MAP)

Definition: The average precision across all recall levels for a set of queries[22].

Calculation Equation:

$$\text{MAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

where:

- **N** is the total number of claims.
- **AP_i** is the average precision for claim i.

Average Precision (AP) for a single claim:

$$\text{AP} = \frac{1}{R} \sum_{k=1}^{n} P(k) \cdot \text{rel}(k)$$

where:

- **R** is the number of relevant articles for the claim.
- **P(k)** is the precision at rank k (number of relevant articles up to rank k divided by k). **rel(k)** is an indicator function equal to 1 if the item at rank k is relevant, and 0 otherwise.

Example: For a claim with 3 relevant articles at ranks 1, 4, and 6 (out of 10 total), the AP is:

$$\text{AP} = \left(\frac{1}{3}\right) \left[\left(\frac{1}{1} \cdot 1\right) + \left(\frac{2}{4} \cdot 1\right) + \left(\frac{3}{6} \cdot 1\right)\right] = 0.67$$

Since we only have one query (claim) in this example, the Mean Average Precision (MAP) is the same as the Average Precision (AP) for that single query.

3.6.3 Normalized Discounted Cumulative Gain (NDCG)

Definition: Measures the overall ranking quality by considering the positions of relevant items and their relevance scores[22].

Calculation Equation:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}}$$

where:

- **DCG@k:** Discounted Cumulative Gain at rank k.
- **IDCG@k:** Ideal Discounted Cumulative Gain at rank k (DCG of a perfect ranking).

Discounted Cumulative Gain (DCG) at rank k:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$$

where:

- **rel_i:** Relevance score of the item at rank i

Example: Consider a claim with relevant articles at ranks 1, 4, and 6, with relevance scores of 3, 2, and 1, respectively. Let's calculate NDCG@5 (considering the top 5 results):

$$\begin{aligned}
\text{DCG@5} &= \frac{3}{\log_2(2)} + \frac{2}{\log_2(5)} + \frac{1}{\log_2(7)} + 0 + 0 \\
\text{IDCG@5} &= \frac{3}{\log_2(2)} + \frac{2}{\log_2(3)} + \frac{1}{\log_2(4)} + 0 + 0 \\
\text{NDCG@5} &= \frac{\text{DCG@5}}{\text{IDCG@5}}
\end{aligned}$$

3.7 Conclusion

This chapter presented the methodology for developing and evaluating the proposed claim-fact matching system, focusing on retrieval strategies and performance metrics. Key evaluation metrics such as Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG) were explained with practical examples to illustrate their significance in ranking and relevance assessment.

By employing these metrics, the methodology ensures a comprehensive evaluation of the system's ability to retrieve and rank relevant fact-check articles accurately. The outlined framework not only validates the system's performance but also serves as a benchmark for comparing it with existing approaches. This methodological foundation sets the stage for analyzing experimental results in the subsequent chapters, highlighting the system's effectiveness in addressing the challenges of misinformation detection.

Chapter 4

Experiments

Introduction

This chapter presents a comprehensive analysis of the experimental results obtained from various claim-fact and tweet-article matching techniques. The experiments are designed to evaluate the performance of retrieval and re-ranking methods, including BM25, SBERT, and sentence-level approaches, across different languages (English and French) and modalities (text, image captions, and OCR).

The chapter highlights the strengths and limitations of these methods, with a particular focus on sentence-level SBERT re-ranking and its ability to overcome challenges such as long article lengths and the informal nature of tweets. The results are analyzed through key evaluation metrics—Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG)—to provide a detailed understanding of the system's ranking accuracy and relevance across different experimental scenarios.

By examining the interplay between retrieval depths, language-specific models, and multimodal approaches, this chapter sheds light on the effectiveness and practicality of each method in real-world claim-fact matching and misinformation detection contexts.

4.1 Datasets

This section details the datasets we used and created for our fact-checking system.

4.1.1 Datasets Used

'all_claims_fr_en_until_22-02-24.json' Dataset

- **Source:** Google Fact Check Explorer, collected by a previous intern with code modifications to address extraction issues.
- **Specific Fixes:**

  - IFCN Website Collection: We updated the code to handle the dynamic nature of the IFCN website (https://ifcncodeofprinciples.poynter.org/signatories) to ensure consistent and accurate collection of newspaper websites.
  - Data Saving: We addressed issues related to data saving, ensuring that collected claims and articles were stored correctly and efficiently.
  - Cron Job for Daily Collection: We implemented a cron job to automate the data collection process, allowing for regular and up-to-date gathering of claims and articles from the Google Fact Check Explorer.

- **Size:** 102,142 entries
- **Format:** JSON objects with the following attributes:
  - **text:** The text of the claim.
  - **claimant:** The person or entity making the claim.
  - **claimDate:** The date the claim was made.
  - **claimReview:** A list of JSON objects, each representing a fact-check review of the claim, with the following attributes:
    * **publisher:**
    * **name:** The name of the fact-checking organization (e.g., "Snopes", "PolitiFact").
    * **site:** The website of the fact-checking organization.
    * **url:** The URL of the fact-check article.
    * **title:** The title of the fact-check article
    * **reviewDate:** The date the fact-check was published or reviewed
    * **textualRating:** The textual rating assigned to the claim by the fact-checker (e.g., "True", "False", "Mostly True").
    * **languageCode:** The language of the fact-check article
- **Content:** Fact-checked claims and their corresponding reviews from various publishers.
- **Statistics:**
  - 102,142 entries in total
  - **text** column: 102,142 non-null values
  - **claimant** column: 83,502 non-null values
  - **claimDate** column: 82,896 non-null values
  - **claimReview** column: 102,142 non-null values
- **Limitations:** This dataset only contains claims and their associated fact-check reviews, but not the full text of the fact-checking articles.
- **Usage:** We utilized the URLs provided in the 'claimReview.url' field to scrape the corresponding fact-checking articles from Snopes, PolitiFact, and Full Fact. This allowed us to create claim-article pair datasets for these specific publishers, which were then used for training and evaluating our system.

articles_afp_until_22-02-24.json:

- **Source:**
- **Size:** The dataset comprises 12,212 entries.
- **Format:** Each entry in the dataset is a JSON object with three attributes:
  - **title** (object): The title of the fact-checking article.
  - **url** (object): The URL of the fact-checking article.
  - **body** (object): The full text of the fact-checking article.
- **Languages:** The dataset includes articles in both English and French.
- **Characteristics:** The articles in the dataset cover a wide range of topics and claim types, reflecting the diversity of fact-checking work conducted by AFP.
- **Usage:** We considered the body attribute as the relevant text for our claim-fact matching task. The title and url attributes were used for reference and to ensure traceability of the articles.

4.1.2 Data collection

Snopes Claim-Article Pairs

- **Data Source:** We began with the all_claims_fr_en_until_22-02-24.json dataset, filtering it to retain only the claims fact-checked by Snopes (identified by the claimReview.publisher.name field).
- **Target URLs:** From each Snopes claim in the filtered dataset, we extracted the URL of the corresponding fact-check article from the claimReview.url field. These URLs served as targets for web scraping.
- **Web Scraping Process:**
  - We used the Python libraries requests and BeautifulSoup4 to scrape the Snopes fact-check articles.
  - The function extract_snopes_claim_and_info (provided code) was used to extract the claim text, review date, journalist name, and article text from each Snopes article page. We also extracted publisher information (publisher_name and publisher_site) from the claimReview object in the original dataset.
  - We cleaned the extracted data, removing unwanted HTML tags and validating the extracted claim text against the original claim.
- **Output Dataset:**
  - Size: The resulting dataset contains 16,025 claim-article pairs.
  - Columns:

    - title: Title of the Snopes fact-check article.
    - url: URL of the Snopes fact-check article.
    - article: Text of the Snopes fact-check article.
    - journalist: Name of the journalist who wrote the article.
    - review_date: Date the article was reviewed/published.
    - publisher_name: "Snopes" for all entries.
    - publisher_site: "snopes.com" for all entries.
    - claim: The extracted claim text from the Snopes article.
    - claimant: This column is empty as claimant information is not consistently available on Snopes.
    - textualRating: The textual rating of the claim (e.g., "Mostly True") from the original dataset.

- **Labeling:** All claim-article pairs in this dataset are labeled as "relevant" since they are sourced from Snopes, a trusted fact-checking organization.

PolitiFact Claim-Article Pairs

- **Data Source:** Similar to the Snopes process, we filtered the all_claims_fr_en_until_22-02-24.json dataset to identify claims fact-checked by PolitiFact (based on claimReview.publisher.name).
- **Target URLs:** We extracted the URLs of PolitiFact fact-check articles from the claimReview.url field of the filtered claims.
- **Web Scraping Process:**

  We used requests and BeautifulSoup4 to scrape PolitiFact articles. The custom functions extract_claim, extract_claimant, extract_journalists_and_review_dates, and extract_articles (provided code) were used to extract relevant information:

  - Claim text
  - Claimant name
  - Journalist(s) involved
  - Review date(s)
  - Article text (excluding embedded content)

  We cleaned the extracted data, removing unnecessary HTML tags and validating the claim text.

- **Output Dataset:**

  - **Size:** 10,754 claim-article pairs.
  - **Columns:**
    - title: Title of the PolitiFact fact-check article.

    - url: URL of the PolitiFact fact-check article.
    - article: Text of the PolitiFact fact-check article (excluding embedded content).
    - journalist: List of journalists who contributed to the article.
    - review_date: Review date (or list of dates if multiple journalists contributed).
    - publisher_name: "PolitiFact" for all entries.
    - publisher_site: "politifact.com" for all entries.
    - claim: The extracted claim text from the PolitiFact article.
    - claimant: The name of the person or entity making the claim.
    - textualRating: The textual rating of the claim (e.g., "True," "False") from the original dataset.

- **Labeling:** All claim-article pairs in this dataset are labeled as "relevant."

Full Fact Claim-Article Pairs

- **Data Source:** Following the same procedure as for Snopes and PolitiFact, we filtered the all_claims_fr_en_until_22-02-24.json dataset to identify claims fact-checked by Full Fact (based on claimReview.publisher.name).
- **Target URLs:** We extracted the URLs of Full Fact fact-check articles from the claimReview.url field of the filtered claims.
- **Web Scraping Process:** We used requests and BeautifulSoup4 to scrape Full Fact articles.

  The custom functions extract_claims, extract_review_date, extract_journalist_name, and extract_article (provided code) were used to extract relevant information:

  - Claim(s) text (since Full Fact articles may address multiple claims)
  - Review date
  - Journalist name
  - Article text

  We cleaned the extracted data, removing unnecessary HTML tags.

- **Output Dataset:**

  - Size: 5,454 claim-article pairs.
  - Columns:
  - title: Title of the Full Fact fact-check article.
  - url: URL of the Full Fact fact-check article.
  - article: Text of the Full Fact fact-check article.
  - journalist: Name of the journalist who wrote the article.

  - review_date: Date the article was reviewed/published.
  - publisher_name: "Full Fact" for all entries.
  - publisher_site: "fullfact.org" for all entries.
  - claims: A list of extracted claim texts from the Full Fact article.
  - textualRating: The textual rating of the claim(s) (e.g., "Mostly True," "False") from the original dataset.

- **Labeling:** All claim-article pairs in this dataset are labeled "relevant." We considered each individual claim within a Full Fact article as a separate relevant pair.

4.1.3 Tweets Dataset

- **Source:** Twitter API, accessed using extracted tweet URLs from fact-checking articles
- **Format:** JSON objects (returned by the Twitter API)
- **Content:** Tweets related to fact-checked claims, including both original tweets and potential misinformation tweets.
- **Collection Process:**

  1. **Link Extraction from Fact-Checking Articles:**
     (1) We visited the web pages of fact-checking articles identified in the dataset.
     (2) We extracted all links within the article content using web scraping techniques, employing requests and BeautifulSoup4.
     (3) **Enhancement:** For dynamic websites or articles with dynamically loaded content (such as archives), we utilized Selenium to ensure accurate and complete link extraction.
  2. **Perma.cc Link Identification:** We identified Perma.cc links (or similar archive links) within the extracted links, recognizing that these often lead to archived versions of the original tweets or sources related to the fact-check.
  3. **Metadata Extraction (for Misinformation Tweets):** For Perma.cc pages archiving misinformation tweets, we extracted the original tweet URL from the webpage's metadata (specifically, the 'content' attribute of the 'meta' tag with 'name="twitter:description"').
  4. **Tweet Retrieval using Twitter API:** We used the extracted tweet URLs (from both Perma.cc links and metadata) to retrieve the corresponding tweet data using the Twitter API.

- **Dataset Statistics:**

  The following table provides statistics on the collected tweets for each fact-checking website:

  Note: For snopes.com and politifact.com, we were unable to extract tweet URLs from the articles.

Website	Unique Articles	Links to Claims	Number of Claims	Number of Tweets
factcheck.afp.com	9322	60716	9172	10974 (4536)
factuel.afp.com	2923	16287	2893	2600 (1141)
fullfact.org	3723	4489	5454	485 (449)
checkyourfact.com	3910	3907	3911	761 (761)
africacheck.org	888	704	1789	68 (63)
verafiles.org	1548	210	1837	2 (2)
lemonde.fr	493	422	542	28 (28)
franceinfo.fr	194	133	205	60 (60)
snopes.com (*)	16024	-	16025	-
politifact.com (*)	9833	-	10761	-

Table 4.1: Tweets Dataset Statistics

- **Usage:** The collected tweets serve as the input claims for our claim-fact matching system. We aim to evaluate how effectively our system can identify relevant fact-checking articles for these real-world claims.

4.2 Claim-Fact Checking Matching

4.2.1 BM25 Retrieval (Pre-Re-ranking)

Overall Performance

BM25 demonstrates impressive performance in retrieving relevant articles for both claims and tweets, especially when considering the top few results. However, there are subtle differences in performance depending on language and data type (claims vs. tweets).

Claim-Article Matching:

Metric	Top 1	Top 3	Top 5
English Claims (EN)			
MRR	0.8967	0.9285	0.9309
MAP	0.8942	0.9270	0.9295
NDCG	0.8968	0.9368	**0.9411**
French Claims (FR)			
MRR	0.8536	0.8878	0.8922
MAP	0.8376	0.8873	0.8920
NDCG	0.8536	0.8981	**0.9062**

Table 4.2: BM25 Retrieval Performance for Claim-Article Matching

Tweet-Article Matching:

Metric	Top 1	Top 3	Top 5
English Tweets (EN)			
MRR	0.7796	0.8060	0.8106
MAP	0.7757	0.8046	0.8094
NDCG	0.7796	0.8128	**0.8213**
French Tweets (FR)			
MRR	0.8065	0.8309	0.8359
MAP	0.8037	0.8309	0.8359
NDCG	0.8065	0.8383	**0.8472**

Table 4.3: BM25 Retrieval Performance for Tweet-Article Matching

Analysis

English Claims (EN):

- High Accuracy: BM25 achieves a high degree of accuracy in retrieving the correct article for English claims, particularly when considering the top few results. For instance, the MRR at Top 1 is 0.8967, indicating that the correct article is often the top result. The NDCG of 0.9411 at Top 5 further demonstrates that BM25 effectively ranks relevant articles higher in the list.

- Marginal Gains Beyond Top 5: While the performance improves slightly with increasing k, the gains become marginal beyond Top 5. This suggests that retrieving more than 5 articles might not be necessary for most English claims, as the correct article is usually found within the top few results.

French Claims (FR):

- Slightly Lower Performance: BM25's performance on French claims is still strong, but slightly lower than on English claims across all metrics. This might be due to differences in the characteristics of the datasets or language-specific nuances that BM25 might be less sensitive to.

- Similar Trend: The overall trend for French claims is similar to English claims, with high accuracy in the top few results and diminishing returns as k increases.

Comparison (EN vs. FR):

- BM25 consistently performs better on English claims compared to French claims, suggesting that it might be slightly more effective at retrieving relevant articles for English claims. However, the differences are relatively small, and BM25 still demonstrates strong performance for both languages.

Tweet-Article Matching:

Overall Performance:

- BM25 maintains its strong performance for tweet-article matching, although the metric values are slightly lower than those for claim-article matching. This could be attributed to the less structured nature of tweets and the greater variability in language use.

English Tweets (EN):

- Good Performance: BM25 achieves good results for English tweets, with a Top 1 MRR of 0.7796, indicating that it often finds the correct article as the top result. However, the performance is slightly lower compared to English claims.

- Steadily Improving Recall: The recall increases as we consider more top results, reaching a NDCG of 0.8213 at Top 5, demonstrating BM25's ability to retrieve relevant articles even if they are not ranked at the very top.

French Tweets (FR):

Comparable to English Tweets: BM25 performs similarly on French tweets as on English tweets, with slight variations in the specific metric values. The overall trend of increasing recall with higher k values remains consistent.

Overall Discussion:

- BM25 demonstrates impressive performance in retrieving relevant articles for both claims and tweets, especially within the top few results. This confirms its effectiveness as an initial retrieval method for fact-checking purposes.

- The performance differences between English and French data, as well as between claims and tweets, highlight the influence of language and data characteristics on BM25's effectiveness.

- While BM25 excels at recall, there is still room for improvement in precision, especially at lower k values. This limitation, along with the need for better semantic understanding, will be addressed in the next section on SBERT re-ranking.

Analysis of Article Presence Plots:

Figure Description (Applicable to Figures 4.1, 4.2, 4.3, 4.4):

These bar plots illustrate the performance of BM25 in retrieving the correct fact-checking article for [data type: English/French claims/tweets]. The x-axis represents the number of top results (k) considered, ranging from 1 to 100. The y-axis shows the number of [data type] for which the correct article is found within the corresponding top-k results. The percentage displayed above each bar indicates the proportion of the total [data type] where the correct article is retrieved within that top-k set.

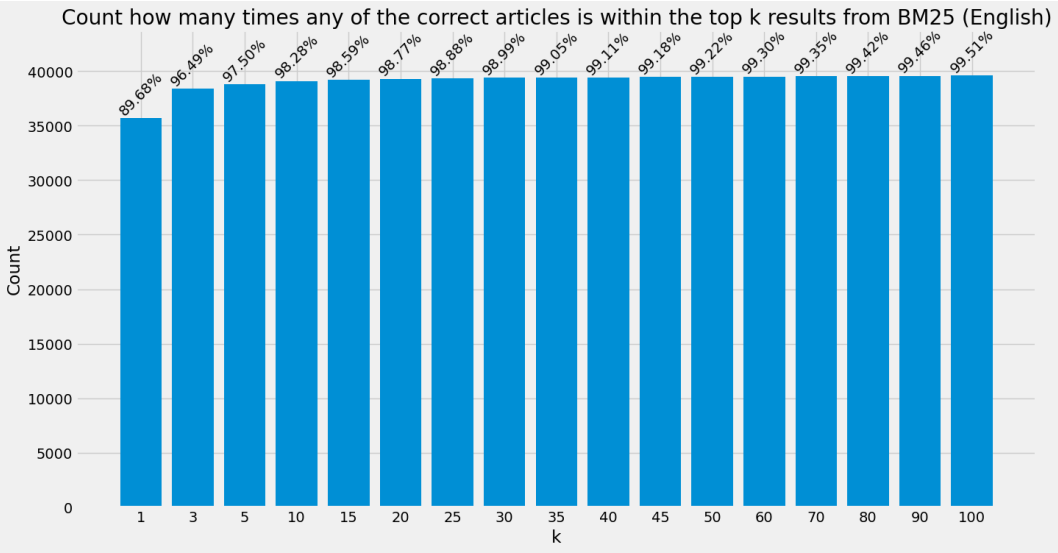

Figure 4.1: BM25 Retrieval Performance: Percentage of English Claims with Correct Article in Top-k Results

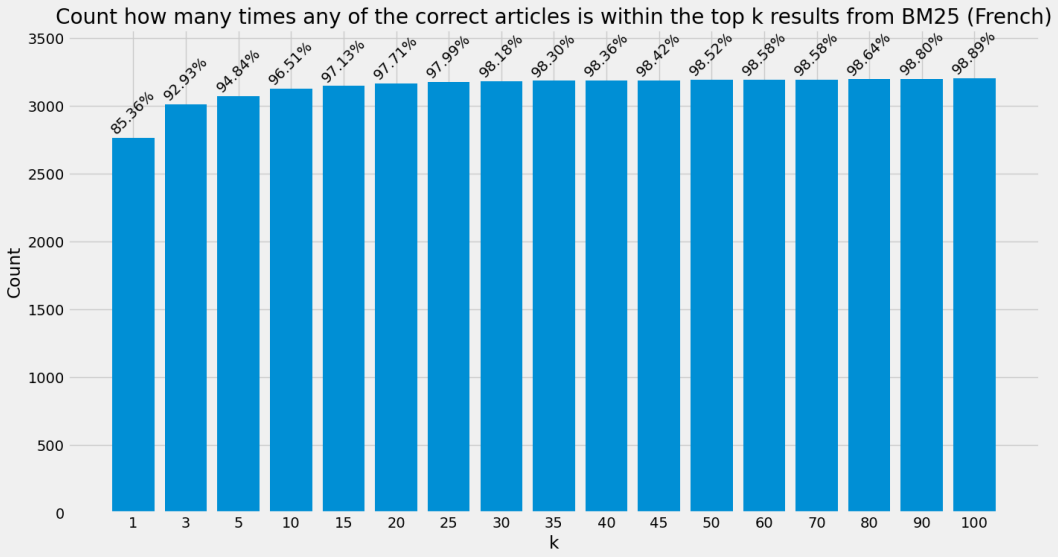

Figure 4.2: BM25 Retrieval Performance: Percentage of French Claims with Correct Article in Top-k Results

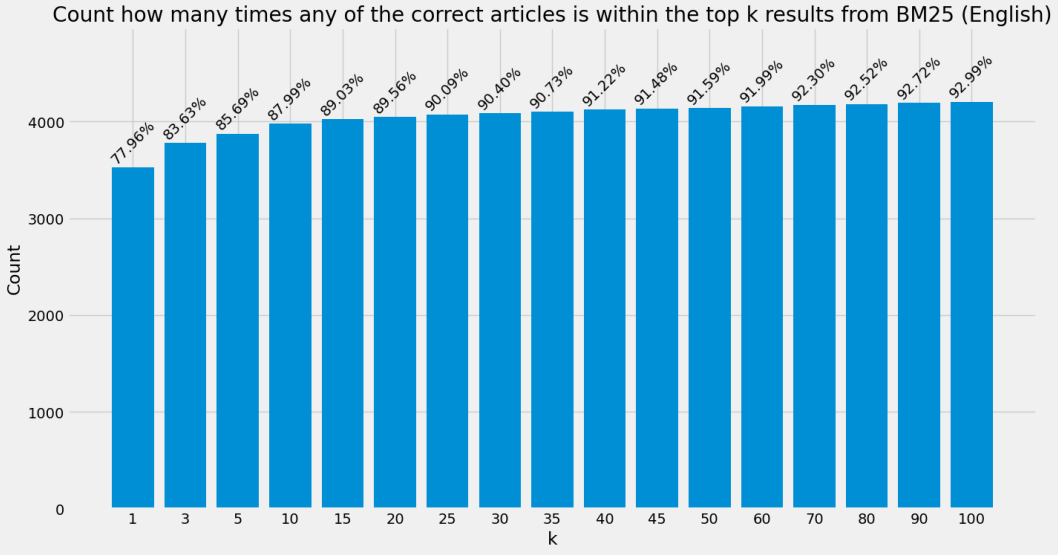

Figure 4.3: BM25 Retrieval Performance: Percentage of English Tweets with Correct Article in Top-k Results

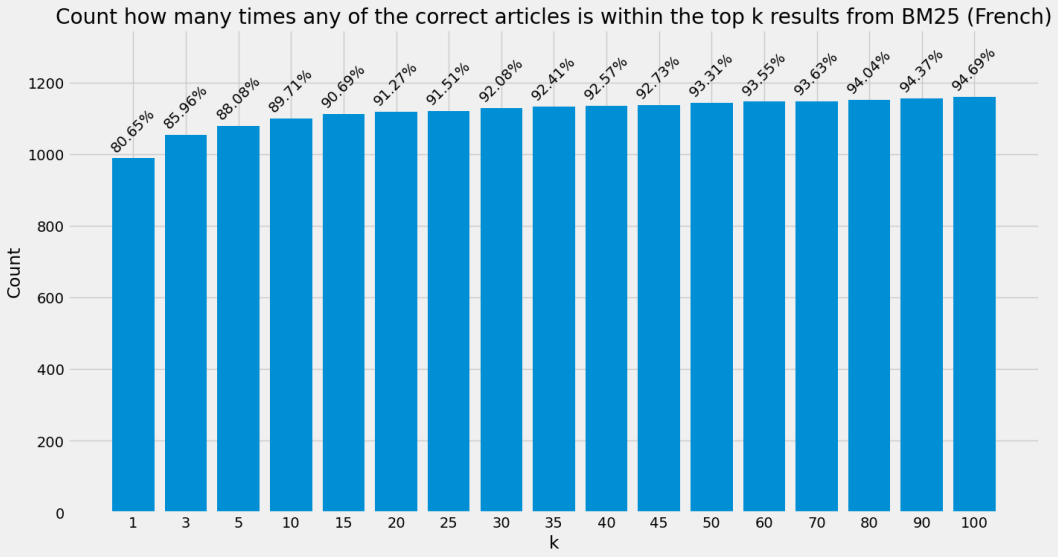

Figure 4.4: BM25 Retrieval Performance: Percentage of French Tweets with Correct Article in Top-k Results

English Claims:

High Recall Across Top-k: BM25 exhibits excellent recall for English claims, with the vast majority of correct articles found within the top 10 results. Even at Top 1, the correct article is retrieved for nearly 90% of claims, highlighting BM25's ability to quickly identify relevant articles. Diminishing Returns: The rate of improvement in recall slows down considerably after Top 10, suggesting that retrieving more articles might not significantly increase the chances of finding the correct article. Precision Could Be Improved: The plot indicates that while BM25 is good at finding the correct article, it might not always rank it as the very top result. This suggests potential room for improvement in precision.

French Claims:

Exceptional Recall: BM25's performance on French claims is even more impressive than on English claims. The correct article is found in the top results for nearly all claims, with almost 90% found at Top 1 and nearly 100% at Top 5. Rapid Convergence: The recall curve for French claims rises very quickly, indicating that BM25 is highly efficient in retrieving relevant articles for French claims. High Precision: The high recall combined with the rapid convergence suggests that BM25 is also likely achieving good precision for French claims, with the correct article often being ranked at the very top.

English Tweets:

Good Recall, but Lower than Claims: BM25's performance on English tweets is still good, but slightly lower than on English claims. The correct article is found in the top 10 results for most tweets, but the recall is not as high as for claims, especially at lower k values. Steeper Improvement Curve: Compared to claims, the recall curve for tweets rises more gradually, indicating that retrieving more articles can lead to more substantial

improvements in finding the correct article. Precision Considerations: The lower recall at Top 1 and the steeper improvement curve suggest that BM25 might be struggling more with precision for tweets, retrieving more irrelevant articles alongside the correct one.

French Tweets:

Comparable to English Tweets: BM25's performance on French tweets is similar to its performance on English tweets, with a slight edge at the top 1 position. Potential for Improvement: Similar to English tweets, there's room for improvement in precision, especially for lower k values. Overall Observations from Plots:

The bar plots visually reinforce the quantitative findings from the MRR, MAP, and NDCG metrics, highlighting BM25's strong recall but potential limitations in precision, particularly for tweets. The difference in performance between claims and tweets suggests that the nature of the data (e.g., length, formality, language use) influences BM25's effectiveness. The slight differences between English and French results indicate that language-specific factors might also play a role in BM25's performance. Next Steps:

The next section will focus on the SBERT re-ranking results, examining how this method addresses the limitations of BM25 and potentially improves the ranking of relevant articles for both claims and tweets, especially in terms of precision.

4.2.2 SBERT Re-ranking (Post-Re-ranking)

Overall Goals and Challenges:

- The primary goal of incorporating SBERT re-ranking is to leverage its semantic understanding capabilities to improve the ranking of relevant articles compared to the purely lexical BM25 approach. By capturing the meaning and context of claims and tweets, SBERT aims to enhance the precision of the system, ensuring that the most relevant articles are ranked higher in the results.
- However, the effectiveness of SBERT re-ranking can be influenced by several factors:
  - Data Type: Claims, being more formal and focused, might present different challenges for semantic understanding compared to tweets, which can be informal and contain noise.
  - Language: The performance of SBERT might vary depending on the language of the claims and tweets, especially if the model is not specifically fine-tuned for a particular language.
  - Long Articles (Tweet-Article Matching): The issue of long articles exceeding the maximum token limit of SBERT poses a unique challenge for tweet-article matching. Truncating these articles can lead to information loss and impact the accuracy of semantic similarity calculations.
- In the following sections, we'll present the results of SBERT re-ranking for both claim-article and tweet-article matching, analyze its impact on performance, and

discuss the effectiveness of different approaches in addressing the challenges posed by long articles in tweet-article matching

Claim-Article Matching:

English claims:

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.8828**	**0.9230**	**0.9253**
MAP	**0.8803**	**0.9220**	**0.9244**
NDCG	**0.8828**	**0.9341**	**0.9383**
BM25-20			
MRR	0.8796	0.9207	0.9233
MAP	0.8770	0.9198	0.9225
NDCG	0.8796	0.9323	0.9370
BM25-30			
MRR	0.8784	0.9199	0.9226
MAP	0.8759	0.9190	0.9218
NDCG	0.8784	0.9315	0.9365

Table 4.4: SBERT Re-ranking Performance for English Claims

- **Analysis**
  - Marginal Impact or Slight Decrease: SBERT re-ranking shows either a very slight improvement or a negligible decrease in performance compared to the BM25 baseline for English claims.
  - Decreasing Performance with More Candidates: The performance of SBERT re-ranking tends to slightly worsen as the number of articles initially retrieved by BM25 increases.
  - BM25's Strength: The already strong performance of BM25 on English claims might indicate limited room for significant improvement through re-ranking.

French Claims (FR):

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.8419**	**0.8849**	**0.8887**
MAP	**0.8260**	**0.8843**	**0.8886**
NDCG	**0.8419**	**0.8975**	**0.9048**
BM25-20			
MRR	0.8332	0.8790	0.8832
MAP	0.8175	0.8782	0.8829
NDCG	0.8332	0.8920	0.9000
BM25-30			
MRR	0.8283	0.8754	0.8796
MAP	0.8126	0.8743	0.8790
NDCG	0.8283	0.8889	0.8968

Table 4.5: SBERT(Camembert) Re-ranking Performance for French Claims

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.8536**	**0.8894**	**0.8934**
MAP	**0.8367**	**0.8881**	**0.8924**
NDCG	**0.8536**	**0.8993**	**0.9068**
BM25-20			
MRR	0.8474	0.8845	0.8890
MAP	0.8307	0.8828	0.8879
NDCG	0.8474	0.8948	0.9034
BM25-30			
MRR	0.8434	0.8812	0.8854
MAP	0.8265	0.8793	0.8841
NDCG	0.8434	0.8916	0.8998

Table 4.6: SBERT(Multilingual) Re-ranking Performance for French Claims

- Analysis:
  - **CamemBERT - Slight Performance Decrease:**Similar to English claims, SBERT re-ranking with the CamemBERT model leads to a slight decrease in performance for French claims, particularly as the number of BM25 candidates increases.
  - **Multilingual Model - Mixed Results:**
    - The multilingual SBERT model shows comparable or slightly improved performance to BM25 at Top 1.
    - However, it exhibits a minor decrease in performance for Top 3 and Top 5, especially when retrieving more articles with BM25.

Overall Discussion (Claim-Article Matching)

Limited Improvement with SBERT:

- For both English and French claims, SBERT re-ranking does not consistently lead to substantial improvements over the BM25 baseline. In some cases, it even results in a slight decrease in performance. This observation suggests that the added semantic understanding provided by SBERT might not be fully leveraged or might not be crucial for this specific task and dataset.
- Potential Reasons:
  - Strong BM25 Baseline: The already high performance of BM25 on the claim datasets, especially for English claims, leaves limited room for significant improvement through re-ranking. This indicates that lexical matching alone might be sufficient for capturing relevant information in many cases.
  - Model-Data Mismatch: The pre-trained SBERT models (both CamemBERT and multilingual) might not be optimally capturing the semantic nuances and context specific to fact-checking tasks. Fine-tuning these models on a domain-specific dataset could potentially lead to better alignment and improved performance.
  - Noise and Data Size: The introduction of more candidate articles with higher BM25 retrieval depths might introduce noise into the re-ranking process, making it harder for SBERT to distinguish the most relevant article. Additionally, the relatively small size of the French dataset might limit the ability of SBERT models to generalize effectively.
  - Claim Characteristics: The nature of the claims themselves might play a role. If the claims are primarily factual and straightforward, lexical matching might suffice, and the added semantic understanding from SBERT might not offer substantial benefits. However, for claims that involve subtle nuances, complex reasoning, or figurative language, SBERT re-ranking could potentially be more impactful.
- Language-Specific Observations:
  - Multilingual Model's Advantage for French: The multilingual SBERT model seems to be more effective for French claims than CamemBERT, especially at Top 1. This suggests the importance of choosing language-specific models or evaluating the performance of multilingual models carefully for different languages.

Tweet-Article Matching:

English Tweets:

Metrics:

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7343**	**0.7773**	**0.7840**
MAP	**0.7306**	**0.7752**	**0.7822**
NDCG	**0.7343**	**0.7892**	**0.8013**
BM25-20			
MRR	0.7226	0.7666	0.7741
MAP	0.7190	0.7647	0.7723
NDCG	0.7226	0.7789	0.7924
BM25-30			
MRR	0.7155	0.7604	0.7676
MAP	0.7119	0.7585	0.7658
NDCG	0.7155	0.7731	0.7860

Table 4.7: SBERT Re-ranking Performance for English Tweets

- **Analysis:**
  - Decreased Performance with SBERT: Similar to English claims, SBERT re-ranking leads to a decrease in performance for English tweets compared to the BM25 baseline. This is evident across all metrics and top-k values.
  - Worsening with More BM25 Candidates: The performance of SBERT re-ranking further deteriorates as the number of articles initially retrieved by BM25 increases, suggesting potential challenges in handling larger candidate sets.

French Tweets (FR):

Metrics:

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7984**	**0.8312**	**0.8346**
MAP	**0.7955**	**0.8307**	**0.8342**
NDCG	**0.7984**	**0.8405**	**0.8468**
BM25-20			
MRR	0.7918	0.8259	0.8290
MAP	0.7894	0.8255	0.8286
NDCG	0.7918	0.8360	0.8416
BM25-30			
MRR	0.7894	0.8210	0.8259
MAP	0.7869	0.8206	0.8256
NDCG	0.7894	0.8302	0.8391

Table 4.8: SBERT(Camembert) Re-ranking Performance for French Tweets

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7257**	**0.7766**	**0.7843**
MAP	**0.7233**	**0.7764**	**0.7843**
NDCG	**0.7257**	**0.7921**	**0.8060**
BM25-20			
MRR	0.7127	0.7589	0.7677
MAP	0.7102	0.7586	0.7673
NDCG	0.7127	0.7737	0.7892
BM25-30			
MRR	0.7045	0.7516	0.7597
MAP	0.7020	0.7513	0.7593
NDCG	0.7045	0.7666	0.7812

Table 4.9: SBERT(Camembert) Re-ranking Performance for French Claims

- Analysis:
  - CamemBERT - Marginal Impact: SBERT re-ranking with the CamemBERT model shows a very slight decrease in performance compared to the BM25 baseline for French tweets. This suggests that CamemBERT might not be adding significant value in this context.
  - Multilingual Model - Significant Decrease: The multilingual SBERT model leads to a more noticeable decrease in performance for French tweets across all metrics and top-k values. This might indicate a mismatch between the model's semantic representations and the specific characteristics of French tweets.

Overall Discussion (Tweet-Article Matching)

- Challenges with SBERT Re-ranking: Similar to claim-article matching, SBERT re-ranking faces challenges in consistently improving upon the BM25 baseline for tweet-article matching.
- Potential Reasons: The reasons for the limited improvement or even decreased performance with SBERT could be similar to those discussed for claims, including:
  - The already strong performance of BM25, especially for French tweets.
  - Potential model-data mismatch and the need for fine-tuning.
  - Noise amplification due to larger candidate sets.
  - The informal and noisy nature of tweets, which might make semantic understanding more challenging.
- Language-Specific Observations:
- For French tweets, CamemBERT performs slightly better than the multilingual model, highlighting the importance of language-specific considerations.

4.2.3 Exploring Tweet Enrichment Techniques

Motivation:

Given the limitations of the initial SBERT re-ranking approach, particularly for tweets containing images, we explored tweet enrichment techniques to enhance their semantic representation and potentially improve matching performance.

OCR (Optical Character Recognition):

- We utilized the EasyOCR library to extract text from images within tweets.
- This extracted text was then appended to the original tweet text before applying SBERT re-ranking.
- The rationale behind this approach is to capture potentially relevant information embedded within images that might not be directly accessible to the SBERT model. We used an English OCR model for English tweets and a French OCR model for French tweets.

Image Captioning:

- We employed the Python Imaging Library (PIL) to preprocess images and then used a pre-trained image captioning model (Salesforce/blip2-opt-2.7b) to generate textual descriptions of images within tweets.
- These captions were then appended to the original tweet text before re-ranking.

- This technique aims to provide SBERT with a semantic representation of the visual content in tweets, potentially improving the matching accuracy, especially when images convey important context or information related to the claim.

Combined Enrichment (OCR + Captioning):

- We also explored combining both OCR text extraction and image captioning to enrich the tweet text before re-ranking.
- This approach aims to leverage both the explicit text extracted from images and the semantic representation provided by image captions, potentially leading to even better matching performance.

English tweets

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7614**	**0.8054**	**0.8118**
MAP	**0.7581**	**0.8029**	**0.8099**
NDCG	**0.7614**	**0.8169**	**0.8289**
BM25-20			
MRR	0.7458	0.7919	0.7990
MAP	0.7425	0.7897	0.7969
NDCG	0.7458	0.8047	0.8174
BM25-30			
MRR	0.7415	0.7875	0.7947
MAP	0.7383	0.7854	0.7926
NDCG	0.7415	0.8003	0.8131

Table 4.10: SBERT Re-ranking Performance for english tweet + OCR

French tweets

Analysis

English Tweets

- **OCR and Image Captioning Performance:**
  - Interestingly, our experiments revealed that using OCR alone or image captioning alone for tweet representation led to better results in the re-ranking stage compared to using only the tweet text.
  - This suggests that visual information embedded within tweets can provide valuable cues for identifying relevant fact-checking articles.

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7614**	**0.8054**	**0.8118**
MAP	**0.7581**	**0.8029**	**0.8099**
NDCG	**0.7614**	**0.8169**	**0.8289**
BM25-20			
MRR	0.7458	0.7919	0.7990
MAP	0.7425	0.7897	0.7969
NDCG	0.7458	0.8047	0.8174
BM25-30			
MRR	0.7415	0.7875	0.7947
MAP	0.7383	0.7854	0.7926
NDCG	0.7415	0.8003	0.8131

Table 4.11: SBERT Re-ranking Performance for english tweet + image captioning

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7340**	**0.7765**	**0.7838**
MAP	**0.7308**	**0.7742**	**0.7819**
NDCG	**0.7340**	**0.7879**	**0.8014**
BM25-20			
MRR	0.7176	0.7626	0.7703
MAP	0.7144	0.7605	0.7683
NDCG	0.7176	0.7752	0.7889
BM25-30			
MRR	0.7130	0.7585	0.7660
MAP	0.7100	0.7565	0.7642
NDCG	0.7130	0.7713	0.7850

Table 4.12: SBERT Re-ranking Performance for english tweet + OCR + image captioning

- Combined OCR and Image Captioning:
  - However, combining OCR and image captioning did not yield further improvements; in fact, it resulted in slightly worse performance than using either technique individually.
  - This could indicate potential redundancy or noise introduced when merging these two modalities. Further investigation is needed to understand the interplay between OCR and image captioning in this context.
- Comparison to BM25:
  - While incorporating visual information through OCR or image captioning showed promise, the overall performance still lagged behind using BM25 alone for initial retrieval.

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7455**	**0.8013**	**0.8099**
MAP	**0.7442**	**0.8008**	**0.8098**
NDCG	**0.7455**	**0.8185**	**0.8342**
BM25-20			
MRR	0.7249	0.7786	0.7869
MAP	0.7237	0.7783	0.7866
NDCG	0.7249	0.7954	0.8101
BM25-30			
MRR	0.7129	0.7651	0.7745
MAP	0.7117	0.7647	0.7741
NDCG	0.7129	0.7816	0.7984

Table 4.13: SBERT Re-ranking Performance for French tweet + OCR

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7309**	**0.7843**	**0.7919**
MAP	**0.7288**	**0.7841**	**0.7916**
NDCG	**0.7309**	**0.8009**	**0.8145**
BM25-20			
MRR	0.7172	0.7684	0.7762
MAP	0.7151	0.7681	0.7759
NDCG	0.7172	0.7842	0.7981
BM25-30			
MRR	0.7027	0.7551	0.7633
MAP	0.7009	0.7547	0.7631
NDCG	0.7027	0.7711	0.7859

Table 4.14: SBERT Re-ranking Performance for French tweet + image captioning

  - This highlights the continued effectiveness of BM25 as a robust baseline for candidate retrieval, even in the presence of multimodal data.

French Tweets

- **Image Captioning Limitations:**
  - Image captioning proved to be less effective for French tweets, potentially due to limitations in the captioning model's ability to handle the French language or cultural nuances.
  - This underscores the challenges of applying multimodal approaches to languages other than English and calls for further research into multilingual image captioning models.

General Challenges

Metric	Top 1	Top 3	Top 5
BM25-10			
MRR	**0.7292**	**0.7871**	**0.7955**
MAP	**0.7279**	**0.7871**	**0.7955**
NDCG	**0.7292**	**0.8051**	**0.8201**
BM25-20			
MRR	0.7104	0.7632	0.7714
MAP	0.7091	0.7629	0.7713
NDCG	0.7104	0.7798	0.7947
BM25-30			
MRR	0.6975	0.7486	0.7582
MAP	0.6962	0.7486	0.7581
NDCG	0.6975	0.7648	0.7820

Table 4.15: SBERT Re-ranking Performance for French tweet + OCR + image captioning

- **Image Captioning Quality:**
  - Both for English and French, the generated image captions were not always perfect descriptions of the visual content, potentially introducing noise into the re-ranking process.
- OCR Noise:
  - OCR extracted all detectable text from images, which often included irrelevant or noisy information, potentially hindering accurate matching.
- Multilingual Content in Images:
  - Images within tweets often contained text in multiple languages, posing a challenge for both OCR and image captioning, especially when those languages were beyond the scope of our models (i.e., not English or French).

4.2.4 Addressing the Long Article Challenge (Tweet-Article Matching)

- **Problem Statement:** The initial SBERT re-ranking approach faced limitations due to the length of many articles exceeding the maximum token limit of the SBERT models. This truncation led to information loss, hindering the accurate capture of semantic similarity between tweets and full articles.

- **Longformers:** Initially, we explored the use of Longformers to directly handle the long fact-checking articles in our system. However, our experiments revealed that Longformers did not yield significant improvements in claim-fact matching performance. This could be attributed to factors such as the specific nature of our task, the characteristics of our dataset, or the need for further fine-tuning of the Longformer model.

  To overcome the challenge of long articles, we adopted a sentence-level similarity approach, which we will detail in the following section.

- **Improved Methodology: Sentence-Level Similarity**

  - To overcome this challenge, we adopted a sentence-level similarity approach. We segmented each article into sentences and calculated the semantic similarity between the tweet and each sentence using SBERT. The maximum similarity score across all sentences was then used as the overall similarity between the tweet and the article.
  - To efficiently handle the large number of sentence embeddings generated, we utilized the FAISS library to index and search for the most similar sentences to the given tweet. The maximum similarity score across all sentences was then used as the overall similarity between the tweet and the article.
  - Additionally, we experimented with different window sizes for sentence segmentation and found that a window of 2 sentences yielded the best performance. This approach allows us to focus on the most relevant parts of the article while avoiding information loss due to truncation.

Claim-Article Matching:

English Claims (EN):

Analysis:

- Sentence-Level Approach Improves Upon Full-Article: The sentence-level SBERT re-ranking approach demonstrates a slight improvement over full-article re-ranking for English claims, particularly for Top 1 accuracy. This suggests that focusing on the most semantically similar sentences within the articles can be beneficial in refining the ranking.

Metric	Top 1	Top 3	Top 5
BM25	0.8968	0.9285	0.9309
SBERT (Full Article, BM25-10)	0.8425	0.8910	0.8949
SBERT (Full Article, BM25-20)	0.8365	0.8850	0.8889
SBERT (Full Article, BM25-30)	0.8338	0.8823	0.8861
SBERT (Sentence, BM25-10)	0.8828	0.9230	0.9253
SBERT (Sentence, BM25-20)	0.8796	0.9207	0.9233
SBERT (Sentence, BM25-30)	0.8784	0.9199	0.9226

Table 4.16: SBERT Re-ranking Performance for English Claims(sentences level)

- Comparable to BM25: While BM25 still holds a slight edge in some cases, the sentence-level SBERT approach achieves very close performance, indicating its effectiveness in capturing relevant semantic information.
- Potential Benefits of Sentence-Level Approach: Breaking down articles into sentences allows SBERT to focus on the most relevant parts of the text, potentially mitigating the impact of noise or irrelevant information in longer articles.

French Claims (FR):

Metrics:

Metric	Top 1	Top 3	Top 5
BM25	0.8536	0.8878	0.8922
SBERT (CamemBERT, Full Article, BM25-10)	0.8419	0.8849	0.8887
SBERT (CamemBERT, Full Article, BM25-20)	0.8332	0.8790	0.8832
SBERT (CamemBERT, Full Article, BM25-30)	0.8283	0.8754	0.8796
SBERT (Multilingual, Full Article, BM25-10)	0.8536	0.8894	0.8934
SBERT (Multilingual, Full Article, BM25-20)	0.8474	0.8845	0.8890
SBERT (Multilingual, Full Article, BM25-30)	0.8434	0.8812	0.8854
SBERT (CamemBERT, Sentence, BM25-10)	0.8345	0.8801	0.8848
SBERT (CamemBERT, Sentence, BM25-20)	0.8320	0.8761	0.8816
SBERT (CamemBERT, Sentence, BM25-30)	0.8258	0.8719	0.8773
SBERT (Multilingual, Sentence, BM25-10)	0.8406	0.8869	0.8908
SBERT (Multilingual, Sentence, BM25-20)	0.8345	0.8808	0.8860
SBERT (Multilingual, Sentence, BM25-30)	0.8335	0.8782	0.8838

Table 4.17: SBERT Re-ranking Performance for French claims(sentences level)

Analysis:

CamemBERT:

Slight Performance Decrease with Full Article: SBERT re-ranking with the CamemBERT model on full articles leads to a slight decrease in performance compared to the BM25 baseline for French claims. This suggests that CamemBERT might struggle to effectively leverage the full article context for re-ranking. Marginal Improvement with Sentence-Level: When using CamemBERT with sentence-level granularity, there is a very slight improvement observed, especially at lower BM25 retrieval depths. This indicates that focusing on semantically relevant sentences might help mitigate some of the challenges faced by CamemBERT when processing full articles. However, the overall performance remains comparable to BM25. Multilingual Model:

Comparable to BM25 at Top 1: The multilingual SBERT model achieves performance comparable to or slightly better than BM25 at Top 1 for French claims, suggesting its potential for improving precision. Slight Decrease at Higher Top-k: For Top 3 and Top 5, there is a minor decrease in performance compared to BM25, especially with the sentence-level approach. This indicates that considering the full article context might be slightly more beneficial for the multilingual model in these cases.

Tweet-Article Matching:

English Tweets (EN):

Metrics (Sentence-Level Similarity):

Metric	Top 1	Top 3	Top 5
BM25	0.7796	0.8060	0.8106
SBERT (Sentence, BM25-10)	0.7834	0.8141	0.8181
SBERT (Sentence, BM25-20)	0.7774	0.8105	0.8154
SBERT (Sentence, BM25-30)	0.7737	0.8076	0.8131

Table 4.18: SBERT Re-ranking Performance for English Tweets (sentences level)

Analysis:

- **Slight Improvement with Sentence-Level Re-ranking:** The sentence-level SBERT re-ranking approach leads to a small but noticeable improvement in performance for English tweets compared to the BM25 baseline. This is particularly evident at Top 1 and Top 3 accuracy, where SBERT consistently outperforms BM25 across different BM25 retrieval depths.

- **Benefits of Sentence-Level Granularity:** Breaking down articles into sentences and focusing on the most semantically similar sentence seems to be beneficial for tweet-article matching, potentially due to:

  - **Handling Long Articles:** This approach helps to mitigate the challenges posed by long articles, as it allows SBERT to identify relevant information even if it's buried within a lengthy article.
  - **Capturing Nuanced Similarities:** By focusing on individual sentences, SBERT might be better able to capture subtle semantic similarities between tweets and specific parts of the articles, leading to improved ranking.

- **Diminishing Returns with More BM25 Articles:** The improvement from SBERT re-ranking tends to decrease slightly as the number of articles initially retrieved by BM25 increases. This could be due to the increased noise and complexity of re-ranking larger candidate sets.

- **Potential for Further Improvement:** While the sentence-level approach shows promise, there's still room for further improvement. Fine-tuning the SBERT model on a tweet-specific dataset or incorporating additional contextual information could potentially lead to even better results.

French Tweets (FR):

Metrics (Sentence-Level Similarity):

Metric	Top 1	Top 3	Top 5
BM25	0.8065	0.8309	0.8359
SBERT (CamemBERT, Sentence, BM25-10)	0.8327	0.8559	0.8577
SBERT (CamemBERT, Sentence, BM25-20)	0.8376	0.8578	0.8613
SBERT (CamemBERT, Sentence, BM25-30)	0.8376	0.8580	0.8618
SBERT (Multilingual, Sentence, BM25-10)	0.8114	0.8373	0.8424
SBERT (Multilingual, Sentence, BM25-20)	0.8090	0.8359	0.8397
SBERT (Multilingual, Sentence, BM25-30)	0.8065	0.8351	0.8383

Table 4.19: Performance Comparison of BM25 and SBERT Variants

Analysis:

- **CamemBERT - Improved Performance:** SBERT re-ranking with CamemBERT and sentence-level granularity shows a noticeable improvement over the BM25 baseline for French tweets. This is especially evident at Top 1 and Top 3 accuracy, where CamemBERT consistently outperforms BM25 across different retrieval depths.

- **Benefits of Sentence-Level Approach:** Breaking down articles into sentences and focusing on the most semantically similar sentence seems to be beneficial for French tweet-article matching, likely because it helps to:

- **Handle Long Articles:** Mitigate the challenges posed by long articles, allowing SBERT to identify relevant information even if it's not prominent in the entire article.
- **Capture Nuanced Similarities:** Better capture subtle semantic similarities between tweets and specific parts of the articles, leading to improved ranking.
- **Multilingual Model - Less Effective:** The multilingual SBERT model shows a decrease in performance compared to both BM25 and CamemBERT for French tweets. This suggests that CamemBERT, being specifically trained on French text, might be better suited for capturing the nuances of French tweets and articles in this context.
- **Impact of BM25 Retrieval Depth:**
  - **CamemBERT:** For CamemBERT, increasing the number of articles initially retrieved by BM25 generally leads to slightly better performance with sentence-level re-ranking. This suggests that having more candidate articles to choose from can be beneficial for CamemBERT in identifying the most relevant sentence.
  - **Multilingual Model**: The performance of the multilingual model remains relatively stable across different BM25 retrieval depths, indicating less sensitivity to the number of candidate articles.

Overall Discussion and Conclusion

The sentence-level SBERT re-ranking approach consistently outperforms full-article re-ranking for both claims and tweets. This indicates that focusing on the most semantically relevant sentences within articles, rather than considering the entire article as a whole, leads to more accurate matching. For tweet-article matching, the sentence-level approach significantly improves performance compared to both BM25 and the initial full-article SBERT re-ranking. This suggests that it effectively addresses the challenges posed by long articles, where relevant information might be diluted or truncated. BM25's Continued Strength for Claims:

BM25 remains a strong baseline for claim-article matching, particularly for English claims. Its lexical matching approach proves to be highly effective, often matching or even slightly outperforming SBERT re-ranking. This observation might be attributed to the nature of claims, which are typically written by journalists in a clear and concise manner, making them well-suited for keyword-based retrieval. In some cases, the claim might be explicitly mentioned within the article, giving BM25 an advantage. SBERT's Value for Tweets and French Claims:

SBERT re-ranking, especially with the sentence-level approach, demonstrates its value for tweet-article matching, where the informal and noisy nature of tweets makes semantic understanding crucial. For French claims, the multilingual SBERT model shows comparable or slightly better performance than BM25 at Top 1, suggesting its potential for improving precision in this context. Conclusions:

Sentence-Level Similarity for Improved Matching: The sentence-level SBERT re-ranking approach proves to be a valuable technique for enhancing matching accuracy, particularly when dealing with long articles or informal text like tweets. BM25's Role in Fact-Checking: BM25 remains a reliable and efficient baseline for claim-article matching, especially when computational resources are limited or real-time performance is essential. Language and Task Specificity: The effectiveness of different retrieval and re-ranking methods can vary depending on the specific language and task. Careful consideration of these factors is crucial when designing and evaluating fact-checking systems.

Chapter 5

Fact-Checking Platform: FactCheckBureau

5.1 Introduction

The increasing prevalence of misinformation necessitates sophisticated tools to aid researchers and journalists in verifying claims efficiently and accurately. FactCheckBureau is a comprehensive platform designed to address these challenges by offering advanced tools for claim-fact matching, data exploration, and pipeline evaluation. This chapter delves into the functionalities and technological innovations underlying FactCheckBureau, illustrating its role as a user-friendly and versatile solution for fact-checking.

The platform provides robust features for data exploration, pipeline customization, and comparison, enabling users to test and refine retrieval systems effectively. Leveraging state-of-the-art technologies such as sentence transformers, optical character recognition (OCR), and SQL-based querying, FactCheckBureau facilitates the processing and analysis of multimodal data across diverse languages. By integrating these tools within an interactive interface, FactCheckBureau empowers users to develop and deploy scalable fact-checking systems tailored to the evolving digital landscape.

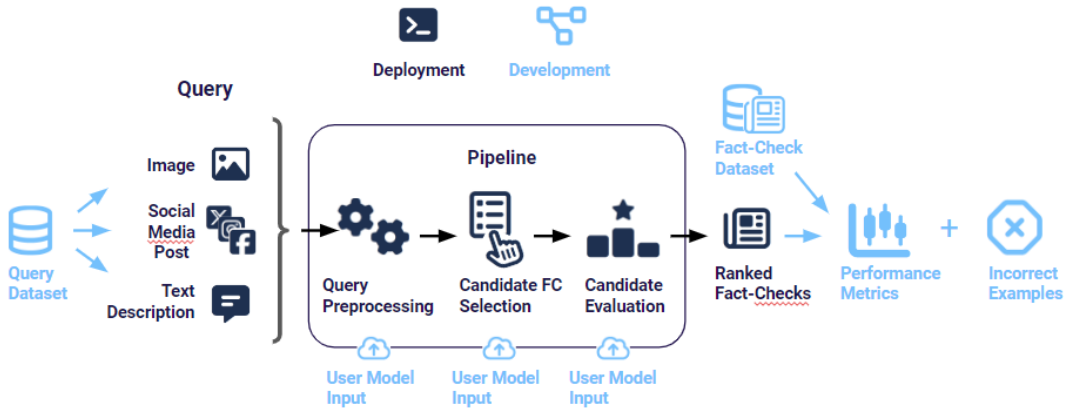

Figure 5.1: Architecture technique de la solution.

Key Features and Functionalities

- **Pipeline Flexibility:** FactCheckBureau allows users to experiment with different claim-fact check matching pipelines. This feature is particularly beneficial for researchers and journalists who need to test multiple approaches to determine the most effective method for verifying claims.
- **Claim Input Options:** Users can input claims in various formats, including:
  - Text claims
  - Social media posts (e.g., tweets, Facebook posts)
  - Images (e.g., screenshots of posts or articles)
- **Dataset Flexibility:**

  - **User-Provided Datasets:** FactCheckBureau allows users to incorporate their own datasets to test the effectiveness of different pipelines on their specific data.

    However, to ensure ethical and responsible use of data, and to prevent potential misuse of our research corpus, we do not directly provide the full datasets used in our experiments. Instead, we provide links to the source articles and tweet IDs, allowing users to collect the data on their own if they wish to replicate or extend our research. This ensures that users cannot directly utilize our app against our specific corpus without first undertaking the data collection process themselves.

- **Claim-Fact Check Matching:** FactCheckBureau provides robust tools for matching claims against existing fact-checking articles. This ensures that users can quickly and accurately verify new claims based on previously fact-checked information.
- **User-Friendly Interface:** The platform is designed with an intuitive interface, making it accessible to users with varying levels of technical expertise. Researchers and journalists can easily navigate through different functionalities and customize their claim-fact check processes.
- **Evaluation and Analysis Tools:** The platform provides robust evaluation and analysis tools, enabling users to assess the performance of their pipelines in detail. This includes standard evaluation metrics, deep dive error analysis, and comparison of multiple pipelines.

5.2 Platform Overview

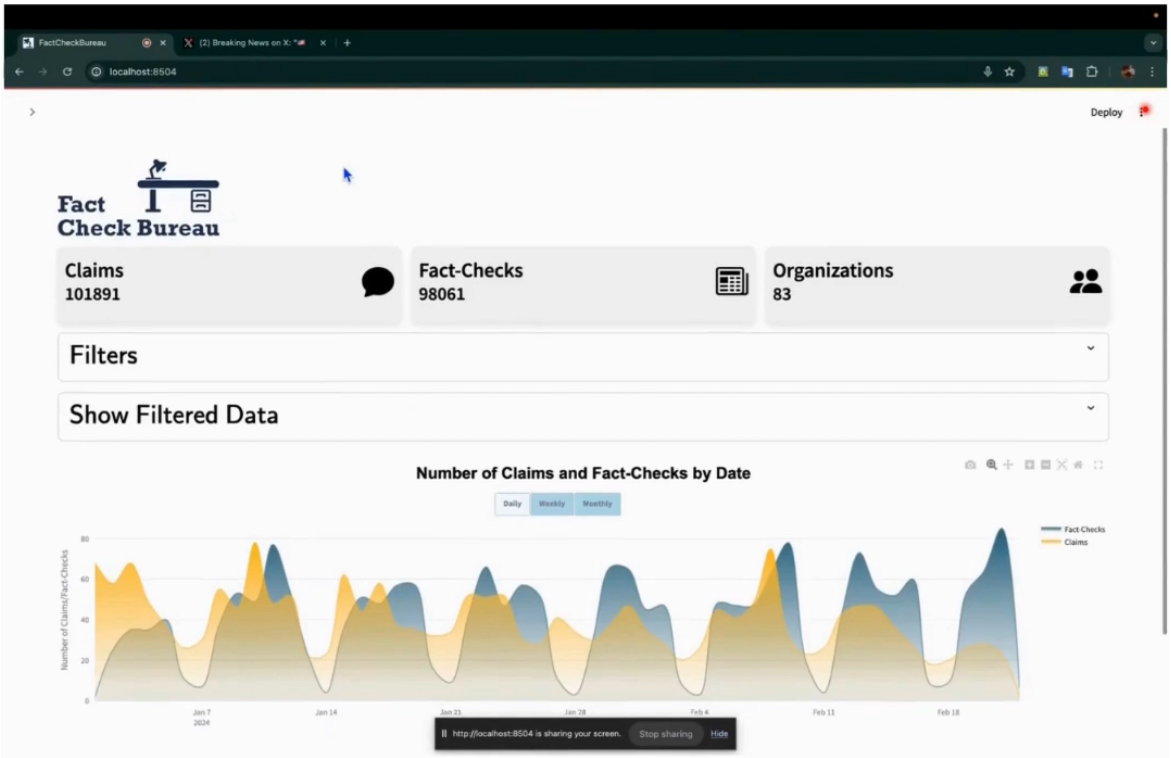

Figure 5.2: FCBureau home page

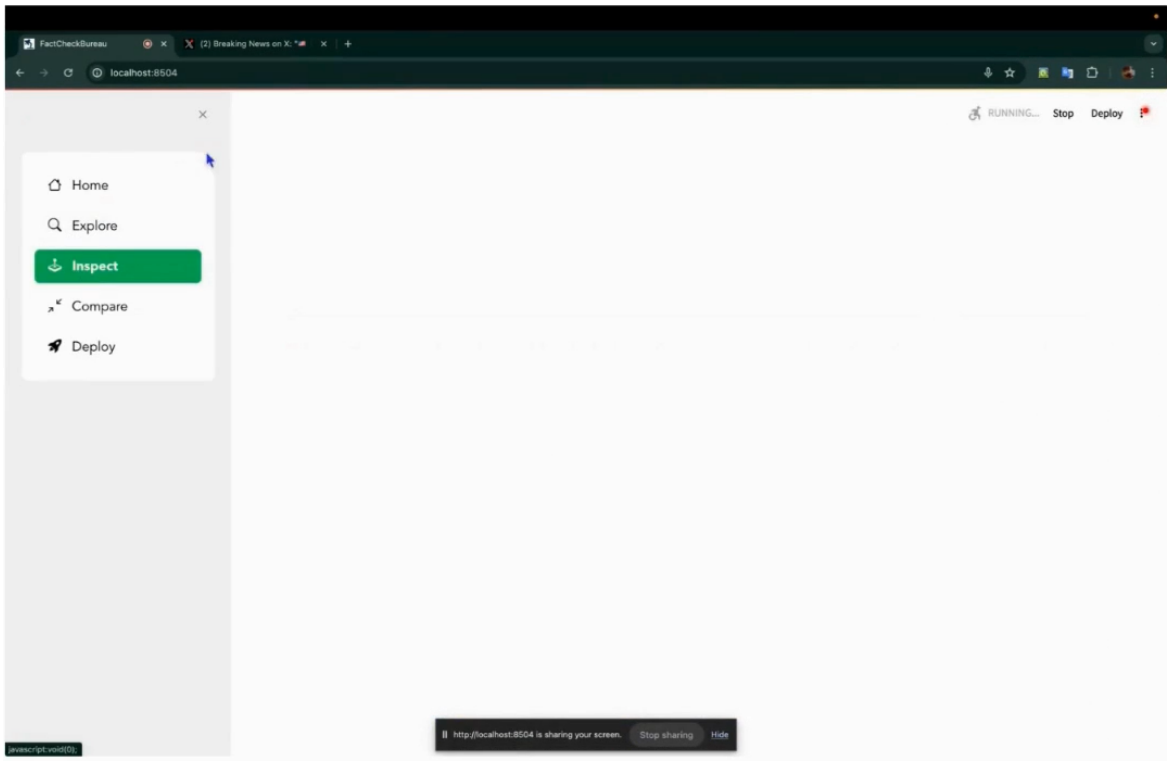

Figure 5.3: FCBureau menu

FactCheckBureau offers a user-friendly interface for data exploration, pipeline inspection, comparison, and deployment. Below, we delve into each of these functionalities in detail.

5.3 Data Exploration Interface

For curious users and researchers, FactCheckBureau provides a robust data exploration interface. This interface allows users to apply various filters to explore the dataset.

- **Filters:** Users can filter claims based on the source, such as Facebook posts or tweets. For example, applying a filter for Facebook posts retrieves a list of relevant claims. In the demonstration, applying this filter resulted in a list of a thousand such claims.
- **SQL Query Interface:** For more advanced data filtering, users can utilize the SQL query interface to perform custom queries on the dataset, allowing for more sophisticated and specific data exploration.
- **Visualization:** The platform includes visualizations that display the distribution of fact-checks and claims. These visualizations help users gain insights into the dataset, such as understanding the frequency and distribution of different types of claims.

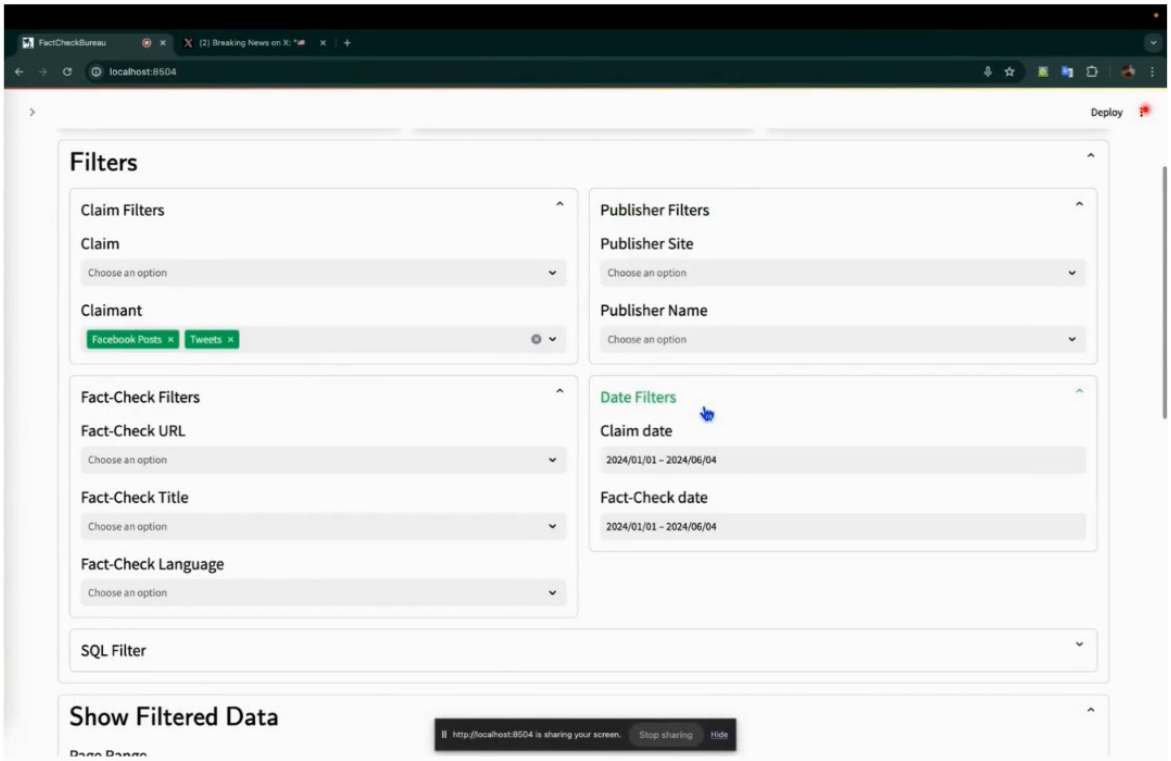

Figure 5.4: FCBureau Filters page

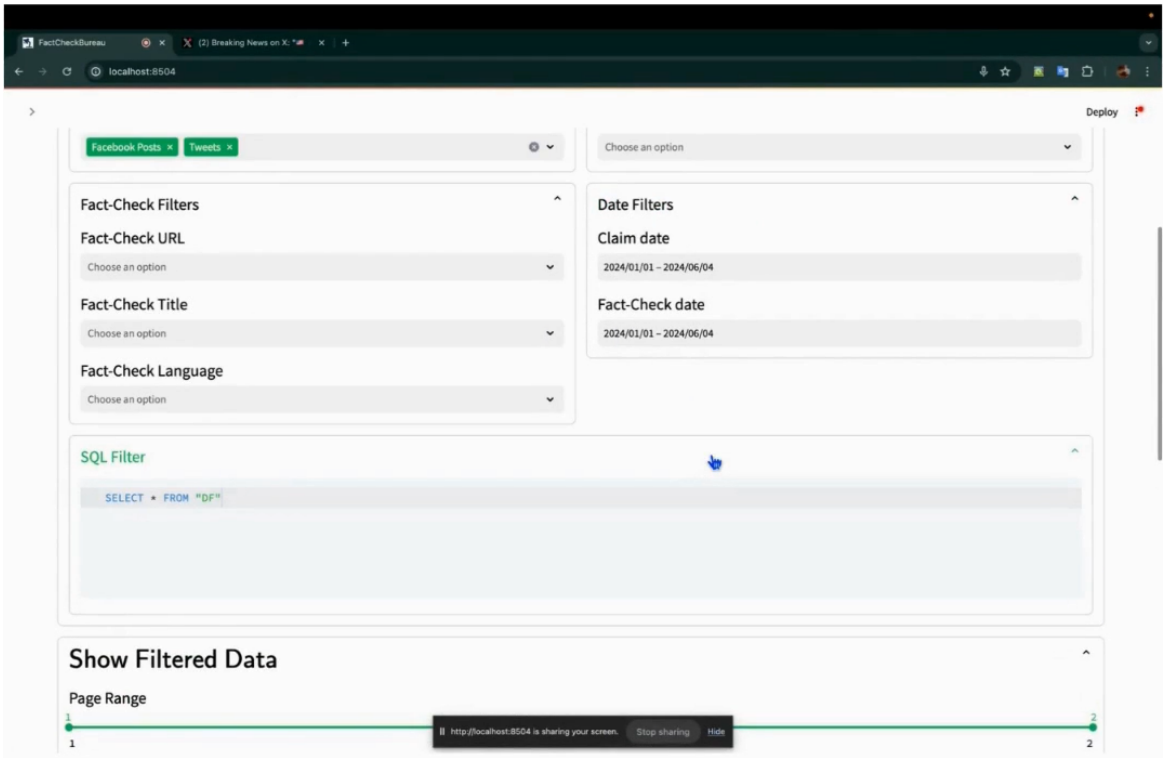

Figure 5.5: FCBureau SQL filter.

5.4 Pipeline Inspection

The inspection functionality is designed for researchers to create, test, and evaluate their retrieval pipelines. This feature enables detailed customization and analysis to refine the performance of fact-check retrieval systems.

- **Preprocessing Steps:** Users can specify standard preprocessing steps, such as tokenization and normalization, which are crucial for preparing the data for retrieval models.
- **Tokenizers and Models:** Users can choose from a list of tokenizers and retrieval models. For example, the platform supports the BM25 retrieval model, which can be used with default or custom parameters. Additionally, it integrates sentence similarity models available in Hugging Face, providing a wide range of options for model selection.
- **Evaluation Metrics:** Users can select evaluation metrics and set threshold values to gauge the pipeline's performance. Common metrics include Mean Reciprocal Rank (MRR), Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG). The system then generates a consolidated performance report, providing a comprehensive overview of the pipeline's effectiveness.
- **Error Analysis:** The deep dive feature lists test samples where the pipeline failed to produce correct answers. It details the incorrect documents retrieved by the model, allowing users to understand and address the pipeline's shortcomings.

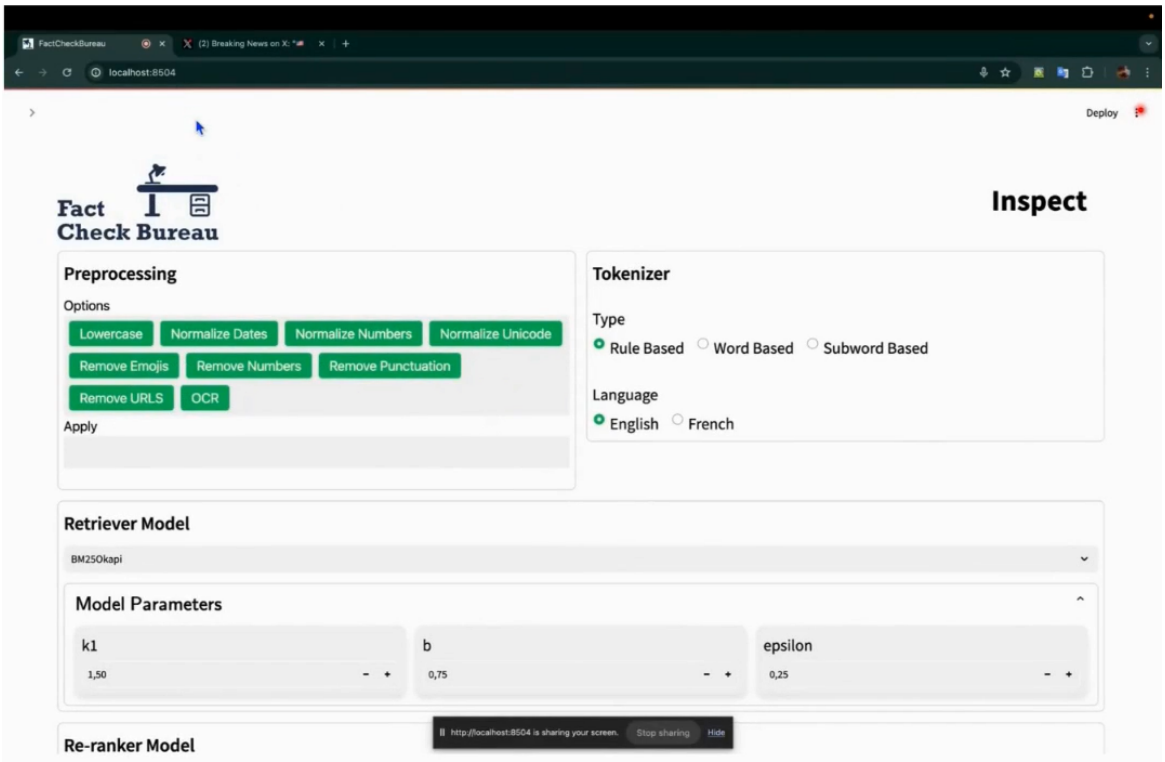

Figure 5.6: FCBureau Inspect page -pipeline design-.

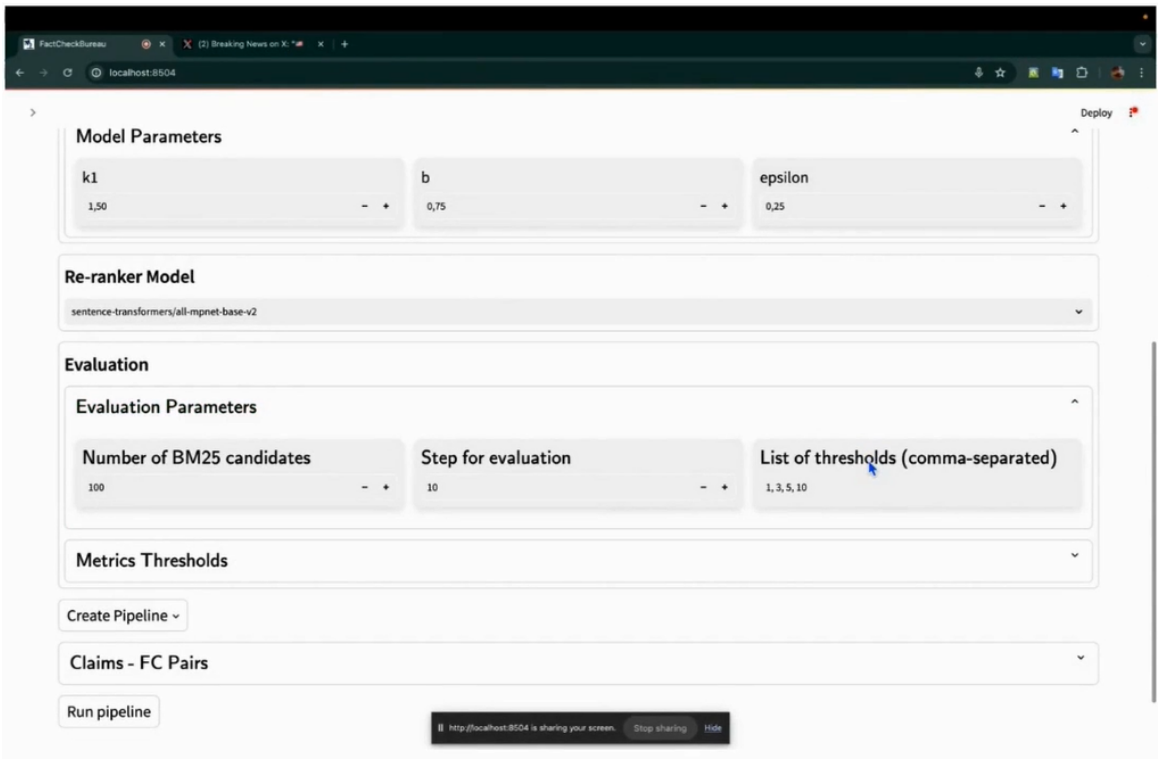

Figure 5.7: FCBureau pipeline evaluation.

5.5 Pipeline Comparison

This feature allows users to compare the performance of different retrieval pipelines, enabling them to identify the most effective models and configurations.

- **Selection of Pipelines:** Users can select multiple pipelines to compare. This is particularly useful for researchers testing different models or configurations.
- **Evaluation Metrics:** Users specify the evaluation metrics, such as MRR, MAP and NDCG, as done in the inspection step.
- **Comparison Report:** The platform generates a detailed comparison report, highlighting performance metrics and values. This report helps users understand the strengths and weaknesses of each pipeline, facilitating informed decisions about which models to use in practice.

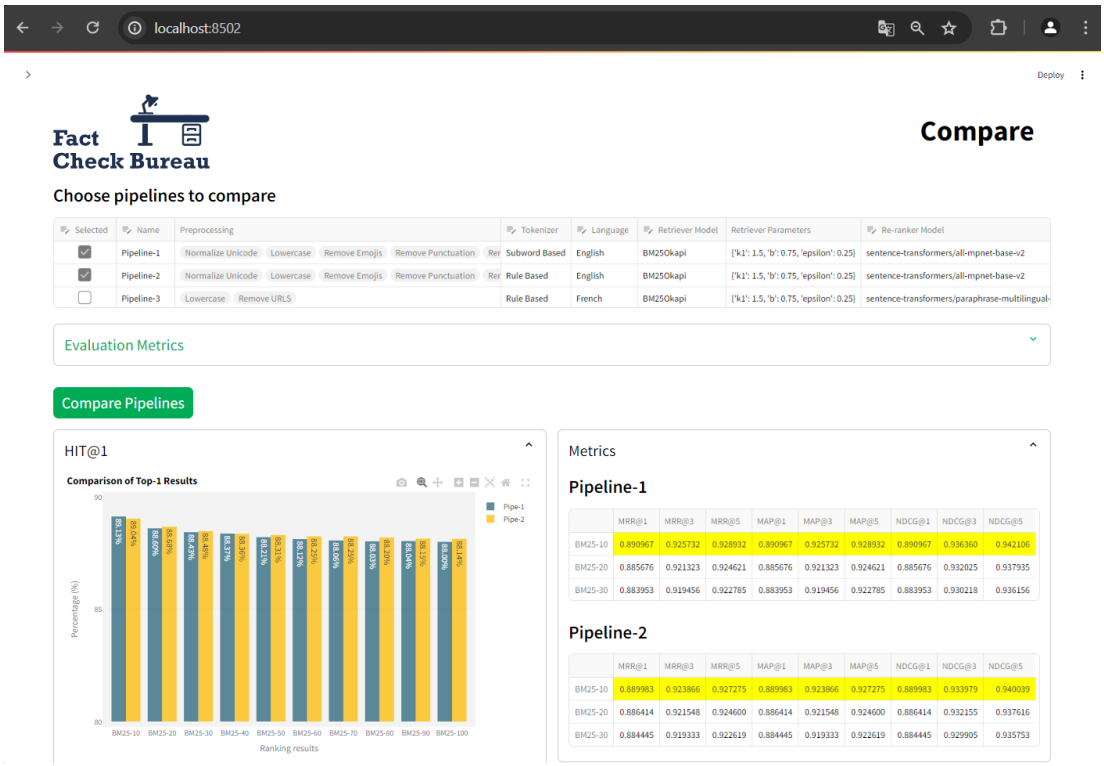

Figure 5.8: FCBureau comparison pag e.

5.6 Technologies Used

The FactCheckBureau platform is built using a variety of open-source technologies, each chosen for its specific capabilities and suitability to the tasks involved in claim verification and fact-checking. These technologies include:

- **Streamlit:** This open-source app framework enables the creation of interactive web applications with minimal code, facilitating rapid development and deployment of the FactCheckBureau platform.
- **Sentence Transformers:** This library provides easy-to-use implementations of state-of-the-art sentence embedding models, such as SBERT, which are crucial for calculating semantic similarity between claims and fact-checking articles.
- **Hugging Face Transformers:** This library allows for seamless integration with the Hugging Face Model Hub, enabling access to a vast collection of pre-trained language models, including those used for sentence embedding, tokenization, and other NLP tasks.
- **spaCy:** This industrial-strength NLP library provides essential tools for text preprocessing, such as tokenization, lemmatization, and part-of-speech tagging, which are vital for preparing text data for analysis.
- **rank-bm25:** This library provides an efficient implementation of the BM25 ranking algorithm, a core component of the initial retrieval stage in the FactCheckBureau platform.
- **dateparser:** This library is used for normalizing dates within text, ensuring consistent and accurate handling of temporal information.
- **demoji:** This library helps to remove emojis from text, facilitating text standardization and preventing potential issues with model compatibility.
- **duckdb:** This in-process SQL OLAP database management system provides a robust and efficient way to perform SQL-based filtering and querying on the dataset within the platform.
- **easyocr:** This library provides optical character recognition (OCR) capabilities, enabling the extraction of text from images within tweets, which can be crucial for capturing additional relevant information.
- **langdetect, lingua, fasttext:** These libraries are used for language detection and identification, ensuring that appropriate language-specific models and processing techniques are applied to the claims and tweets.
- **num2words:** This library converts numerical values into their word representations, facilitating text normalization and potential improvement in semantic understanding.
- **unicodedata:** This Python module provides tools for handling and normalizing Unicode characters, ensuring consistent text representation and preventing encoding-related issues.
- **plotly:** This graphing library enables the creation of interactive and visually appealing visualizations, enhancing data exploration and analysis within the platform.

- **ranx:** This library provides a comprehensive suite of evaluation metrics for ranking tasks, allowing for robust assessment and comparison of different retrieval pipelines.

By leveraging these diverse technologies, the FactCheckBureau platform offers a comprehensive and user-friendly solution for researchers and journalists to explore, develop, and deploy effective fact-checking systems.

5.7 Conclusion

The FactCheckBureau platform represents a significant advancement in facilitating efficient and effective fact-checking processes. By providing a user-friendly interface, flexible pipeline configurations, and robust evaluation tools, FactCheckBureau empowers researchers and journalists to explore, develop, and deploy accurate claim-fact check matching systems. The platform's adaptability to various data formats, including text claims, social media posts, and images, enhances its versatility and applicability in the dynamic landscape of online information.

The development of FactCheckBureau has also highlighted important considerations for building robust fact-checking systems:

- **Language-Specific Considerations:** The performance variations observed between English and French datasets underscore the importance of language-specific models and fine-tuning for optimal results. Choosing appropriate language models and adapting techniques to the nuances of different languages are essential for building accurate and reliable fact-checking systems.
- **The Need for Data Diversity:** The limitations encountered with smaller datasets, particularly for French claims, emphasize the need for diverse and representative data for training and evaluating fact-checking models. Building robust systems that can generalize well to various types of claims and languages requires comprehensive and balanced datasets.
- **The Role of Human Expertise:** While FactCheckBureau provides powerful tools for automated claim-fact check matching, it's important to recognize that human expertise remains crucial in the fact-checking process. The platform is designed to assist and enhance human efforts, not replace them. Critical thinking, contextual understanding, and source verification are still essential for accurate and responsible fact-checking.

In conclusion, FactCheckBureau offers a valuable platform for advancing research and practice in automated fact-checking. Its development and the insights gained from its evaluation contribute to a better understanding of the challenges and opportunities in this domain, paving the way for more effective and reliable tools to combat misinformation and promote accurate information online.

Conclusion and Future Directions

Summary of Contributions

In an era where misinformation spreads rapidly and impacts society on a global scale, this research has tackled the challenge of developing automated tools to facilitate fact-checking processes. The work presented here provides significant advancements in claim-fact matching systems, addressing critical issues in misinformation detection and verification.

Key contributions include:

- **Development of Automated Fact-Checking Systems:** The implementation of an end-to-end system for matching claims and tweets to corresponding fact-checking articles.
- **Innovative Sentence-Level Re-Ranking Approach:** A novel methodology for sentence-level similarity computation using SBERT, which effectively addresses challenges posed by long articles, resulting in improved accuracy for tweet-article matching.
- **Creation of the FactCheckBureau Platform:** A versatile, user-friendly web application designed to empower researchers and journalists by providing tools for data exploration, pipeline customization, and performance evaluation.
- **Comprehensive Experimental Analysis:** Rigorous evaluation of retrieval methods (e.g., BM25 and SBERT) across different data types (claims vs. tweets), languages (English vs. French), and modalities (text, OCR, image captions).

These contributions collectively advance the state of automated fact-checking and provide a foundation for future research in this critical domain.

Key Findings

- **Effectiveness of Sentence-Level Re-Ranking:** The introduction of sentence-level SBERT re-ranking significantly enhanced matching accuracy, particularly for tweet-article matching. This approach demonstrates the importance of focusing on granular semantic similarities to refine retrieval results.

- **Strength of BM25 for Claims:** BM25, with its lexical matching capabilities, remains a strong and efficient baseline for claim-article matching, particularly for English datasets where claims are often directly stated in the text.
- **Challenges with Multimodal Content:** Incorporating OCR and image captioning added complexity and noise, particularly for multilingual content. This underscores the need for more robust tools to process multimodal data in diverse linguistic and cultural contexts.
- **Language-Specific Observations:** Models like CamemBERT outperformed multilingual models for French claims and tweets, emphasizing the importance of language-specific training for optimal performance.

Broader Implications

This research contributes to several fields beyond fact-checking:

- **Information Retrieval (IR):** Insights into improving ranking techniques for relevance and scalability, particularly for challenging datasets such as tweets and long articles.
- **Natural Language Processing (NLP):** Advancements in multilingual and sentence-level semantic similarity modeling provide valuable knowledge for broader NLP applications.
- **Artificial Intelligence (AI):** The research highlights the role of AI in tackling real-world problems like misinformation, while emphasizing the importance of ethical and explainable AI.

Research Areas and Future Directions

While significant progress has been made, this research opens avenues for further exploration:

- **Multilingual and Low-Resource Language Support:** Develop tailored models for low-resource languages and dialects. Enhance cross-lingual embeddings to improve matching across diverse linguistic contexts.
- **Advancements in Multimodal Fact-Checking:** Improve OCR and image captioning tools to handle complex multimodal data effectively. Extend capabilities to verify videos and audio content, which are becoming prominent in misinformation campaigns.
- **Temporal and Contextual Awareness:** Incorporate temporal reasoning into models to verify claims based on their time-specific relevance. Develop systems capable of understanding nuanced contexts, such as sarcasm or cultural references.

- **Explainable and Transparent AI:** Enhance the interpretability of automated fact-checking systems to ensure users understand and trust the results. Explore models that provide clear reasoning for their matching decisions.
- **Real-Time Misinformation Detection:** Integrate live monitoring systems for social media platforms to enable real-time detection and flagging of misinformation. Ensure these systems can scale to handle high-volume data streams.
- **Collaborative Human-AI Systems:** Design frameworks where humans and AI collaborate seamlessly, with AI automating repetitive tasks and humans handling complex, context-dependent decisions.
- **Ethics and Bias Mitigation:** Address biases in datasets and models to ensure fairness across languages, cultures, and demographics. Explore ethical implications of automated fact-checking, particularly in politically or socially sensitive contexts.

Final Remarks

This research represents a meaningful step toward building automated systems that assist in combating misinformation and fostering a more informed society. The insights gained underscore the potential of combining advanced NLP techniques, retrieval models, and user-friendly platforms to address the growing challenges of misinformation in the digital age.

However, automated systems are not standalone solutions. They are most effective when integrated with human expertise, enabling fact-checkers to focus on nuanced and complex investigations. By empowering individuals and organizations with efficient and accurate tools, this work aims to contribute to a future where misinformation is mitigated, and truthful information prevails.

Looking ahead, the ongoing evolution of AI and NLP provides endless opportunities to refine and expand the capabilities of fact-checking systems. By addressing current limitations and exploring new research directions, we can ensure that these tools remain relevant and impactful in an ever-changing digital landscape. Together, technology and human effort can create a more resilient and informed society, capable of navigating the complexities of the information age with confidence and clarity.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer". In: *arXiv preprint arXiv:2004.05150* (2020). Accessed: 2024-10-02.

[2] Shane Connelly. *L'algorithme BM25 en pratique - 2e partie : l'algorithme BM25 et ses variables*. Accessed: 2024-06-26. Apr. 2018.

[3] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 4171–4186.

[4] Devopedia. *Information Retrieval*. Version 15. Accessed: 2024-06-26. Feb. 15, 2022. URL: https://devopedia.org/information-retrieval.

[5] FullFact. *The challenges of online fact checking*. https://fullfact.org/media/uploads/coof-2020.pdf. 2020.

[6] Anwar ul Haque, Sayeed Ghani, and Muhammad Saeed. "The Storyteller: Computer Vision Driven Context and Content Generation System". In: *Research Square* (2021). Accessed: 2024-06-26.

[7] Rani Horev. "BERT Explained: State of the art language model for NLP". In: *Towards Data Science* (Nov. 2018). Accessed: 2024-06-26.

[8] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[9] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Third. Draft version, comments and typos welcome. Draft of August 20, 2024. Stanford University and University of Colorado at Boulder, 2024.

[10] Michael King. *The Technical SEO Renaissance: The Whys and Hows of SEO's Forgotten Role in the Mechanics of the Web*. Accessed: 2024-06-26. Oct. 2016. URL: https://moz.com/blog/the-technical-seo-renaissance#let%E2%80%99s-make-seo-great-again.

[11] Zhenzhong Lan et al. "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations". In: *International Conference on Learning Representations (ICLR)*. 2020.

[12] Farhad Mortezapour Shiri et al. "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU". In: *arXiv preprint arXiv:2305.17473* (2023). Accessed: 2024-10-02.

[13] Preslav Nakov et al. "Automated Fact-Checking for Assisting Human Fact-Checkers". In: *IJCAI*. 2021.

[14] Matúš Pikuliak et al. "Multilingual Previously Fact-Checked Claim Retrieval". In: *arXiv preprint arXiv:2305.07991* (2023). Accessed: 2024-10-02.

[15] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *arXiv preprint arXiv:1908.10084* (2019). Accessed: 2024-06-26.

[16] Eshant Sah. *Deep Learning — In simple words…* Accessed: 2024-06-26. Dec. 2018.

[17] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2020).

[18] Shaden Shaar et al. "That is a Known Lie: Detecting Previously Fact-Checked Claims". In: *arXiv preprint arXiv:2005.06058* (2020). Accessed: 2024-10-02.

[19] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[20] Nguyen Vo and Kyumin Lee. "Where Are the Facts? Searching for Fact-checked Information to Alleviate the Spread of Fake News". In: *arXiv preprint arXiv:2010.03159* (2020). Accessed: 2024-10-02.

[21] Soroush Vosoughi, Deb Roy, and Sinan Aral. "The spread of true and false news online". In: *Science* 359 (2018).

[22] Benjamin Wang. *Ranking Evaluation Metrics for Recommender Systems. Towards Data Science*. Accessed: 2024-10-02. 2021. URL: https://towardsdatascience.com/ranking-evaluation-metrics-for-recommender-systems-263d0a66ef54.

[23] Gokul Yenduri et al. "GPT (Generative Pre-trained Transformer) – A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions". In: *arXiv preprint arXiv:2305.10435* (2023).

Appendix A

External Data Sources and APIs

A.1 Google Fact Check Explorer API

- **Purpose:** The Google Fact Check Explorer API is a programmatic interface that allows developers to access and utilize the data available in the Google Fact Check Explorer. This API enables the retrieval of fact-checking information related to claims or topics, providing a valuable resource for combating misinformation and promoting transparency.
- **Functionality:** The Google Fact Check Explorer API primarily provides the following functionalities:
  - **Search for Fact Checks:** Retrieve fact checks related to specific claims or keywords.
  - **Filter Fact Checks:** Apply filters based on criteria such as publisher, date range, language, or claim review rating.
  - **Access Fact Check Details:** Retrieve detailed information about each fact check, including the claim, the publisher's assessment, the evidence provided, and the date of publication.
- **Use Cases:** The Google Fact Check Explorer API has a wide range of potential applications, including:
  - **Fact-Checking Tools & Platforms:** Integrate fact-checking information into applications or platforms to help users verify the accuracy of claims.
  - **Misinformation Research:** Collect and analyze large-scale fact-checking data to study the spread of misinformation and develop countermeasures.
  - **News & Media Analysis:** Track the fact-checking landscape and analyze the performance of different publishers.
  - **Educational Purposes:** Use fact-checking data to educate users about critical thinking and media literacy.

- **Access & Authentication:** To use the Google Fact Check Explorer API, developers need to obtain an API key from Google. The API might have usage limits or restrictions, so it's important to review the documentation and terms of service.

A.2 Twitter API

- **Purpose:** The Twitter API (Application Programming Interface) is a set of tools and protocols that allow developers to interact with Twitter data and functionality programmatically. It essentially acts as a bridge between external applications and the vast amount of information available on Twitter.
- **Functionality:** The Twitter API enables developers to:
  - Access Tweet Data: Search for and retrieve tweets based on various criteria such as keywords, hashtags, user mentions, or specific timeframes.
  - Manage Accounts: Perform actions on behalf of Twitter users, such as posting tweets, following other users, or liking tweets, with proper authorization.
  - Gather User Information: Retrieve information about Twitter users, including their profiles, followers, and tweets.
  - Analyze Trends: Identify trending topics and hashtags in real-time or over specific periods.
  - Stream Live Data: Receive a continuous stream of tweets matching specific criteria, enabling real-time monitoring and analysis.
- **Use Cases:**

  The Twitter API has a wide range of applications, including:
  - Social Media Monitoring & Analytics: Track brand mentions, analyze sentiment, and gain insights into audience behavior.
  - Research & Data Analysis: Collect and analyze large volumes of tweets for various research purposes.
  - Customer Service & Engagement: Monitor and respond to customer inquiries and feedback on Twitter.
  - Content Creation & Automation: Schedule tweets, create bots, and automate various social media tasks.
  - Access & Authentication: To use the Twitter API, developers need to create a developer account and obtain API keys and access tokens for authentication. Twitter enforces rate limits to prevent abuse and ensure fair usage of the API.

Appendix B

FactCheckBureau Application - Installation and Usage Guide

This annex provides a comprehensive guide to installing and using the FactCheckBureau application, an end-to-end solution that enables researchers to easily and interactively design and evaluate FC retrieval pipelines.

The FactCheckBureau application is available on GitLab at the following link: https://gitlab.inria.fr/cedar/factcheckbureau.git

1. Installation

System Requirements

Operating System: Linux
Python Version: 3.10
CUDA Toolkit: Required for GPU support. Visit https://developer.nvidia.com/cuda-downloads to download and install.

Git LFS

Ensure Git Large File Storage (LFS) is installed. See the instructions at https://git-lfs.com/.

Then, set it up:

```
git lfs install
```

Cloning and Installing

Clone the repository:

```
git clone https://gitlab.inria.fr/cedar/factcheckbureau.git
```

Navigate to the project directory:

```
cd factcheckbureau
```

Run the installation script (ensure it has execute permissions):

```
chmod +x scripts/install.sh ./scripts/install.sh
```

To specify a Python interpreter:

```
./scripts/install.sh –python=path_to_python_interpreter
```

Troubleshooting

If you encounter the error "os error: No space left on disk," specify temporary directories for pip:

```
./scripts/install.sh –tmpdir=/path/to/tmp –cache-dir=/path/to/cache
```

2. Usage

Run the app:

```
chmod +x scripts/run.sh ./scripts/run.sh
```

For a guide on using the app, see this video: https://drive.google.com/file/d/1_LRHwkHeWbmdGP-wh9affPQjN_9bE2uG/view?usp=sharing.

Data Requirements

Add your fact-check articles to '/data/articles.csv' with these fields:

- **id:** Unique identifier for each article
- **url:** The article's URL
- **title:** The title of the article
- **body:** The body of the article (initially empty, to be filled by the user)

Using Your Own Corpus

To use your own corpus, upload it on the ”Inspect” or ”Compare” pages. The corpus directory should have this format:

- **queries.csv:** Contains the queries.
  - First column: Query IDs
  - Second column: Query strings
- **documents.csv:** Contains the documents.
  - First column: Document IDs
  - Second column: Document strings
- **relations.csv:** Specifies query-document pairs.
  - First column: Query IDs
  - Second column: Document IDs
- **images/ (Optional):** Required if OCR is in the pipeline. Each image file should be named with the corresponding query ID. For supported image formats, see the PIL module documentation (https://pillow.readthedocs.io/en/stable/) or run:

  ```
  python3 -m PIL
  ```

  or the function:

  ```
  PIL.features.pilinfo()
  ```

Example:

queries.csv

```
id,query
1,“query$_a$”
2,“query$_b$”
3,“query$_c$”
```

documents.csv

id,document
1,“doc$_a$”
2,“doc$_b$”
3,“doc$_c$”
4,“doc$_d$”

relations.csv

query$_i d$, *document*$_i d$
1,1
2,2
3,3
3,4

3. Closing the App

Hit CTRL + C to close the app.