University of Milano-Bicocca

Department of Computer Science, Systems and Communication

Master's Degree in Data Science

INTEGRATING KNOWLEDGE WITH NATURAL LANGUAGE PROCESSING FOR ENHANCED FACT-CHECKING

Supervisor: Marco Viviani

Co-supervisor: Rafael Penaloza Nyssen

Master's Thesis by:

Simone Farallo

889719

Academic Year 2022-2023

Contents

Abstract

In the digital age, where the proliferation of information intersects with the rise of sophisticated digital platforms, the challenge of distinguishing between fact and fiction has become a pivotal concern. This thesis addresses the critical issue of misinformation and disinformation through the development and evaluation of advanced automated fact-checking systems [Lazarsk and Mahmood Al-Khassaweneh, 2021]. Leveraging state-of-the-art Natural Language Processing techniques [Schopf et al., 2023], Large Language Models [Radford et al., 2019], and Graph-based analytics [Hogan et al., 2023], this research introduces novel methodologies aimed at enhancing the accuracy, reliability, and explainability of fact-checking mechanisms. Central to this work is the comparative analysis of embedding-based and graph-based approaches to automated fact-checking, exploring their respective strengths and limitations in addressing the complexity of verifying digital content. Navigating the intricacies of evaluating digital content, the study uncovers that methods based on embeddings shine in grasping subtle semantic layers within various statements. Meanwhile, graph-centric strategies provide a clear, structured system for assessing claims with reference to well-defined knowledge collections. By synthesizing these methodologies, the thesis advances a hybrid fact-checking model that combines the depth of semantic analysis with the precision of graph-based verification, aiming to improve the overall efficacy of fact-checking systems. Furthermore, this thesis places a strong emphasis on the principle of explainability within automated fact-checking processes. Recognizing the importance of transparency and accountability, it proposes mechanisms that not only determine the veracity of claims but also elucidate the rationale behind verification decisions. This dual focus on performance and explainability seeks to foster trust in automated fact-checking systems among users and stakeholders, contributing to the broader goal of combating misinformation in the digital landscape.

Through empirical evaluation across a range of original and generated claims, the study demonstrates the adaptability and precision of the proposed hybrid model, showcasing its capability to navigate the intricate chal-

lenges posed by the digital information ecosystem. The results highlight not only the technological advancements achieved but also the critical need for ongoing innovation in the fight against misinformation. This thesis contributes to the field of automated fact-checking by developing a framework that combines computational techniques with a focus on AI practices. It establishes a basis for further research to enhance the precision, efficiency, and clarity of automated fact-checking systems, signifying a notable development in pursuing accuracy in the digital era.

Chapter 1

Introduction

1.1 Context

In today's digital age, the sheer volume of information flowing through countless channels has rendered the task of discerning truth from falsehood more daunting than ever before. The rise of advanced technologies, coupled with the widespread reach of social media, has accelerated the spread of information to a global scale, complicating efforts to sift through the vast expanse of the digital information landscape. The evolution of digital platforms, coupled with the explosive growth of social media, has democratized content creation and distribution, enabling information to traverse borders with unprecedented speed. However, this democratization has also paved the way for a proliferation of misinformation and disinformation, casting shadows of doubt across the digital landscape and threatening the integrity of public discourse. The field of automated fact-checking has thus gained significant momentum, driven by the urgent need to counteract the spread of falsehoods and ensure the integrity of information shared online. These systems, which employ a variety of computational techniques to assess the credibility of claims, represent a pivotal frontier in the battle against misinformation.

However, the task of automated fact-checking is fraught with challenges. The nuances of language, the context-dependent nature of truth, and the dynamic evolution of knowledge all contribute to the complexity of verifying claims automatically. Furthermore, the rise of generative AI models capable of producing highly realistic and nuanced text has introduced new dimensions to the challenge, blurring the lines between authentic content and fabricated narratives.

1.2 Contribution

This thesis represents a comprehensive exploration of the current landscape of automated fact-checking, aiming to address some of the most pressing challenges in the field. By integrating state-of-the-art NLP techniques, leveraging Large Language Models (LLMs), and employing graph-based approaches, this work seeks to push the boundaries of what is achievable in automated verification of claims. The contributions of this research are multifaceted, encompassing the development of novel methodologies, the refinement of existing techniques, and the empirical evaluation of system performance across a diverse array of claims.

A significant focus of this research lies in enhancing the explainability and reliability of fact-checking systems. Recognizing the critical importance of transparency and accountability in automated decision-making, this thesis endeavors to develop mechanisms that not only assess the veracity of claims with high precision but also provide clear and interpretable rationales for their verdicts. This dual emphasis on accuracy and explainability serves to bolster the credibility of automated fact-checking systems, making them more trustworthy and user-friendly.

The contributions of this thesis are manifold, reflecting a deep engagement with both the technological and ethical dimensions of fact-checking in the digital era. It introduces novel methodologies that extend beyond the state of the art, refining and applying sophisticated computational strategies to confront the specters of misinformation and disinformation head-on. This research is distinguished not only by its technological innovations but also by its commitment to the principle of explainability. In elevating the importance of understanding the 'why' behind each fact-checking decision, this work acknowledges the critical role of trust and interpretability in the interaction between automated systems and human users. Moreover, this thesis ventures into the comparative analysis of embedding-based versus graph-based approaches to fact-checking, shedding light on the unique advantages and challenges each methodology presents. Through empirical evaluation and rigorous analysis, it reveals how the fusion of these methodologies can yield a more robust, adaptable, and effective fact-checking system.

1.3 Document Structure

The structure of this document is meticulously designed to guide the reader through the intricacies of automated fact-checking research. Following this introduction, the thesis unfolds as follows:

- **Chapter 2: Background and Related Work**, delves into the theoretical foundations and current state of the art in automated fact-checking. This chapter provides a critical review of the literature, outlining the evolution of the field and highlighting key methodologies, technologies, and challenges.
- **Chapter 3: Methodology**, presents the research design and methodologies employed in this study. It details the computational techniques and algorithms developed, the data sources utilized, and the criteria for evaluating system performance.
- **Chapter 4: Results and Discussion**, offers a comprehensive analysis of the findings. It evaluates the performance of the proposed fact-checking system across various metrics, discusses the implications of the results, and provides insights into the effectiveness of different fact-checking approaches.
- **Chapter 5: Conclusion and Future Work**, concludes the thesis by summarizing the key contributions and outcomes of the research. This chapter also outlines potential avenues for future investigation, reflecting on the ongoing journey towards more effective and reliable automated fact-checking systems.

By traversing these chapters, the reader will gain a deep understanding of the challenges and opportunities in the field of automated fact-checking, along with a detailed account of this research's contributions towards advancing the state of the art.

Chapter 2

Background and Related Work

2.1 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science that focuses on developing algorithms and models for the analysis and synthesis of natural language, which is the language spoken or written by humans. The primary aim of NLP is to empower computers to comprehend human language in a manner akin to human understanding [Schopf et al., 2023]. NLP merges computational linguistics and rule-based human language modeling with statistical, machine learning, and deep learning approaches. Together, these technologies allow computers to process human language in the form of textual or vocal data and "understand" its complete meaning, including the author's or speaker's intention and sentiment. NLP stands as one of the most promising and rapidly evolving areas within artificial intelligence and computing, aiming to bridge the gap between human language understanding and its interpretation by machines. At its core, NLP is dedicated to the study and development of algorithms that enable computers to process and "understand" natural language data, aiming to realize human-machine interactions that are as smooth and natural as possible.

From its inception, NLP has gone through various evolutionary phases, starting from simple rule-based approaches to modern machine learning and deep learning techniques, significantly expanding its capabilities and applications. These advancements have enabled significant progress in areas such as personalized virtual assistance, automated processing of large volumes of text and data, and natural interaction with digital devices and services.

Today's NLP is characterized by the use of complex machine learning models, especially deep neural networks, which have proven to be particularly effective in processing natural language due to their ability to learn rich and

layered representations of data. Trained on vast text corpora, these models can capture subtle semantic and contextual nuances, making possible an increasingly human-like understanding and generation of language.

NLP is a fundamental field of study and application for the advancement of computing technologies and artificial intelligence, promising to revolutionize how we interact with machines and manage information in the digital world. With ongoing research and innovation, the future of NLP holds great potential, opening new frontiers in automation, service personalization, and the creation of increasingly intuitive and engaging user experiences.

2.1.1 Evolution of NLP

The origins of NLP can be traced back to the earliest attempts at automating language translation in the 1950s, a period marked by the pioneering work of Alan Turing, whose ideas laid the groundwork for computational linguistics. Turing's introduction of the Turing Test was a significant milestone, challenging the notion of machine intelligence and its capability to understand human language. The subsequent decades saw incremental advancements, with the 1960s introducing the first chatbot, ELIZA [Weizenbaum, 1966], simulating a psychotherapist session, showcasing the potential of machines in processing written language despite its simplicity and pattern-matching limitations.

The evolution of NLP took a significant turn with the advent of machine learning in the 1980s and '90s [Jurafsky and Martin, 2019], where statistical methods began to replace rule-based systems, allowing for more nuanced language processing. This era laid the foundation for the use of probabilistic models and decision trees in language tasks, which was a departure from the rigid structures of the past.

The real transformation in NLP, however, came with the introduction of deep learning techniques in the 2010s. The shift to neural network models, particularly *Recurrent Neural Networks* (RNNs) [Hochreiter and Schmidhuber, 1997], and later, Transformer models like BERT and *Generative Pre-trained Transformer* (GPT), represented a paradigm shift, enabling machines to process and generate human language with remarkable levels of complexity and subtlety. These models leveraged vast amounts of data to learn language patterns, context, and semantics, far surpassing the capabilities of earlier systems.

In recent years, NLP has become increasingly democratized, with tools and platforms making it accessible to a broader audience beyond those with deep technical expertise. The rise of low-code and no-code platforms, alongside advanced language models offered by companies like OpenAI, has made

sophisticated language processing tools available to non-specialists, expanding the field's reach and application. The release of models like GPT-3 has further spotlighted NLP's potential, capturing public imagination and demonstrating the vast possibilities of generative language models. This ongoing evolution underscores the dynamic nature of NLP, as it continues to push the boundaries of how machines understand and interact with human language.

2.1.2 Text Representation

In NLP text representation techniques have evolved remarkably, enabling computers to process human language with increasing sophistication. Early methods like *Bag of Words* (BoW) and *Term Frequency-Inverse Document Frequency* (TF-IDF) provided simple yet effective ways to convert text into numerical data, focusing on word occurrence frequencies without considering the order or context of words [Manning et al., 2008]. The advent of word embeddings marked a significant leap forward. Techniques such as Word2Vec [Mikolov et al., 2013a] and GloVe [Pennington et al., 2014] emerged, capturing not just the presence of words but also the semantic relationships between them by mapping words into a continuous vector space. These embeddings reflected the contextual nuances of language, allowing for much richer interpretations of text [Mikolov et al., 2013b]. These techniques allow for the transformation of words or phrases into vectors of real numbers, thus mapping natural language into a mathematical space that is more manageable and interpretable for computers. This transformation occurs by moving from a high-dimensional space to a reduced one, optimizing the process of language processing and semantic analysis.

Embeddings [Manning and Schütze, 1999] are based on vector semantics, where the similarity between words is reflected in the proximity of their vectors in mathematical space. This is achieved through various representations: from sparse one-hot matrices, which, although intuitive, are excessively large and lack context, to distributed representations that reduce sparsity and better capture semantic relationships between words by using reduced context dimensions. A fundamental approach is the use of co-occurrence matrices, which leverage the distributional hypothesis to represent words based on their context. However, raw frequency tends to overestimate the importance of common words, making weighting schemes like TF-IDF or *Pointwise Mutual Information* (PMI), with its positive variant (PPMI), necessary to mitigate biases towards less frequent events.

Count-based models, such as *Singular Value Decomposition* (SVD) with *Latent Semantic Analysis* (LSA), *Latent Dirichlet Allocation* (LDA), and

GloVe, fit into this framework, aiming to reduce the dimensionality of term-context matrices and to produce dense, informative vectors for each word [Pennington et al., 2014]. GloVe, in particular, manages to capture global co-occurrence information, optimizing the relationship between words through a least squares minimization approach.

Parallel to this, predictive models like Word2Vec [Mikolov et al., 2013a] introduce a different perspective, based on the ability to predict a word from its neighbors, generating dense embeddings that reflect the semantics emerging from the training data. The Skip-Gram and CBOW architectures, supported by training techniques such as Hierarchical Softmax and Negative Sampling, offer powerful tools for language comprehension, though with specific strengths and limitations depending on the application context.

In an attempt to capture the dynamic nature of language, contextual embeddings like ELMO [Peters et al., 2018] and BERT [Devlin et al., 2019] then emerge, surpassing the limitations of previous models by including context sensitivity. These approaches, based on neural language models and innovative architectures such as bidirectional *Long-Short Term Memory* networks (LSTMs) for ELMO and the masking approach for BERT, open new frontiers in text semantic interpretation, enabling a more nuanced and dynamic understanding of meaning. Embeddings have revolutionized the NLP field, offering increasingly sophisticated tools for the semantic analysis of natural language. From simple one-hot representations to complex deep learning-based models, these techniques have become fundamental for applications ranging from NLP to semantic search, demonstrating the crucial importance of data science in understanding and interpreting human language.

2.1.3 NLP Applications

NLP has significantly broadened its impact across multiple fields, transforming the way we interact with technology and interpret vast amounts of data [Young et al., 2018]. In automatic translation, platforms like Google Translate have evolved to provide not only word-for-word translation but also contextual understanding, making conversations, documents, and web pages accessible to a global audience without language barriers. This advancement is particularly crucial in educational resources, business negotiations, and international relations, enabling smoother cross-cultural exchanges. Voice recognition technologies, embodied by virtual assistants such as Siri, Alexa, and Google Assistant, have made significant strides, integrating into our daily routines from setting reminders to controlling smart home devices. This interaction mode has not only made technology more intuitive but has also been a game-changer for users with disabilities, offering them greater in-

dependence. Sentiment analysis has become a pivotal tool in social media analytics, enabling businesses and policymakers to monitor public opinion and emotional trends in real-time. By analyzing customer feedback, reviews, and social media posts, companies can tailor their strategies, products, and services to better meet their audience's needs. This technology has also been instrumental in crisis monitoring and management, where understanding public sentiment is crucial. The text generation capabilities of models like GPT-3 have opened new frontiers in content creation, from automating routine correspondence and generating reports to creating novel content and even poetry. These models can produce text that is increasingly indistinguishable from that written by humans, offering potential efficiencies in content production, creative writing, and even coding.

Furthermore, NLP applications in general fields have demonstrated profound impacts, from parsing clinical notes to assist in diagnosis and treatment plans, to analyzing patient conversations to monitor mental health status. In the legal field, NLP aids in document analysis, helping to sift through and summarize vast legal texts, contracts, and case law, streamlining the review process. Each of these applications not only showcases the versatility of NLP in automating and enhancing textual tasks but also highlights its role in making information more accessible, fostering innovation, and driving efficiency across industries. As NLP technologies continue to evolve, their integration into various domains is expected to deepen, further revolutionizing our interaction with digital systems and enhancing our ability to derive meaningful insights from language data.

2.1.4 Challenges and Ethical Con siderations

As NLP technology progresses, it encounters critical challenges and ethical dilemmas that necessitate careful consideration and action. Privacy and data security stand out as paramount concerns. NLP systems, by their nature, require access to extensive personal and sensitive information to learn language patterns and make predictions. This raises significant questions about the safeguarding of such data against misuse and breaches [Hovy and Spruit, 2016].

Furthermore, the issue of bias in NLP is a matter of significant concern. Training data for models, if not carefully curated, can reflect and perpetuate existing societal biases. These biases, when embedded in NLP applications, can lead to unfair or prejudiced outcomes, affecting everything from job hiring processes to legal sentencing recommendations. Another ethical consideration is the potential for NLP to be used in spreading misinformation or conducting surveillance without consent. The ease with which NLP can

generate convincing text makes it a powerful tool for creating fake news or impersonating individuals online, posing a threat to public trust and individual privacy.

To navigate these challenges, a multifaceted approach is required. Implementing stringent data protection measures and ensuring transparency in how data is collected, used, and stored is critical for maintaining user trust. Actively identifying and mitigating bias in training datasets is essential for building fair and equitable NLP systems. This may involve diversifying data sources, applying fairness algorithms, and continuous monitoring for biased outcomes.

Moreover, establishing ethical guidelines for the development and application of NLP technologies is imperative. These guidelines should address concerns about consent, transparency, accountability, and the societal impact of NLP applications. Engaging with ethicists, policymakers, and the wider community can help ensure that NLP technology develops in a way that respects individual rights and benefits society as a whole.

2.2 Knowledge Bases

Knowledge Bases (KBs) constitute a crucial pillar in the field of Data Science, serving as fundamental tools for organizing, managing, and efficiently utilizing information. Thanks to their ability to facilitate access and processing of large quantities of structured and semi-structured data, KBs stand out for their support in data-driven decision-making processes and analytical depth; KBs go beyond mere data storage, providing a framework for interpretation and analysis [Bollacker et al., 2008].

Their structure is a fundamental aspect, as it allows representing information in a comprehensible format for both humans and machines. This is vital for supporting machine learning techniques and artificial intelligence, enabling the creation of data models that reflect real-world complexities. Furthermore, KBs are essential for powering recommendation algorithms and AI-based assistance systems, offering a deep knowledge base.

The use of KBs to enrich existing datasets significantly enhances the quality of training sets for machine learning models, thereby increasing the accuracy of predictive analyses. This approach also allows for the discovery of hidden patterns and correlations.Kbs play a pivotal role as organized collections of information, facilitating the storage, enrichment, and querying of both structured and unstructured data. These resources enable users and systems to efficiently access and manage information, incorporating database techniques, artificial intelligence, and semantic web technologies to make data

easily retrievable and usable. Knowledge bases are applied across a variety of contexts, providing decision support through quick access to relevant and consolidated information, enhancing customer service efficiency via management of FAQs and support guides, and serving as the backbone for artificial intelligence and expert systems by providing the necessary context and information for reasoning and machine learning. Additionally, they support research by aggregating vast datasets and scientific literature into an organized and queryable format.

The structure of a knowledge base can vary widely depending on its specific use. There are structured approaches, where data is rigidly organized with defined schemas, such as in relational databases that strictly type and relate data to each other. Semi-structured knowledge bases combine elements of structured and unstructured data, like graph databases and content management systems, offering greater flexibility in data organization. Lastly, unstructured knowledge bases gather data not organized in any specific format, necessitating the use of NLP techniques for information extraction.

Data acquisition involves collecting information from diverse sources and enriching it through annotation or integration. Organizing and storing such data requires defining schemas, ontologies, or data models to efficiently and scalably organize the information. Moreover, it is crucial to implement flexible and powerful query interfaces, often utilizing specific query languages to facilitate data interrogation and access. Security and privacy are another critical aspect, ensuring that data access is secure and that sensitive information is adequately protected.

In their use, knowledge bases allow for querying and analysis by users or systems to retrieve specific information or conduct complex analyses. They are often integrated with other technologies, such as artificial intelligence systems, recommendation engines, or analytical dashboards, to enhance their functionalities. These knowledge bases also facilitate the sharing and dissemination of knowledge within organizations or research communities, promoting collaboration and innovation.

Knowledge bases constitute an indispensable component within the data science landscape, offering advanced tools for efficient management, organization, and utilization of information. Their implementation and use require a deep understanding of both technical and application aspects, presenting a rich field of study and work full of challenges and opportunities. The ability to develop, maintain, and effectively exploit knowledge bases can significantly impact the success of numerous data science initiatives, from scientific research to business innovation, underscoring their fundamental importance in an increasingly data-driven era [Bollacker et al., 2008].

2.2.1 Knowledge Bases and NLP

In NLP knowledge bases play an essential role by serving as repositories of structured information that can be leveraged to enhance the understanding, generation, and manipulation of natural language by computers. These knowledge bases are pivotal for a myriad of applications, ranging from semantic analysis and machine translation to question answering systems and chatbots, providing a rich source of semantic information that helps bridge the gap between the complexity of human language and the computational models used to process it [Nickel and Poon, 2016].

Knowledge bases in NLP contain a vast array of information about the world, including facts about entities, their attributes, and the relationships between them. This information is crucial for tasks such as named entity recognition, disambiguation, and relation extraction, enabling systems to understand the context and meaning behind the text. For instance, a knowledge base might contain detailed information about geographical locations, historical figures, or technical concepts, which can be used to inform NLP models and improve their ability to interpret queries or documents accurately.

The construction of knowledge bases in NLP often involves the extraction of structured information from unstructured text, a process known as information extraction. This can include the identification of entities and their properties from news articles, books, or websites, and the aggregation of this information into a coherent structure that can be easily queried. Semantic web technologies, such as Resource Description Framework (RDF) [World Wide Web Consortium (W3C), 2014] and Web Ontology Language (OWL) [World Wide Web Consortium (W3C), 2012], are commonly used to represent the data within knowledge bases, allowing for sophisticated querying and reasoning capabilities that can support complex NLP tasks.

One of the most significant challenges in utilizing knowledge bases for NLP is ensuring their comprehensiveness and accuracy. The dynamic nature of language and the world it describes means that knowledge bases must be continually updated and expanded to remain relevant. This requires not only the automated extraction of information from new sources but also the curation and validation of this information to prevent the propagation of errors or biases within NLP applications.

Knowledge bases are not only used to support the understanding of text but also its generation. In tasks such as automated storytelling, content generation, and conversational agents, knowledge bases provide the factual and contextual grounding needed for generating coherent, relevant, and engaging output. By tapping into a structured repository of information, these systems can produce responses that are not only grammatically correct but also

rich in content and meaning.

Furthermore, the integration of knowledge bases with machine learning models in NLP offers a promising avenue for improving the performance of these models. By incorporating structured world knowledge into the training process, models can develop a deeper understanding of the nuances of language and improve their ability to make inferences, handle ambiguity, and generate more accurate predictions.

Knowledge bases providing the essential semantic foundation required for processing and understanding natural language at a level that approaches human understanding. They enable a wide range of applications, from enhancing the accuracy of translation services to powering sophisticated conversational agents, and their importance will only continue to grow as the field of NLP advances. The development and refinement of knowledge bases, coupled with their integration into NLP models, represent a vibrant area of research and development that holds great promise for the future of human-computer interaction.

2.3 Knowledge Graphs

A *Knowledge Graph* is a structured representation of knowledge, information, and relationships among real-world concepts [Hogan et al., 2023]. It is a directed graph where nodes represent entities, concepts, or facts, while edges (often called "triples") indicate semantic relationships between these nodes.

Nodes represent real-world entities such as people, places, events, objects, concepts, or anything that can be described or identified. Each node has a unique identifier. Edges are direct relationships between nodes, connecting two nodes together; each edge is annotated with a predicate specifying the nature of the relationship between the two nodes. For example, a triple might link a "Person" node to a "Place" node with the edge "Born in," indicating that the person was born in that place. In addition to relationships between nodes, a knowledge graph can include properties or attributes associated with the nodes themselves. These provide additional information about the concepts represented by the nodes.

The knowledge graph is constructed with the goal of meaningfully representing knowledge; the relationships between nodes carry semantic meaning, enabling computers and applications to understand the information in the graph more deeply.

2.3.1 Ontologies

A graph represents data, whereas an *ontology* represents knowledge; a graph often incorporates an ontology or a semantic structure that formally defines the types of nodes, relationships, and semantic rules governing the graph. Ontologies provide a semantic context that helps interpret and organize the information in the graph [Hogan et al., 2023].

An ontology is characterized by a semantic structuring that provides a framework for defining key concepts in the domain, their properties, and the relationships between them. Ontologies define a hierarchy of classes or concepts representing different entities or categories of objects in the domain. For instance, in a medical knowledge graph, there could be classes for "Diseases," "Symptoms," "Medications," and so on. Each class can have specific attributes and properties. Furthermore, ontologies define semantic relationships between concepts. These relationships specify how different classes and instances are related to each other. For example, in the medical domain, there might be a "Treats" relationship between the "Medication" class and the "Disease" class to indicate which drugs are used to treat specific diseases.

Ontologies can also include logical assertions or rules specifying how concepts and relationships should behave within the knowledge graph. These axioms can be used to infer new information or to check the consistency of the data.Knowledge graphs facilitate reasoning and inference capabilities, allowing the tracing of connection paths within the graph to identify links and additional information that can contribute to verifying or refuting a claim.

2.3.2 Types of Graph

There are different types of graphs that can be used depending on the objectives and the field of application [Hogan et al., 2023]:

- *Tabular Structure Graphs* (Relational Databases). This representation uses a tabular structure to manage event data. However, this representation has encountered problems due to the diversified nature of event data, such as events with multiple names, locations, or types. The relational model has several limitations because data requirements are not always known in advance, and any schema changes require costly remodeling and data reloading.

- *Labeled Directed Edge Graphs*, also known as multi-relational graphs, consist of nodes and directed edges with labels connecting them. This model offers flexibility in representing event data, names, types, start and end times. Additionally, adding information involves inserting new

nodes and edges, simplifying the representation of incomplete information.

- *Heterogeneous Graphs* assign "type" (referring to a property or attribute) to both nodes and edges, where each node and edge is associated with a "type," allowing node subdivisions. This model might limit flexibility in representing complex data structures.

- *Property Graphs* offer flexibility in modeling complex relationships by associating property-value pairs and labels with both nodes and edges. Each node and edge can have properties, and labels capture the type of relationship. A practical example is Neo4j.

- *Graph Sets* consist of different named graphs and a default graph; each named graph is identified by a unique ID and contains a graph. The default graph is used when a specific graph ID is not specified. This approach is useful for managing multiple graphs from different sources, updating data, and distinguishing between reliable and unreliable sources. Nodes and edges can be repeated across graphs, allowing data integration during graph merging.

These graphs can be modeled based on various factors such as the study's objectives. Graph modeling involves defining a set of nodes and edges that represent the elements and relationships of the domain of interest. Graphs can be modeled in various ways, depending on the specific problem's requirements. There are several modeling types that can be used to represent a knowledge graph. The most commonly used modeling types are [Seo et al., 2022]:

- *Graph-Based Modeling*, the most common method for representing a knowledge graph, where entities are represented as nodes, and the relationships between them are represented as edges; some of the well-known graph formats are *Resource Description Framework* (RDF) and *Web Ontology Language* (OWL).

- *Triple-Based Modeling*, where information in the knowledge graph is represented as RDF triples; this is a highly flexible way to represent complex relationships between entities.

- *Ontological Modeling*, where an ontology is used to define the classes, properties, and relationships between entities in the knowledge graph. The ontology provides a formal structure to organize and categorize entities and relationships, making data management and processing easier.

- *Vector-based Modeling*, where entities and relationships are represented as vectors in a multidimensional vector space. These vectors can be trained using machine learning techniques like Word2Vec or GloVe to capture semantic relationships between entities.
- *Neural Graph-based Modeling*, a more recent approach that uses neural networks to learn representations of graph nodes and relationships; algorithms like *Graph Convolutional Networks* (GCNs) and *Graph Neural Networks* (GNNs) are used to incorporate graph information into entity representations.
- *Rule-based Modeling*, where explicit rules are used to infer new information within the knowledge graph, rules can be written manually or learned automatically from existing data.
- *Logic-based Modeling*, where formal logic is used to represent and infer relationships in the knowledge graph, for example, first-order logic can be used to formulate complex queries on graph data.

Each modeling type has its advantages and limitations, and the choice often depends on the specific application requirements. It is also possible to combine different approaches to achieve better results in knowledge graph representation and processing.

2.4 Large Language Models

Large Language Models (LLMs) represent the cutting edge of artificial intelligence in the field of NLP, embodying a significant leap forward in machines' ability to understand, generate, and interact with human language. These models are built upon the foundation of deep learning, particularly utilizing a type of neural network architecture known as the *Transformer* [Lewis et al., 2020], which has revolutionized the way computers process language [Radford et al., 2019].

At their core, LLMs are trained on extensive corpora of text data, ranging from books, articles, and websites to specialized texts in various fields. This training involves the analysis of billions of words and phrases to learn patterns of language, including grammar, syntax, semantics, and even some aspects of common sense reasoning and world knowledge. The Transformer architecture, which is central to these models, utilizes self-attention mechanisms to weigh the importance of different words in a sentence or text segment, allowing the model to generate contextually relevant responses or analyses.

The structure of an LLM like GPT is based on layers of Transformer blocks, each consisting of multi-head self-attention mechanisms and fully connected neural networks. These layers work together to process input text, with earlier layers learning to recognize basic patterns in the data, such as common word pairings or sentence structures, and deeper layers learning to grasp more complex relationships and abstract concepts. The output of the model is a probability distribution over possible next words or tokens, enabling the generation of coherent and contextually appropriate text. Training an LLM is a resource-intensive process that requires vast amounts of computational power. The models are trained using a method called unsupervised learning, where the model is presented with text and learns to predict the next word in a sequence based on the words that precede it. This process, known as autoregressive training, allows the model to learn a comprehensive understanding of language structure and content without needing explicit labels or annotations for the training data [Wei et al., 2022].

One of the key technical innovations in LLMs is their ability to handle long-range dependencies in text. Traditional models struggled to maintain context over longer passages, but the self-attention mechanism in Transformer models allows for direct relationships to be learned between words, regardless of their position in the text. This capability enables LLMs to maintain coherence over longer texts and understand complex, nuanced relationships between concepts. Issues such as bias in the training data, potential for misuse, and the environmental impact of training large models are areas of ongoing concern and research. Understanding and mitigating how these models might generate misleading or incorrect information remains a critical area of focus. LLMs embody a significant advancement in the field of NLP, offering unprecedented capabilities in understanding and generating human language. Their development is a testament to the progress in deep learning and neural network architectures, opening up new possibilities for human-computer interaction. As these models continue to evolve, they hold the promise of further bridging the gap between human and machine communication, making interactions more natural, efficient, and insightful.

2.4.1 LLM's Applications

LLMs have found a wide range of applications across various industries, showcasing their versatility and power. In customer service, LLMs automate and personalize interactions, significantly improving response times and customer satisfaction. For content generation, they create everything from articles to code, demonstrating an ability to produce coherent and contextually relevant text. In the realm of fact-checking, LLMs analyze vast amounts of

data to verify the accuracy of claims, enhancing the reliability of information disseminated online [Wei et al., 2022].

Particularly impactful is their role in the healthcare domain, where LLMs process and analyze large datasets of medical texts, research papers, and patient records. This capability supports medical professionals by providing quicker access to diagnostic information, treatment options, and the latest research findings [Jiang et al., 2020]. By automating the analysis of medical literature, LLMs also facilitate the identification of trends and patterns, potentially accelerating the discovery of new treatments and understanding of diseases. The application of LLMs exemplifies their potential to not only streamline operations but also contribute significantly to advancements in medical research and patient care. Despite their impressive capabilities, LLMs face significant challenges and limitations. One major issue is their tendency to generate plausible but inaccurate information, complicating tasks requiring high precision. Additionally can perpetuate and amplify biases present in their training data, raising ethical concerns about their applications. Furthermore, the environmental impact of training these models is considerable due to their high energy consumption, prompting a need for more efficient computing techniques and sustainable practices in AI development. These challenges highlight the importance of ongoing research and ethical considerations in the advancement of LLMs technologies.

2.4.2 Challenges and Ethical Considerations

LLMs have found a wide range of applications across various industries, showcasing their versatility and power. In customer service, LLMs automate and personalize interactions, significantly improving response times and customer satisfaction. For content generation, they create everything from articles to code, demonstrating an ability to produce coherent and contextually relevant text. In the realm of fact-checking, LLMs analyze vast amounts of data to verify the accuracy of claims, enhancing the reliability of information disseminated online [Bender et al., 2021].

Current research trends in LLMs are driven by the need to address their limitations while expanding their capabilities and applications. One significant area of focus is the development of more energy-efficient models, reducing the environmental impact associated with training and operating these computationally intensive systems. Researchers are exploring new algorithms and hardware optimizations that can lower energy consumption without compromising performance.

Enhancing the reasoning capabilities of LLMs is another critical area of advancement. Efforts are being made to improve models' understanding of

complex logical structures and their ability to apply this understanding in problem-solving contexts. This includes training LLMs to perform better on tasks that require deep reasoning and comprehension, such as advanced question-answering and decision-making.

Adapting LLMs to specific knowledge domains, particularly in sectors like healthcare, is also a priority. Tailoring models to understand and generate text related to specialized fields requires fine-tuning with domain-specific data, improving their accuracy and utility in professional contexts. In healthcare, this means developing LLMs that can interpret medical literature, patient reports, and research findings with a high degree of precision, assisting medical professionals in diagnosis, treatment planning, and keeping abreast of the latest medical research.

2.4.3 Limits

LLM such as GPT-3.5 and its successors mark a transformative advancement in the realm of NLP, offering unprecedented capabilities in generating coherent, context-aware text and facilitating a broad spectrum of language-based applications. Despite these groundbreaking achievements, LLMs come with a set of limitations that are crucial to acknowledge for their responsible utilization and future development [Wei et al., 2022]. One of the primary challenges associated with LLMs is their occasional struggle with accuracy and appropriateness in generating responses for highly specialized or nuanced queries. While these models excel in producing human-like text, their performance can be inconsistent when tasked with complex subject matter, leading to outputs that, while plausible, may be factually incorrect or contextually mismatched.

Bias in LLMs presents another significant concern. These models are trained on vast datasets compiled from the internet, mirroring the biases inherent in their training materials. Consequently, LLMs can perpetuate and amplify existing stereotypes and prejudices, reflecting the societal biases captured in their training data.

Furthermore, LLMs are limited by their static knowledge base, which is confined to the information available up to the point of their last training update. This limitation restricts their ability to process or provide real-time information, rendering them less effective for tasks requiring up-to-date knowledge or insights into recent events.

The computational resources required for training and operating LLMs pose additional challenges. The environmental impact and financial costs associated with these demands can be substantial, raising concerns about scalability, accessibility, and the broader implications of deploying such resource-

intensive technologies.

Lastly, the complexity and opacity of LLMs complicate efforts to understand or interpret their decision-making processes. This "black box" nature hinders transparency and accountability, making it difficult to diagnose errors, understand model reasoning, or provide explanations for specific outputs.

In LLMs models, "hallucinations" [Smith et al., 2022] refer to instances where the model generates false, inaccurate, or completely fabricated information as though it were true. This phenomenon is particularly evident in text generation tasks, where the goal is to produce coherent and informative content based on the context provided by the prompt. Despite their advanced ability to understand and generate natural language, GPT models can sometimes "hallucinate" details or data, leading to outputs that can be misleading or incorrect. These hallucinations can be categorized into several types:

- *Factual hallucinations*: The model presents statements as facts that are objectively false or unverifiable. For example, it might generate an incorrect historical date, erroneously attribute a quote to the wrong person, or invent events that never occurred.
- *Detail hallucinations*: Here, the model adds specific details that were not present in the prompt and that have no basis in reality, such as names of people, specific places, or precise numerical data, without any verifiable source to support these details.
- *Coherence hallucinations*: In this case, the model generates text that is internally inconsistent or contradicts information previously provided in the same generated text. This can include changing the gender of a character, altering the course of events without explanation, or jumping from one topic to another in ways that do not follow a clear logic.

The causes of hallucinations in these models are multifaceted and can include limitations in understanding context, in maintaining coherence over long stretches of text, or simply in the nature of the training data itself. These models are trained on vast corpora of text sourced from the internet, which include a range of information quality and accuracy. As a result, the models may learn text generation patterns that do not always correspond to objective reality or that reflect the inaccuracies present in the training data.

Addressing the issue of hallucinations is a key challenge in LLMs research and development. Techniques such as training with more accurate and curated data, employing post-generation information verification methods, optimizing prompts to guide models towards more accurate responses,

and developing evaluation metrics that penalize hallucinations, are all active areas of research aimed at reducing the frequency and impact of this phenomenon.

2.4.4 OpenAI

Before delving into the essence and transformative impact of ChatGPT on the artificial intelligence landscape, it's essential to acknowledge the originators behind it: OpenAI, a private research organization established with the mission to advance and steer AI technologies towards generating tangible benefits for humanity. Initiated in 2015 with the initial aim of developing AI-driven tools specifically for Video Games, OpenAI was brought to life by a collective of researchers including visionaries like Elon Musk and Sam Altman (the latter serving as the current CEO), based in San Francisco.

OpenAI adopting a policy of making patents publicly accessible to ensure they can be found and utilized, provided they pose no threat to security, represents a cornerstone of the company's ethos. Transitioning from a nonprofit to a for-profit entity, OpenAI LP was officially named in 2019. It was in 2021 that OpenAI unveiled Dall-E, a groundbreaking generative AI model capable of creating images from textual descriptions. However, the true jewel in the company's crown, which significantly altered its trajectory, is ChatGPT, launched in November 2022—a topic that will be explored further in the following section.

As previously noted, GPT stands for *Generative Pretrained Transformer* [Lewis et al., 2020] and the term "chat" indicates its function as a chatbot, designed to interact with humans in a manner that mimics real-life conversation as closely as possible.

It has been refined using a method known as *Reinforcement Learning from Human Feedback* (RLHF) [Christiano et al., 2017], utilizing a vast compilation of data amounting to around 300 billion words (encompassing internet content, books, Wikipedia) gathered up until 2021. The substantial size of these models and their capability for immediate responses entail significant computational expenses.

GPT Models

The technical foundation of GPT-4 involves a sophisticated mechanism for processing vast and complex sequences of data, enabling the model to capture the intricacies of human language in unprecedented detail.

At the heart of GPT-4 lies the Transformer architecture, which employs multi-head attention mechanisms to model dependencies between all tokens

in an input sequence, irrespective of their distance from one another [Wei et al., 2022]. This feature is crucial for comprehending complex context and semantics. Each Transformer block within the model comprises two main components: a multi-head attention mechanism and a positionally-fed forward neural network. These blocks process the sequential input in parallel, significantly enhancing computational efficiency compared to models based on RNNs or *long short-term memory* (LSTM [Hochreiter and Schmidhuber, 1997]) networks.

The multi-head attention mechanism allows the model to focus on different parts of a sentence simultaneously, capturing a variety of linguistic dynamics and improving the model's understanding of text context and coherence. Following the attention mechanism, the output undergoes processing by a positionally fed forward neural network. This network applies identically across each sequence position, ensuring that every token is processed in the context of others.

2.4.5 Prompt Engeneering

The practice of *prompt engineering* [Brown et al., 2020] has become a pivotal technique within the realm of NLP, particularly with the rise and adoption of LLMs like the *Generative Pre-trained Transformer* (GPT) series [Radford et al., 2019]. This methodology focuses on the strategic crafting of textual inputs (prompts) to guide language models in generating desired outputs, optimizing human-machine interactions for specific applications. The efficacy of prompt engineering stems from its ability to leverage the vast, implicit knowledge acquired by models during pre-training, enabling them to perform tasks without the need for task-specific training or labeled data. The context of fact-checking, prompt engineering plays an especially vital role. By crafting well-designed prompts, it's possible to direct LLMs towards accurately analyzing claims, verifying truthfulness, and identifying reliable sources, significantly contributing to the fight against misinformation. The main challenge in this area lies in formulating prompts that are both specific, to avoid interpretative ambiguities, and flexible enough to adapt to various contexts and information formats.

Research in the field of prompt engineering has expanded significantly, exploring various strategies to maximize prompt effectiveness. These include tailoring prompts to specific tasks, employing machine learning techniques to optimize prompt formulation, and developing iterative approaches that refine prompts based on model feedback. A key insight from these studies is the importance of a deep understanding of both model capabilities and natural language characteristics, to design prompts that effectively guide

models towards accurate and relevant responses.

There is also growing interest in automating prompt engineering, where artificial intelligence algorithms are used to dynamically generate and evaluate prompts, reducing human resource burden and increasing process efficiency. This approach has shown particular promise for scalable applications, where the ability to rapidly adapt prompts to new tasks or changes in information flow is crucial.

However, despite advancements, prompt engineering in the context of LLMs still faces significant challenges [Brown et al., 2020]. Balancing specificity and generality, managing unexpected or inaccurate responses from models, and the difficulty of assessing prompt effectiveness in complex scenarios are all open issues that require further research. Moreover, the matter of transparency and ethics in prompt design raises important questions regarding responsibility and social impact of LLMs usage. These techniques are central to leveraging LLMs for a wide range of tasks, including but not limited to natural language understanding, generation, and fact-checking.

- *Zero-Shot Learning*: it refers to the model's ability to perform a task without any specific training on that task. The model relies solely on its pre-trained knowledge and the information provided in the prompt. A zero-shot prompt typically includes a detailed description of the task at hand and might ask the model to make predictions or generate responses based on this description alone.

- *Few-Shot Learning*: it involves providing the model with a small number of examples (shots) within the prompt to guide its understanding of the task. This method relies on the concept that a few examples can help the model generalize from its pre-trained knowledge to the specific task context. Each example typically consists of a task description, a specific instance of the task, and the correct output.

- *One-Shot Learning*: as a specific case of few-shot learning, one-shot learning provides the model with exactly one example to learn from. This method tests the model's ability to generalize from a single instance. In prompt engineering, a one-shot prompt for fact-checking might include one statement along with its verification status to help the model understand what kind of output is expected when presented with a new statement.

- *Many-Shot (or Multi-Shot) Learning*: this approach extends the few-shot methodology by providing the model with a larger number of examples. While still not as extensive as a full dataset used in traditional

machine learning training, many-shot learning can give the model a more comprehensive understanding of the task nuances. This method is particularly useful for more complex tasks or when the desired output is highly nuanced.

- *Chain-of-Thought Prompting*: a recent advancement in prompt engineering, chain-of-thought prompting involves constructing prompts that guide the model through a step-by-step reasoning process. This method is designed to tackle complex questions or tasks that require multi-step reasoning. The prompt encourages the model to "think aloud" by generating intermediate steps or reasoning paths that lead to the final answer. This approach has shown promise in improving the performance of LLMs on tasks requiring deep comprehension or logical reasoning.

- *Prompt Tuning and Prompt Programming*: beyond the above-mentioned methods, there is also research into more sophisticated techniques like prompt tuning, where the prompts themselves are optimized through gradient descent, akin to how model parameters are tuned. Prompt programming involves crafting prompts that explicitly encode instructions or algorithms for the model to follow, combining principles from computer programming with NLP.

Each of these prompting techniques offers unique advantages and challenges, and the choice of which to use depends on the specific task, the available data, and the capabilities of the LLM being employed. As research in this area continues to evolve, even more nuanced and effective prompting strategies will likely emerge, further expanding the versatility and efficacy of LLMs across a wide range of applications. Prompt engineering represents a critical frontier in the evolution of NLP and LLMs, offering powerful tools to improve human-AI interaction. Its application in fact-checking highlights the technique's potential to contribute to more accurate and reliable information.

2.5 Disinformation

Disinformation [Westa and Bergstromb, 2019] is a term that refers to the intentional spread of false or misleading information with the aim of deceiving or manipulating people. It can be disseminated through various means such as traditional media, social media, and personal conversations. It is important to note that disinformation differs from simple dissemination of incorrect information as it is intentional and aims to influence public opinion

or achieve specific objectives. Disinformation represents a problem of growing global significance, reaching proportions that can be described as a crisis. This phenomenon has a significant impact on various spheres of society, and it is crucial to fully understand its extent and implications.

Disinformation undermines the ability to identify solutions to global problems and compromises the formation of a collective consensus based on accurate information, thus hindering the search for effective solutions. One of the key factors contributing to the spread of disinformation is the change in communication technology and the monetization of information. In the past, when news revenue was based on subscriptions, there were fewer incentives to publish catchy but inaccurate articles. However, with the rise of online platforms, the focus has shifted towards generating clicks and engagement, leading to a higher probability of sensationalized or misleading content.

One of the sectors where disinformation has an immediate impact is public health, where the spread of false information can negatively influence the public response to control measures. An evident example is the dissemination of disinformation during the SARS-CoV-2 pandemic, which compromised the management capacity of health authorities and put public health at risk. The World Health Organization even declared an "infodemic" [Smith and Johnson, 2022] due to the massive spread of erroneous information related to the pandemic, making the danger of disinformation and its ability to spread in an era of global communication even more apparent. Other areas where news has a relevant impact are politics and current affairs news, such as the United States elections in 2016 [Allcott and Gentzkow, 2017].

There are several issues associated with this problem; the rapid dissemination of information on the Internet makes it complex to identify and monitor disinformation before it reaches a wide audience. Disinformation dissemination tactics constantly evolve, making it difficult to develop effective detection methods that can adapt to the changing information landscape. The complex language and technical context characteristic of health information can make it difficult to distinguish between accurate and erroneous information. Limited access to expert opinions in the field, capable of providing accurate assessments of health information, represents an additional challenge in verifying the correctness of content, as divergent opinions among experts on health topics can generate conflicting information, making it challenging to determine the truthfulness of the content. Emotions and biases can influence the perception and interpretation of health information, making it difficult to objectively evaluate its accuracy. Furthermore, the detection of health-related disinformation may involve the analysis of personal health data, raising concerns about privacy and ethical issues.

In the context of disinformation in the healthcare domain, it is crucial

to explore the most effective approaches for its online assessment [Barve and Saini, 2021]. Some of these methods have been developed to tackle the growing problem of the dissemination of erroneous and potentially harmful health-related information on the Internet.

Manual approaches, while the most accurate, are limited by the vast amount of online information, making detailed analysis of individual articles or blogs on a large scale impractical. Another approach involves examining user behavior, where the actions of users, including their interaction with content and feedback, are analyzed. The goal is to assess the credibility of health information based on how users interact with it. This approach has provided an overview of misinformation dissemination patterns and has helped identify potential indicators of falsehood. An approach that has seen exponential growth in recent years is the use of machine learning algorithms [Viviani and Sotto, 2022] combined with various features of online content. These approaches enable the development of sophisticated tools to automatically detect online misinformation. One central aspect involves using textual representation features based on the analysis of text in online content, such as keywords, word counts, and linguistic features. Other features focus on social interactions associated with online content, such as the number of shares, comments, and ratings. This analysis helps detect the spread of disinformation through social networks, highlighting its influence on the public.

More recently, deep learning models have been introduced to address the problem. These models utilize neural networks to analyze the content of web pages, particularly extracting local and global features of texts and information present on the pages. Some strategies integrate the use of knowledge graphs to assess the credibility of online health information. These models leverage medical knowledge graphs and bipartite article-entity graphs to propagate node representations representing web pages, classifying health information as authentic or non-authentic. These advanced approaches represent a significant research frontier in online misinformation detection [Kim et al., 2023].

2.6 Fake News

A clear and accurate definition lays solid foundations for the analysis of *fake news*. Referring to the work [Zhou and Zafarini, 2020], there is no universal definition of *fake news*, not even within the realm of journalism. Current research frequently links fake news with terms and concepts like deceptive news, fabricated information, satirical content, misinformation, disinformation, selective reporting, clickbait, and rumors. By analyzing the definitions

of these terms and concepts, we can identify fake news through three distinct traits: veracity (the presence of inaccurate claims), intent, and the nature of the information.

As mentioned earlier, there is no universal definition of fake news to date, which has been considered, for example, as "a news article that is intentionally false" or "an article or message published and propagated through the media, conveying false information regardless of the means and motivations behind it." Moreover, the concept of news is more difficult to define because it can change depending on the context [Allcott and Gentzkow, 2017]. The broad interpretation aligns with prevalent research and datasets on disinformation provided by fact-checking entities. This hones in on deceptive news aiming to appear credible, spotlighting both the genuineness of the information and the motives behind its propagation.

2.7 Fact-Checking

The definition of *fact-checking* refers to the process of verifying the accuracy and truthfulness of claims, statements, or information [Zhou and Zafarini, 2020]. Fact-checking involves several crucial steps to ensure precision and reliability. In the initial phase called claim detection, it involves identifying, filtering, and prioritizing claims from various sources, such as social media, political campaigns, and public speeches. The credibility of a claim is a crucial aspect in evaluating its truthfulness and reliability. To determine if a claim is trustworthy, it is essential to consider several key factors [Zeng et al., 2021].

First, the source of the claim plays a fundamental role, as sources with an established reputation or recognized expertise in the field in question tend to be more reliable. It is also important to examine whether the claim is supported by solid empirical evidence, verifiable data, or scientific research. Consistency with other sources is another significant indicator; when different reliable sources converge on a claim, this strengthens its credibility. Similarly, the consensus of experts in the field can add weight to the claim.

The context in which the claim was made must also be considered, as credibility can vary significantly depending on whether it is a personal opinion or a scientific statement. Bias and personal interests must be carefully examined, as they can influence the credibility of the source. A claim made by a source with an obvious personal interest can raise doubts about its objectivity. The method of information collection is another key factor, as a rigorous and well-structured approach to information collection contributes to the claim's credibility. Verifiability is also essential, as claims that can be

independently verified are generally considered more reliable [Lazarsk and Mahmood Al-Khassaweneh, 2021].

Finally, the source's history is an important indicator, as a source with a history of accuracy and reliability is more likely to produce credible claims. Subsequently, evidence collection is crucial for fact-checking, and evaluating the provenance helps assess the quality of information sources, while evidence is collected to support or refute the claim. The next phase refers to fact-checking and evaluating the truthfulness, involving the analysis of the claim and the evidence collected to determine its truthfulness;NLP techniques are used to analyze the linguistic properties of the claim and assess its truthfulness through models and machine learning algorithms. The last phase is the analysis of results, which must be understandable to all users; justifications and explanations are provided to clarify the model's determinations and help users understand the reasoning behind the fact-checking result.

There are various methodologies and approaches [Barve et al., 2023, Thorne and Vlachos, 2018] that can be used depending on the study's objectives or field of application. The most classic approach is manual and involves carefully analyzing information sources to assess their credibility and reliability. Fact-checkers consider the reputation and potential bias of such sources to determine if the information provided is reliable. Human evaluators can also provide feedback on the quality and effectiveness of the process.

Another common approach is comparing information from different reliable sources; this method allows for comparing and verifying the accuracy of claims through a cross-evaluation of available data and information. For claims requiring specialized expertise, fact-checkers consult industry experts for an accurate assessment; these experts bring their experience and analysis to validate or refute claims based on their specialized knowledge.

Data analysis is another fundamental methodology where fact-checkers use datasets based on empirical claims, surveys, and scientific research to assess the accuracy of claims. Datasets with known truth labels can be used to evaluate the performance of automated systems. These datasets contain claims and the corresponding fact-checking verdicts, allowing for a quantitative assessment of the system's accuracy.

Another methodology involves using patterns generated from sources like WikiData [Wikidata contributors] to identify claims that can be verified; these patterns are then sent to search engines to obtain the best results. A widely used approach combines the previous ones by using various sources of evidence, such as fact databases, the internet, and other external sources, along with analyzing the origin of the claims and the language used; to train fact-checking models, specific corpora and datasets containing articles, newspapers, and encyclopedic texts are employed. Search engines can also be

integrated to search for information related to the claims to be verified, for example, Claim Buster integrates search results into its database of claims.

Analyzing the origin of the claim and the language used is exploited to improve the fact-checking process and the identification of specific linguistic features, such as interrogative phrases or the count of pronouns. Comparing the performance of different systems or approaches can provide insights into their effectiveness.

These methodologies can be integrated with user studies that involve collecting feedback from users interacting with fact-checking tools or platforms; these include surveys, interviews, or usability tests to assess user satisfaction, understanding, and trust in the process. Crowdsourcing platforms are also used to collect judgments from a large number of users and assess the veracity of claims. This approach leverages the collective wisdom of the crowd to evaluate the accuracy of information.

These sites employ a team of experts who investigate and verify the accuracy of claims in the medical and health field; they use a combination of research, evidence-based information, and expert opinions to debunk false claims. Although automatic verification through NLP is feasible with current technology, the best results are obtained when working on restricted domains of facts or resorting to advanced search engines, as well as a detailed analysis of the author and source of the statements.

Fact-checking is a highly complex and sensitive challenge due to the severe consequences that misinformation can entail. Various techniques and methodologies are proposed to minimize errors and enhance accuracy [Burel et al., 2022, Viviani and Sotto, 2022]. Sentiment Analysis involves analyzing the sentiment of the text to determine whether it contains positive or negative information. Articles or blogs with a higher number of positive words are more likely to contain truthful information. Document Similarity Measures utilize metrics such as Euclidean distance, cosine similarity, and Jaccard distance to measure the similarity between documents. By comparing the similarity of a document with a manually fact-checked reference dataset, misinformation can be identified. Machine Learning Classifiers apply algorithms to classify URLs or documents as true or false based on features extracted from the text, such as sentiment, grammar, and document similarity measures.

2.8 Fact-Checking with Knowledge Base

Before delving into the intricacies of the fact-validation procedure, let's acquaint ourselves with key definitions borrowed from relevant literature [Nickel

et al., 2012, Zhou and Zafarini, 2020]:

DEFINITION 1. ***Knowledge*** *A compilation of triples (Subject, Predicate, Object) (SPO) derived from provided data, illustrating the conveyed knowledge.*

DEFINITION 2. ***Fact*** *A fact is a knowledge (SPO triple) verified as truth.*

DEFINITION 3. ***Knowledge Base*** *Knowledge Base is a set of facts.*

DEFINITION 4. ***Knowledge structure*** *A knowledge structure embodies the SPO triples in an information repository, portraying entities (subjects or objects in SPO triples) as nodes and relationships (predicates in SPO triples) as edges.*

To ascertain the accuracy of articles, a comparison between the knowledge extracted from the news content (SPO triples) and the established facts (knowledge) is imperative. The information repository serves as a dependable source for fundamental truths, assuming that existing triples in the information repository denote verifiable assertions[Luo and Long, 2020]. The initial methodology [Zhou and Zafarini, 2020] involves a correspondence check between the triples gleaned from articles and those within the graph. However, for non-existent triples, their legitimacy is contingent upon three conjectures:

- **Closed-World Presumption**: Absent triples denote false knowledge.
- **Open-World Presumption**: Non-existent triples indicate unfamiliar knowledge, potentially true or false.
- **Localized Closed-World Presumption**: The validation of non-existent triples follows this rule: Assuming T(s, p) represents the set of existing triples in the knowledge structure for subject s and predicate p.

Depending on the selected presumptions and the domain of application, certain aspects of fact-checking procedures may fluctuate. Generally, the methodology encompasses three stages:

- **Identification of Entities**: Aligning the subject (or object) with a graph node mirroring the same entity as the subject (similarly, the object); diverse techniques can be employed to identify accurate matches.
- **Verification of Relationships**: The veracity of the triple (Subject, Predicate, Object) is affirmed if an edge (Predicate) in the KS connects nodes representing the Subject and Object. In the absence of such a connection, its legitimacy depends on the previously posited assumption, for instance, being considered false according to the closed-world presumption or ascertainable post inference.

- **Inference of Knowledge**: When the triple (Subject, Predicate, Object) is absent from the KS, the likelihood of an edge labeled as Predicate existing from the node representing Subject to the one representing Object in the knowledge structure can be computed. Various prediction methods, such as semantic proximity, discriminative predicate paths, or LinkNBed , can be employed. This step is discretionary and hinges on the earlier hypotheses.

In the work [Huynh and Papotti, 2018] is laid for a benchmark enabling the comparison of fact-checking algorithms based on external information in the form of RDF knowledge bases. The study introduces various fact-checking algorithms, categorizing them into three main types: Path Based Algorithms, Sub-Graph Based Algorithms, and Embedding Based Algorithms. They utilize DBPedia as their RDF knowledge base, sourced from Wikipedia's extracted triples. The evaluation employs diverse metrics, including the ROC curve, to assess model accuracy. In the study [Zhou and Zafarini, 2020], a graph-based fact-checking approach is proposed. Utilizing the knowledge graph in information verification involves two primary steps: fact extraction and knowledge comparison. During the fact extraction phase, relevant information is sourced from various outlets to construct a robust knowledge graph. This serves as a structured database of factual information, drawing from traditional sources like Wikipedia and specialized fact-checking websites. The primary objective is to gather as many accurate facts as possible to comprehensively populate the graph. Once the graph is constructed, the knowledge comparison phase involves matching the knowledge assertions in a news article for verification with the stored facts within the graph. The goal is to evaluate the authenticity of the news article by comparing the similarity or probability of the knowledge assertions in the article. To execute this verification, the news article is represented as a set of knowledge assertions expressed in Subject-Predicate-Object (SPO) triples. Each triple signifies a specific piece of information in the article. Subsequently, the knowledge assertions in the news article are compared with the facts within the graph. The objective is to find correspondences between the entities (subjects and objects) and predicates (relationships) of the knowledge assertions in the article and those stored in the graph. If a direct match for a specific knowledge assertion is not found, the probability of a relationship existing between the subject and object can be inferred using link prediction methods. A similar approach is proposed in the paper [Kim et al., 2023] where the fact-checking process involves extracting triples from the statement and comparing them with triples in the knowledge graph. The statement is represented as a textual declaration that can be verified against the knowledge graph. The system uti-

lizes a dataset called FACTKG, comprising natural language statements and corresponding RDF triples from DBpedia. The dataset is categorized into five types of reasoning: One-hop, Conjunction, Existence, Multi-hop, and Negation. The system generates statements based on the types of reasoning. For instance, one-hop statements cover only a single knowledge triple, whereas conjunction statements involve multiple triples connected by logical conjunctions. These statements are formulated based on specific models and rules for each type of reasoning. To match the statements with RDF triples, the system performs triple matching. It compares the entities and relationships mentioned in the statement with those in the RDF triples. The goal is to find evidence in the knowledge graph that supports or refutes the statement. In the paper [Barve et al., 2023], a knowledge graph containing a set of verified facts is created. The fact-checking process involves comparing the statement with the knowledge base and performing classification. The verification process using knowledge graphs includes the following steps:

- *Fact Extraction*: The first step involves extracting relevant facts from various sources and creating a knowledge base; these facts are verified and confirmed as true.
- *Assertion Verification*: The assertion or statement is compared with the facts in the knowledge graph. The statement is analyzed to identify keywords and entities that can be compared with the facts in the graph.
- *Classification*: Based on similarity measures such as Jaccard distance or Euclidean distance, the assertion is classified as true or false.

If the assertion closely matches the verified facts in the knowledge graph, it is considered true. Otherwise, it is classified as false. In another study [Cui et al., 2021], reference is made to DETERRENT, a model specifically designed for detecting misinformation in the healthcare sector. It combines a text encoder, such as a *Bidirectional Gated Recurrent Unit* [Cho et al., 2014], with a knowledge graph. The model uses information propagation and attention mechanisms to capture relationships between entities in the knowledge graph and the text. By modeling both positive and negative relationships, the model can effectively detect misinformation in the health field. This system also relies on comparisons between triples; the model matches triples in the knowledge graph with information in health-related articles using a knowledge-guided graph. The verification process using knowledge graphs involves extracting verified facts, comparing assertions with the knowledge graph, measuring similarity, and classifying assertions based on their credibility.

Chapter 3

Modelling and Studying the Problem

This chapter presents the work carried out, describing the methodologies used to enhance the effectiveness and efficiency of automated fact-checking through leveraging NLP, LLMs and graph-based techniques [Hogan et al., 2023]. The chapter is positioned after a comprehensive background and related work discussion, setting the stage for a deep dive into the innovative models proposed to tackle the challenges inherent in fact-checking. This section details the step-by-step process involved in fact-checking, from data collection and pre-processing to analysis, triple extraction, generation of new claims and explanations, and ultimately, semantic and graph-based matching. It aims to provide an in-depth understanding of the technical strategies and algorithms developed or adapted, as well as the methodological choices underpinning the research. Through a systematic and detailed narrative, the chapter elucidates how datasets are explored and prepared for analysis. Significant focus is placed on the Triple Extraction process detailed in Section 3.4, both in its lemmatized form and through the generation of new triples, revealing how information is structured to facilitate comparison with the knowledge base. The section 3.5 Generating news claims and explanations describes the methods used to generate new claims and explanations, this is to try to improve the semantics of the original text . Another improvement we would like to have with the new claims is to improve matching with triple The discussion on Matching detailed in Sections 3.6.1 and 3.6.2, both triple-based and semantic, addresses the core of the veracity analysis, comparing claims against a reliable knowledge base. The chapter on modeling provides a detailed and technically rigorous exposition of the research methodologies, laying the foundation for the subsequent chapters on evaluation results and conclusions. The following papers were followed as benchmarks: [Huynh and

Papotti, 2018, Luo and Long, 2020, Zhou and Zafarini, 2020].

3.1 Pipeline

The methodology is grounded in a comprehensive approach that begins with the establishment of a robust knowledge base, derived from selected dataset of literature [Kotonya and Toni, 2018]. Building knowledge base, consisting exclusively of texts validated as "true," forms the cornerstone against which the veracity of claims from our test dataset is assessed. To construct this base, we exclusively extract texts labeled as true, thereby ensuring a foundation built on verified facts. The process then advances to the extraction of triples—structured sets of subject, predicate, and object—from this vetted knowledge base. More extraction methods are employed, each bringing unique capabilities to the task of parsing and structuring the information. In parallel, the generation of new text from each entry in the knowledge base is initiated using generative models, with careful selection of prompts to optimize effectiveness in the subsequent matching phase. This generation is conducted with a dual focus: on creating new claims from the extracted triples and original claims, and on formulating new explanations from these new claims and the original explanations. Attention is then shifted to the test dataset, similarly annotated for truthfulness, where the process of triple extraction and generation of new claims and explanations is replicated using the same methodologies. This ensures a consistent framework for analyzing and comparing the information across the knowledge base and the test dataset. During the triple matching phase, the triples associated with each claim from the test dataset are compared against those within the "true" knowledge base. The objective is to ascertain whether even a single match between the test claim's triples and those in the knowledge base is sufficient to classify the claim as true (method "a") or whether a stricter criterion, such as a match of over 50% of the triples (method "b"), is required. Furthermore, the analysis is extended to the newly generated claims, applying the same rigorous extraction and matching processes. This recursive approach allows for the assessment not only of the original claims but also of the dynamically generated content, ensuring a comprehensive evaluation of the fact-checking capabilities.

In addition to triple matching, the veracity of new claims and new explanations is evaluated through semantic analysis, comparing generated summaries from the test texts against those in the knowledge base using word embeddings. This involves calculating cosine similarity between the embeddings, with thresholds set at 50%, 70%, and 90% to determine the text's

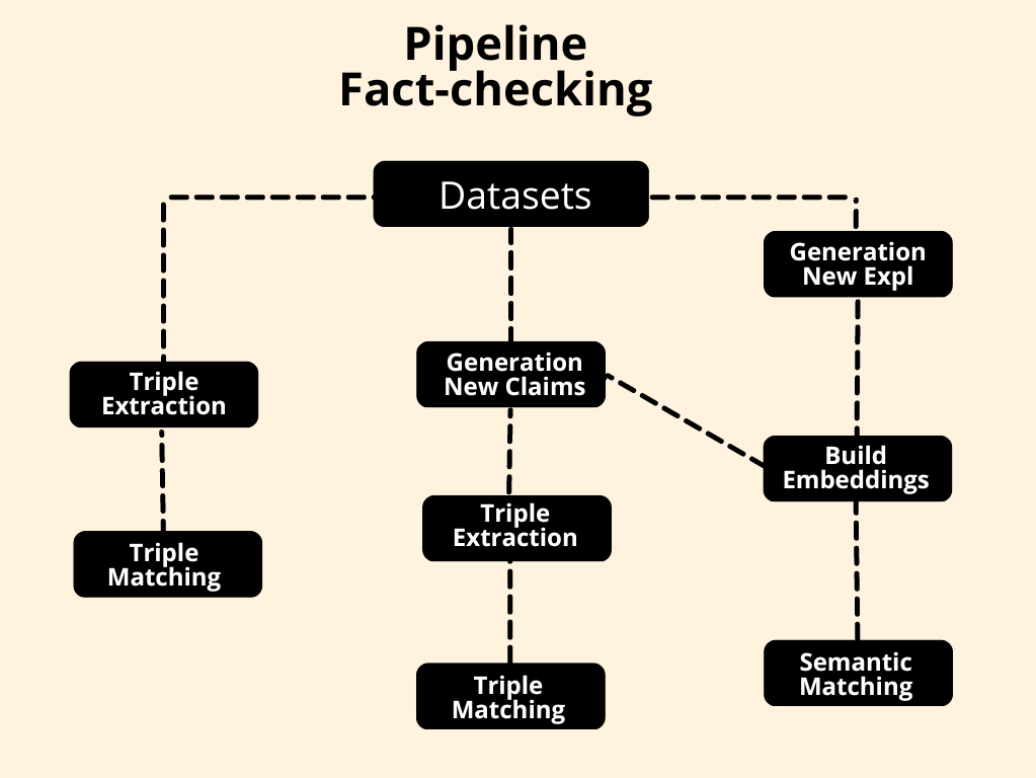

Figure 3.1: The pipeline followed in this work

truthfulness. This method offers an in-depth understanding of semantic relationships and provides an alternative mechanism for claim verification. Upon completing the matching phase and applying truth and falsehood labels to the test dataset, the process proceeds to calculate key performance metrics such as accuracy and F1-*score*. These metrics serve as critical indicators of the effectiveness of the fact-checking models, allowing for an in-depth evaluation of the various methods and approaches utilized throughout this research. Through this detailed pipeline, the thesis aims to contribute significantly to the field of automated fact-checking. By incorporating a cycle of claim generation, triple extraction, and rigorous matching for both original and newly generated claims, a novel, comprehensive approach designed to tackle the multifaceted challenges.

3.2 Exploration Data Analsysis

Exploratory data analysis (EDA) is a fundamental approach for analyzing and studying data sets to summarize their main characteristics, often through

visual methods. This foundational step is key to gaining a deeper understanding of the data's distribution, trends, and patterns before applying any statistical modeling or machine learning techniques. It serves to uncover patterns, trends, and relationships within the data that might not be immediately apparent, offering a lens through which analysts can observe and interpret the underlying structures of the data. Through this, anomalies or outliers are identified, which could indicate measurement errors, data entry mistakes, or genuine points of interest, thus ensuring the data's integrity and reliability. This analytical approach provides a foundation for hypothesis testing, allowing for the validation or refutation of assumptions made about the data prior to deep analysis. It aids in determining the most appropriate modeling techniques for the dataset at hand and identifies any necessary data transformations before modeling begins. This preparation phase is crucial for aligning the data analysis process with the objectives of the project, ensuring that subsequent models are built on a solid and well-understood foundation. Additionally, plays a vital role in data quality assurance, spotlighting potential issues such as missing values, data inconsistency, or normalization needs, thus guiding the cleaning and pre-processing steps. EDA is an indispensable step in data-based studies, providing a robust understanding of the available dataset, highlighting potential issues, and informing methodological decisions for subsequent analysis phases.

3.2.1 Dataset Pubhealth

The dataset chosen for the analysis and fact-checking of public health claims is named PUBHEALTH [Kotonya and Toni, 2018]. This dataset, composed of 11,067 claims, spans a wide array of topics from biomedical research to governmental health policies and other public health narratives, collected from an assortment of fact-checking and news review websites. Each claim is paired with journalist-authored explanations, serving as a gold standard for the verification labels assigned to these claims, thereby laying the groundwork for explainable automated fact-checking endeavors that demand specialized expertise. The construction of the PUBHEALTH dataset commenced with an extensive data scraping endeavor, harvesting 39,301 claims from notable fact-checking platforms such as Snopes, Politifact and FactCheck, among others. This initial collection was further enriched by claims sourced from the health sections of reputable news outlets, including Associated Press and Reuters, and the specialized review site, Health News Review. The data scraping process was designed to capture not only the claims but also the full texts of articles discussing their veracity, the explanations justifying the veracity labels, and the URLs of sources cited within these articles. This

comprehensive approach ensured the dataset's richness in content and context, vital for the nuanced task of fact-checking in the public health arena. A critical phase in the dataset's construction involved the standardization of veracity labels to a uniform classification system comprising four categories: true, false, mixture, and unproven. Furthermore, the dataset underwent a rigorous cleaning process, filtering out claims lacking a biomedical context and ensuring that each claim and its accompanying explanation were succinct yet informative, adhering to a character limit that balanced comprehensiveness with clarity. The dataset's uniqueness lies in its inclusion of journalist-crafted explanations alongside each claim, a pioneering feature that sets PUBHEALTH apart from existing fact-checking datasets. These explanations not only provide the rationale behind the veracity labels but also enrich the dataset with nuanced insights into the complex nature of public health information. This dataset stands as a valuable resource for the subsequent exploration of explainable fact-checking. The structure of the PUBHEALTH dataset is defined by several key columns, each playing a crucial role in understanding and analyzing the statements. These primary columns include:

- **claim_id:** Unique identifier associated with each claim in the dataset.
- **claim:** Text of the claim or statement undergoing fact-checking.
- **date_published:** Publication date of the claim.
- **explanation:** Field containing the explanation or justification provided for assigning a specific truth label to the claim.
- **fact_checkers:** Information about fact-checkers or organizations that have examined the claim.
- **main_text:** Main text associated with the claim, potentially including additional details or context.
- **sources:** Information about the sources cited or used to support the claim.
- **label:** The truth label assigned to each claim, with values such as "true," "false," "unproven," or "mixture."
- **subjects:** Topics or subjects addressed in the claim.

The veracity labels in the PUBHEALTH dataset provide a solid foundation for evaluating public health claims. Each label reflects a nuanced assessment, enabling the analysis of intricacies within statements and contributing to an overall understanding of the phenomenon. Furthermore the inclusion of an explanation field provides additional transparency to the fact-checking process, enhancing comprehension of the decisions made and fostering confidence in the obtained results. Its construction reflects a comprehensive approach to data collection, cleaning, and standardization, resulting in a dataset that not only serves as a foundational tool for NLP research but also contributes to the broader effort of ensuring the accuracy and reliability of public health information in the digital discourse. In the Figure 3.2 the first 5 lines of the test set are shown.

	claim_id	claim	date_published	explanation	fact_checkers	main_text	sources	label	subjects
0	15661	"The money the Clinton Foundation took from fr...	April 26, 2015	"Gingrich said the Clinton Foundation ""took m...	Katie Sanders	"Hillary Clinton is in the political crosshair...	https://www.wsj.com/articles/clinton-foundatio...	false	Foreign Policy, PunditFact, Newt Gingrich,
1	9893	Annual Mammograms May Have More False-Positives	October 18, 2011	This article reports on the results of a study...		While the financial costs of screening mammogr...		mixture	Screening,WebMD,women's health
2	11358	SBRT Offers Prostate Cancer Patients High Canc...	September 28, 2016	This news release describes five-year outcomes...	Mary Chris Jaklevic,Steven J. Atlas, MD, MPH,K...	The news release quotes lead researcher Robert...	https://www.healthnewsreview.org/wp-content/up...	mixture	Association/Society news release,Cancer
3	10166	Study: Vaccine for Breast, Ovarian Cancer Has ...	November 8, 2011	While the story does many things well, the ove...		The story does discuss costs, but the framing ...	http://clinicaltrials.gov/ct2/results?term=can...	true	Cancer,WebMD,women's health
4	11276	Some appendicitis cases may not require 'emerg...	September 20, 2010	We really don't understand why only a handful ...		"Although the story didn't cite the cost of ap...		true	

Figure 3.2: First 5 rows of the PubHealth test set

The initial dataset is bifurcated into two distinct parts: a training set and a test set. The training set comprises 9,832 rows, while the test set is made up of 1,235 rows. Both sets exhibit identical structures and columns, ensuring consistency and uniformity in the analysis and modeling approach. This division between training and testing data is crucial in data science and machine learning fields, as it allows for the training of models on one dataset and their performance evaluation on a completely separate set, testing their generalization capabilities. Focusing on the specified columns claim, label, and subjects, their significance and the roles they play in dataset analysis will be explored.

The **claim** column contains statements or assertions subjected to verification within the dataset context. These claims represent the basic unit of analysis and are crucial for understanding the data's nature, as each row presents a statement that can be examined for truthfulness. The **explanation** column serves as a comprehensive background or justification for the label assigned to each claim within the dataset. It provides a detailed account of the reasoning, evidence, or context that supports the veracity verdict

of the claim. The **label** column signifies the verdict on the claim's veracity. Labels may vary depending on the dataset's specific implementation but typically include categories such as true, false, misleading, or unverifiable. This column is essential for training classification models in the context of automatic fact-checking, as it provides the "correct answer" that the model aims to predict. Lastly, the **subjects** column categorizes the claims based on the topics they address. This categorization helps organize the claims into thematic groups, facilitating topic-specific analyses and potentially improving the models' ability to learn subtle differences between claim categories.

Analyzing these columns provides a solid foundation for understanding the dynamics within the dataset, allowing for the identification of patterns, trends, and potential challenges in classifying claims. For instance, an EDA might reveal if certain subject categories are more prone to contain false claims than others, or if the distribution of labels varies significantly across different topics. Understanding these aspects is crucial for any data science project as they provide insights into the dataset's structure and content, guiding subsequent analyses and modeling efforts.

First, let's analyze the training set; The label column in the training dataset has five unique values: 'false', 'mixture', 'true', 'unproven', and a special category for *missing values* (NaN). This diversity in labels is indicative of the complexity and nuance within the dataset. The majority of claims, precisely 5,078, are labeled as 'true', highlighting a potential imbalance in the dataset, in total there is 9832 claims. Moving on to the subjects column, it's evident that the dataset covers a wide range of topics, with a significant focus on health-related issues. The subjects with the highest number of claims include 'Health', 'Health News' and a 'General News' category, among others. This distribution suggests that the dataset is particularly rich in health and health news topics, which could be beneficial for tasks aimed at understanding or combating misinformation in these crucial areas. The training dataset presents a fascinating mix of labels and subjects, with a strong emphasis on health-related claims. The analysis of unique labels and the distribution of subjects lays the groundwork for further exploration and model development. It highlights the importance of considering label distribution and subject matter expertise in designing models that are both accurate and relevant to specific domains.

The test dataset has the same structure as the training dataset, of course, the numerosities change. Like in training the label column in the test dataset has four unique values: 'false', 'true', 'unproven', and 'mixture' and NaN. This indicates a slightly simpler structure compared to the training set, The most common label in this dataset is 'true', with 599 claims marked as such on 1235 claims. This prevalence of 'true' claims mirrors the training set,

suggesting a consistent focus across both datasets but also pointing to a potential bias towards verifiable claims. When looking at the subjects column, the test set appears to cover a broad range of topics with a total of 840 unique subjects. This diversity underscores the complexity and wide applicability of the dataset. The most represented subjects are 'Health', 'Health News' , 'Politics', 'General News' and 'National' . This distribution highlights a strong emphasis on health and health news, similar to the training set, but also introduces a significant focus on politics and national news, which could provide an interesting angle for analyzing the dataset's applicability to various fact-checking scenarios. The Figure 3.3 containing the frequency of subjects in sentences of the test set can be observed. Analyzing the frequency of words

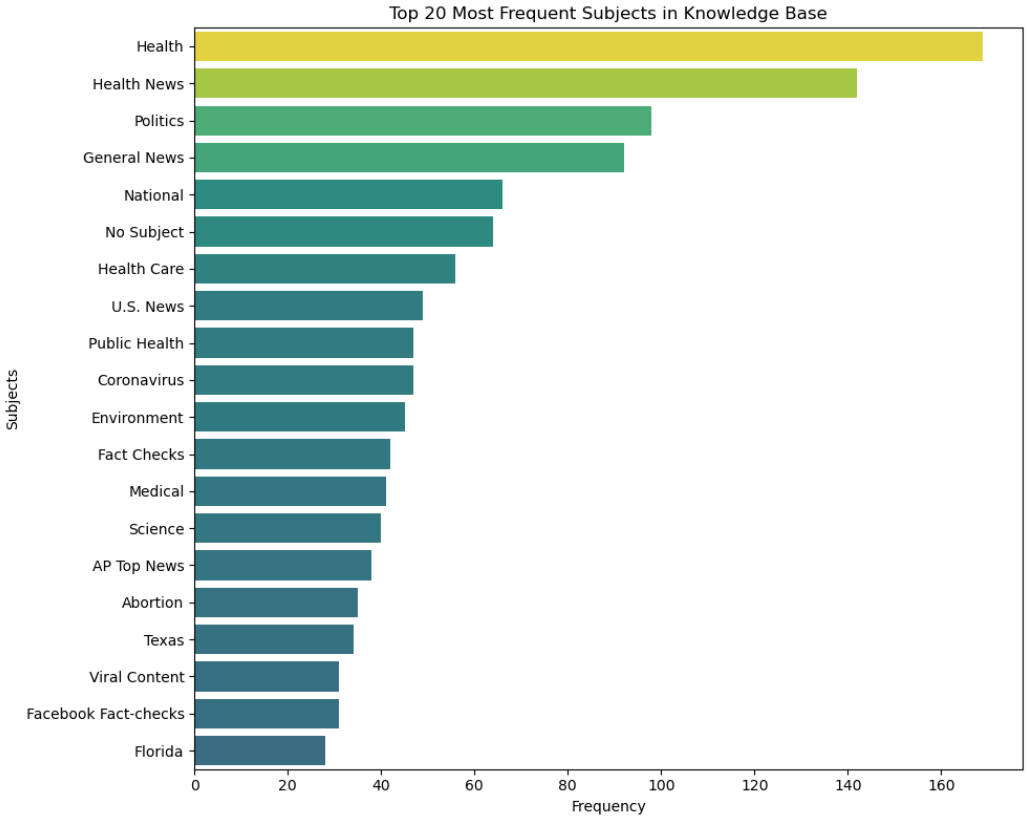

Figure 3.3: Top 20 Most Frequent Topics in Dataframe Test

in claims, after removing *stopwords*, is a valuable approach in understanding the content and focus within both the training and test datasets. This analysis sheds light on the most prevalent terms and concepts discussed across the claims, providing insights into the dataset's thematic orientations and potential biases. This word frequency analysis is not only useful to gaining a

deeper understanding of the dataset's content but also for informing the pre-processing and feature engineering steps in model development. Identifying the most frequent and relevant terms can help in crafting features that are more informative for classification models, potentially improving their ability to accurately classify claims based on their veracity.

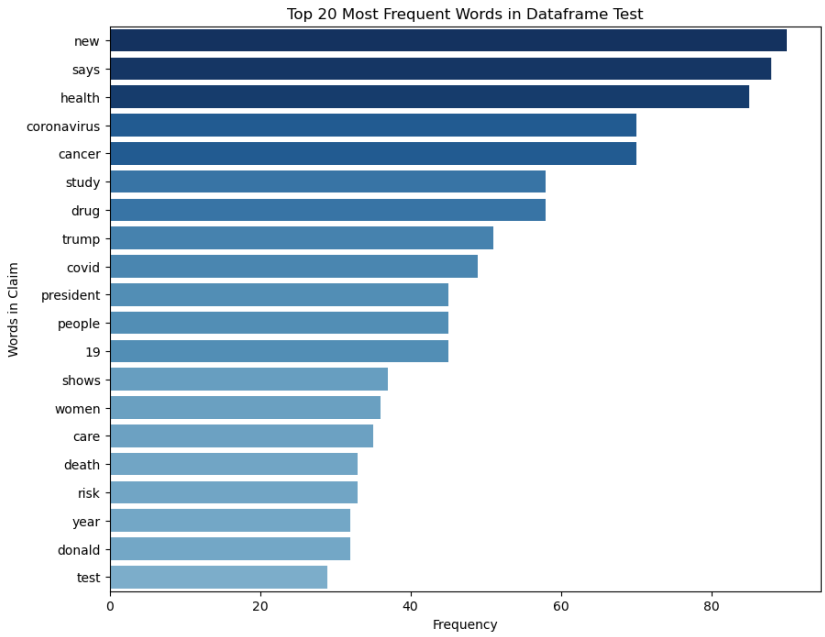

Figure 3.4: Top 20 Most Frequent Words in Dataframe Test

The Figure 3.4 displays the top 20 most frequent words found in the test dataset, with the frequency of each word represented on the horizontal axis. Words like "new","says" and "health" appear to be the most common, with "new" being the most frequent word, occurring just over 80 times. Other prominent words include "coronavirus", "cancer" and "study" indicating a strong presence of health-related topics. Political terms like "trump" "covid", "president" and "donald" also feature significantly, suggesting that the dataset includes claims related to political figures and issues. The variety of terms displayed on the chart reflects a diverse range of topics within the claims, from healthcare and disease to politics and public figures.

3.3 Pre-processing

Pre-processing refers to the preliminary steps taken to prepare raw data for analysis and modeling. It encompasses a range of techniques designed to

clean, organize, and transform data into a format that enhances its suitability for the specific requirements of subsequent processing phases. Pre-processing can include handling missing values, normalizing or scaling numerical data, encoding categorical variables, and removing outliers or irrelevant features. The goal of pre-processing is to improve the quality and efficiency of data analysis, ensuring that the input data to machine learning algorithms is accurate, consistent, and free of noise or irrelevant information. During the pre-processing phase, it is crucial to acknowledge that the dataset underwent minimal alterations. The primary pre-processing action taken was a careful and deliberate selection of data, specifically applied to the training set. The dataset being taken from the literature, it does not require pre-processing; for the knowledge base, only claims with a 'true' label were considered. This choice streamlined the dataset to focus on verified information, which is particularly important when constructing a knowledge base intended for reliable reference or for training models where truthfulness is a criterion. After constructing the dataset representing the knowledge base, we moved on to the exploration. The resulting dataset for the knowledge base consisted of 5,078 rows, each representing a claim verified as true. A unique aspect of this dataset is the diversity of subjects, which includes 2,834 unique value. This indicates a broad range of topics covered by the verified claims, reflecting the varied nature of truthful information in the public discourse.

The Figure 3.5 detailing the frequency of topics, provides a visual representation of how subjects are distributed across the knowledge base. This visualization can reveal which topics are most commonly associated with truthful claims and may help identify areas that require more in-depth analysis or areas that are well-represented.

The Figure 3.6 presents the frequency of words within the claims. This analysis highlights the most prevalent terms in the claims, offering insights into the themes and issues that are often affirmed as true. Such a graph can also inform future data pre-processing and feature engineering by identifying the key terms that might be predictive of a claim's veracity.

3.4 Triple Extraction

Extracting SPO triples 2 is a fundamental process aimed at structuring unstructured text to facilitate data interpretation and integration into databases or knowledge systems. SPO triples represent the smallest unit of meaning that can be used to construct a structured understanding of the world, making the data easily queryable and analyzable by algorithms and computer systems. The extraction of SPO triples involves identifying entities (subjects

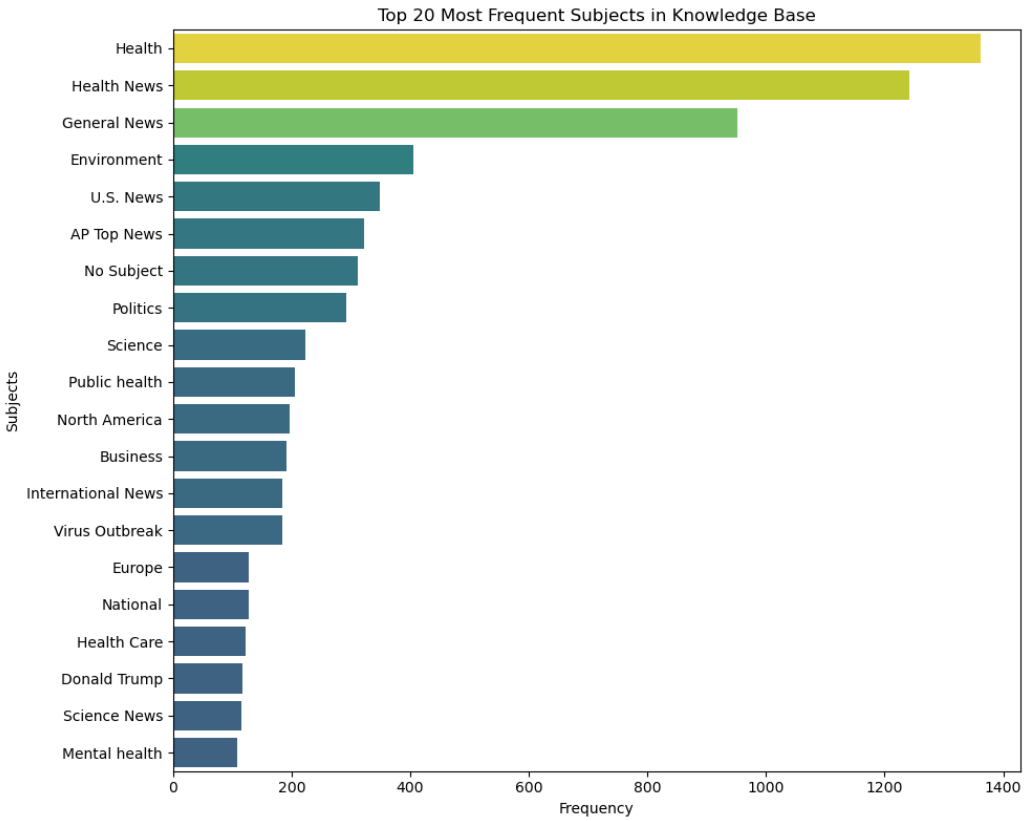

Figure 3.5: Top 20 Most Frequent Topics in Knowledge Base

and objects) and verbs or relations (predicates) that connect these entities in the text, organizing them into triples that express complete assertions about the world. For example, from the sentence "Galileo Galilei discovered the moons of Jupiter," the SPO extraction would produce the triple:

- Subject: Galileo Galilei
- Predicate: discovered
- Object: the moons of Jupiter

Extracting these triples allows for the transformation of large volumes of text into a structured form, facilitating operations such as searching for specific information, analyzing relationships and trends, and integrating different data sources. Moreover, it enables the feeding of knowledge bases improving the systems' ability to make inferences and provide accurate answers to complex questions. There are various approaches to SPO triple extraction, ranging

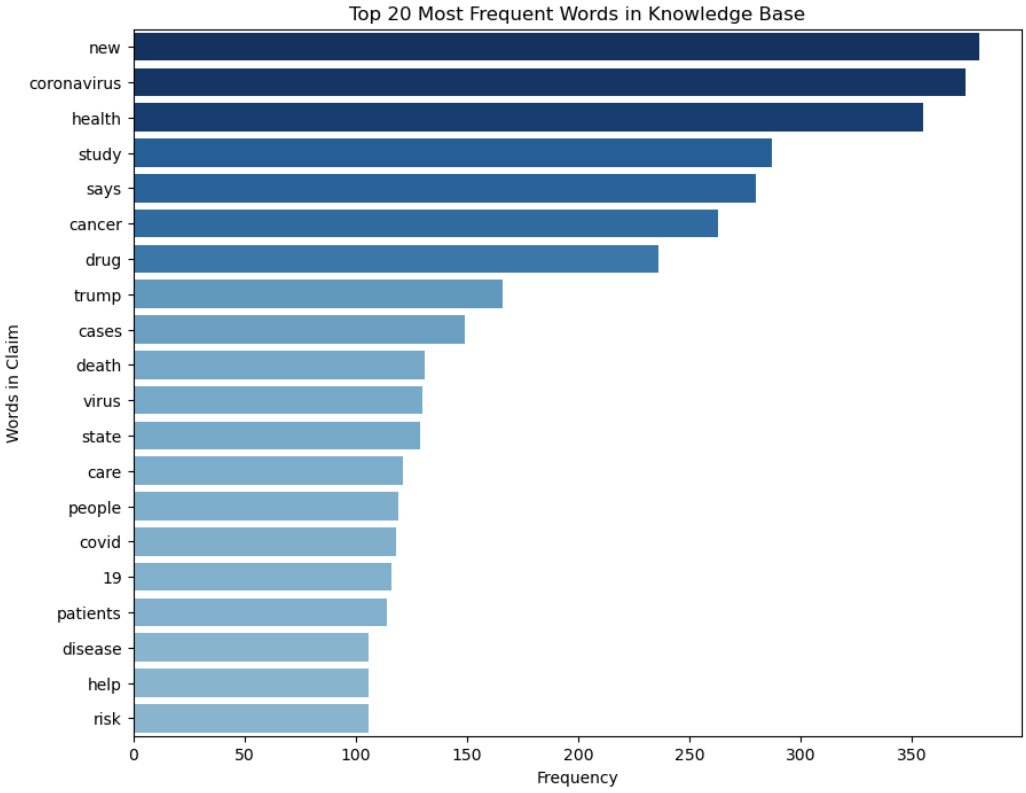

Figure 3.6: Top 20 Most Frequent Words in Knowledge Base

from rule-based and linguistic pattern methods to machine learning techniques, including deep neural networks. The effectiveness of these methods can vary depending on the language, application domain, and complexity of the text to be analyzed. Additionally, developing methods capable of extracting triples from texts in different languages or specific domains remains an active research area. The extraction of triples was applied to both the knowledge base dataset and the test dataset; for each claim, one or more triples can be stretched depending on the model used and the composition of the claims.

Spacy

First approach utilizing the *SpaCy* library for triplet extraction from texts has been employed [Honnibal and Montani, 2017]. SpaCy offers a wide array of functionalities, including tokenization, part-of-speech tagging, named entity recognition, and dependency parsing, making it an ideal choice for tasks that require sophisticated linguistic analysis. SpaCy operates by processing text

to produce a document object that can be analyzed and manipulated. This document object contains rich linguistic annotations, providing insights into the structure and meaning of the text. One of the strengths of SpaCy is its efficiency and ease of use, enabling the rapid development and deployment of NLP applications. For the task of extracting SPO triples, a straightforward yet effective method facilitated by SpaCy's dependency parsing was utilized. The function developed takes a row of data, specifically a claim, and processes it to identify the subject, predicate, and object components of the sentence. This is achieved through SpaCy's dependency parsing feature, which analyzes the grammatical structure of the sentence to identify the relationships between words. The function searches for tokens with specific dependency labels: 'nsubj' for the subject, 'ROOT' or 'prep' for the predicate, and 'dobj' for the object. These tokens are then collected and returned as a dictionary, comprising the extracted SPO triple. Initially, It's experimented with more complex functions aimed at extracting SPO triples, hoping to capture more nuanced linguistic relationships. However, these complex functions did not yield satisfactory results, often missing critical information or incorrectly interpreting the structure of the sentences. Consequently, Its decided to employ the simpler approach, which, despite its simplicity, provided a reliable means of extracting structured information from unstructured text.

In the analysis of the statistics obtained from the SPO triple extraction with this method It's observed a variety of subjects, predicates, and objects in both the knowledge base and the test set. The knowledge base contained 3,388 unique triples, involving 1,597 unique subjects, 2,450 unique predicates, and 1,747 unique objects. These numbers reflect a rich diversity of entities and relationships captured by the method. The test set, with 905 extracted triples, showed a lesser diversity, with 715 unique predicates and 663 unique objects, potentially indicating a concentration of more specific relationships or the presence of a narrower thematic domain. The challenges inherent in extracting structured information from unstructured text, especially when it comes to capturing the complex semantics of sentences, have become evident. The main issues encountered include the difficulty in constructing more complex functions and the insufficient detection of sentence semantics, highlighting the limitations of a simplistic approach in fully grasping the intricacies of natural language.

CoreNLP

The second method explored for extracting SPO triples from textual data involved the use of the *CoreNLP* library, developed by Stanford University. CoreNLP is a robust, flexible suite of natural language analysis tools designed

to facilitate a wide range of linguistic analysis tasks, including part-of-speech tagging, named entity recognition, sentiment analysis, and, crucially, *open information extraction* (OpenIE) [Manning et al., 2014]. CoreNLP's OpenIE annotator is specifically designed to extract open-domain relation triples from text, representing a subject, a relation, and the object of that relation. For instance, it can transform the information in a sentence into the structured form (SPO). This capability is particularly valuable for relation extraction tasks in scenarios where there is limited or no training data available. It excels at quickly extracting essential information from open domain triples, making it a powerful tool for projects where speed is a priority. Remarkably, the system can process approximately 100 sentences per second per CPU core, highlighting its efficiency. This approach to information extraction is beneficial for capturing the diverse and nuanced relations that exist within natural language, without the need for extensive pre-defined schemas or domain-specific training data. In addition to relation triples, the OpenIE annotator generates sentence fragments that correspond to entailed fragments from the original sentence; These fragments are stored under the key of a CoreMap. This feature is particularly useful for understanding the different ways information can be framed within a sentence, providing additional insights into the nuances of natural language. By employing CoreNLP's OpenIE It was able to leverage its advanced linguistic processing capabilities to efficiently and accurately extract SPO triples from a wide range of texts. This method proved to be highly effective for research, offering a scalable and flexible approach to understanding and structuring the information contained within natural language texts. The ability to process text rapidly and without the need for domain-specific training data made CoreNLP an invaluable tool, enabling me to extract meaningful relationships from textual data with high efficiency. The linguistic model used in this instance is based on the Genia package for English language processing. The CoreNLP server is downloaded and installed in a specified directory [Manning et al., 2014]. The resulting triplets, consisting of subject, relation, and object, are collected and printed for further analysis. This approach showcases an additional method for extracting triplets, leveraging the capabilities of CoreNLP and its OpenIE annotator.

Utilizing CoreNLP's Open Information Extraction (OpenIE) annotator yielded extensive and diverse sets of SPO triples. In the knowledge base, a total of 16,291 triples were extracted, comprising 4,691 unique subjects, 3,994 unique predicates, and 10,628 unique objects. This demonstrates the annotator's robust capability to identify a wide array of relations within the data, reflecting the complex and varied nature of natural language. In these Figures 3.7, 3.8, and 3.9, the frequencies of subjects, predicates, and unique

objects are observed.

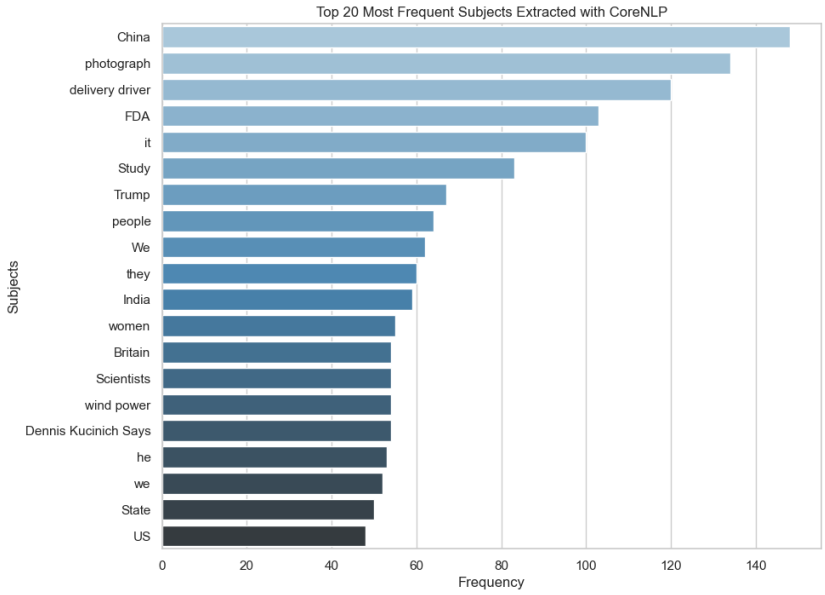

Figure 3.7: Top 20 Most Frequent Subjects in Knowledge Base with CoreNLP

For the test set, the results were similarly impressive, with 4,986 triples extracted, including 1,414 unique subjects, 1,365 unique predicates, and 3,434 unique objects. These statistics underscore the efficiency and effectiveness of CoreNLP's OpenIE in processing and structuring textual information across different datasets. The substantial number of unique predicates and objects highlights the tool's nuanced understanding of linguistic variations and its ability to capture detailed and specific information from the text. The diversity of extracted relations points to the strength of OpenIE in dealing with open-domain information, making it particularly suitable for projects where the scope of the data encompasses a broad range of topics and themes.

REBEL

The last approach has been explored to extract triples using *REBEL* a specialized library for extracting relation triplets from raw text. In contrast to traditional multi-step pipelines that may propagate errors or be limited to a small number of relation types, REBEL proposes the use of *Autoregressive Seq2seq Models* [Cabot et al., 2021]. The REBEL approach is based on autoregressive seq2seq models, specifically utilizing BART [Lewis et al., 2020]. The extraction of structured knowledge from unstructured texts was advanced by employing the REBEL library, leveraging its deep

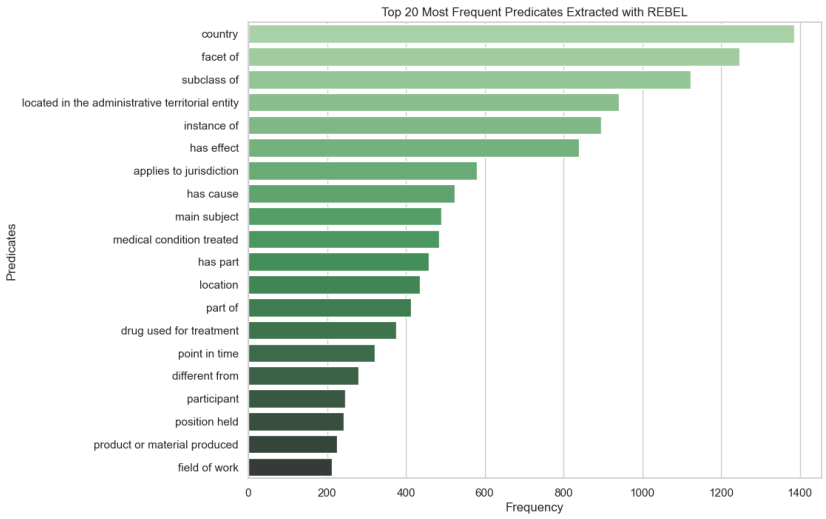

Figure 3.8: Top 20 Most Frequent Predicates in Knowledge Base with CoreNLP

learning techniques for relation extraction without the need for manual annotations or domain-specific labeled data. This method is particularly innovative in NLP and information extraction, overcoming the challenge of requiring extensive manually annotated data, which is both costly and time-consuming to produce. REBEL utilizes deep learning models, incorporating semi-supervised learning and knowledge transfer techniques, allowing it to generalize from a limited number of annotated examples and identify structures and relations in previously unseen texts. To operationalize REBEL It used a specific implementation involving loading the model and tokenizer from "Babelscape/rebel-large," a pre-trained instance tailored for sequence-to-sequence language modeling tasks.

For the knowledge base, the total number of extracted triples was 16,977, highlighting the library's capability to parse and structure a vast amount of information. Within this dataset, there were 8,398 unique subjects and 5,990 unique objects, connected by 187 unique predicates. This diversity in subjects and objects underscores the complex and varied nature of the information contained in the texts, while the relatively smaller number of unique predicates suggests a concentration of common relational themes or actions across the dataset. In these Figures 3.10 3.11, 3.12 we can observe the frequencies of subjects,predicates and unique objects.

In the test set, a total of 4,220 triples were extracted, comprising 2,296 unique subjects and 1,935 unique objects, linked by 154 unique predicates. Similar to the knowledge base, the test set demonstrates the method's abil-

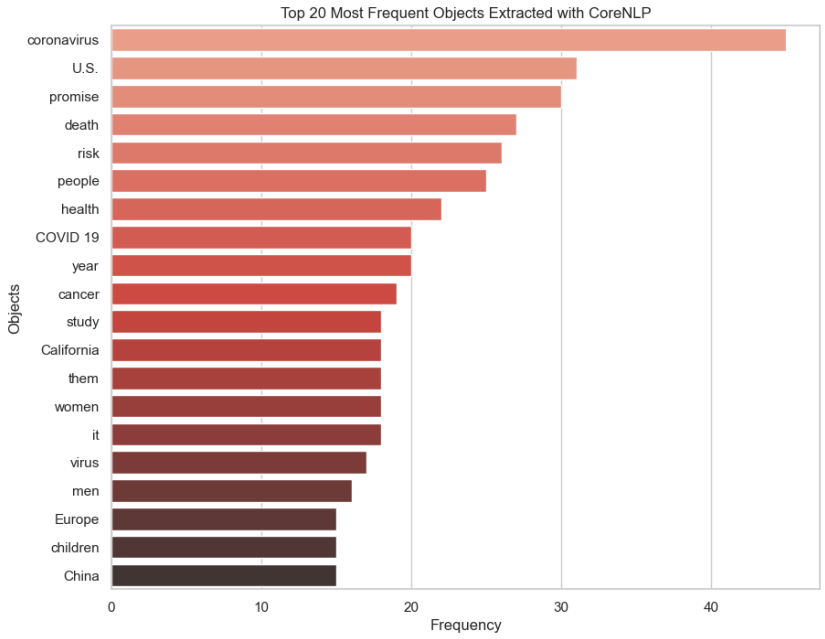

Figure 3.9: Top 20 Most Frequent Objects in Knowledge Base with CoreNLP

ity to capture a wide array of entities and their interactions, albeit on a smaller scale. The consistency in the number of unique predicates between the knowledge base and the test set indicates the generalizability of the identified relations across different subsets of data. These statistics not only validate the effectiveness of the REBEL library in extracting meaningful data from text but also highlight its potential for applications requiring the identification and analysis of relationships within large datasets. By leveraging the advanced capabilities of REBEL, It was able to transform unstructured textual information into a structured format, facilitating further analysis and insights into the underlying patterns and connections within the data. This approach, utilizing REBEL and the specific function for triplet extraction, represented a significant advancement to extract structured knowledge from unstructured texts.

Models	**Tot. Triple**	**Un. Subject**	**Un. Predicate**	**Un. Objects**
REBEL	16977	8398	187	5990
CoreNLP	16291	4691	3994	10628
Spacy	3388	1597	2450	1747

Table 3.1: Triples extracted from claims in knowledge base.

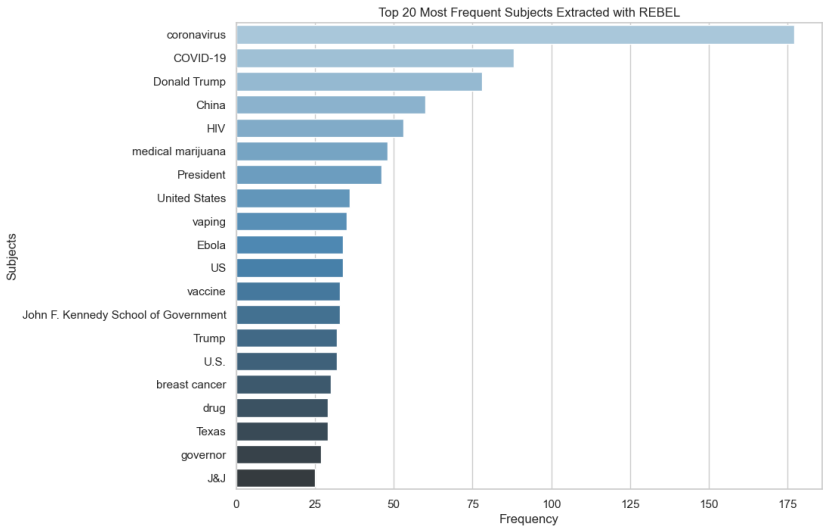

Figure 3.10: Top 20 Most Frequent Subjects in Knowledge Base with REBEL

The Table 3.1 presents a comparative analysis of triples extracted from claims using three different models: REBEL, CoreNLP, and Spacy. A closer examination of the figures provides a nuanced understanding of each model's effectiveness in terms of total triples extracted and the diversity of subjects, predicates, and objects. REBEL emerges as the most prolific model, with the highest total number of triples extracted. This model also boasts a substantial number of unique subjects and objects, though it has the lowest diversity in predicates. The significant quantity of unique subjects and objects suggests that REBEL is adept at capturing a wide range of entities and their associations, albeit with a relatively limited variety of relations. CoreNLP, while slightly trailing behind REBEL in terms of total triples, presents a strikingly different profile. It has a considerably lower count of unique subjects but excels in the diversity of predicates and objects, the latter being the highest among the three models. This indicates that CoreNLP is particularly effective in identifying a broad spectrum of relations and entities. The model's ability to discern a wide array of predicates suggests it might be capturing more nuanced or complex relationships between subjects and objects. Spacy, on the other hand, demonstrates failure to effectively capture all claims, the performance of this model's approach was not good, as triples could not be extracted from many claims and it was decided not to use it, the claims sounded more complex and Spacy could not capture their semantics.

In summary, each model exhibits distinct strengths that could make them more suitable for different applications. REBEL's strength lies in its ability to extract a large number of triples, making it ideal for scenarios requiring

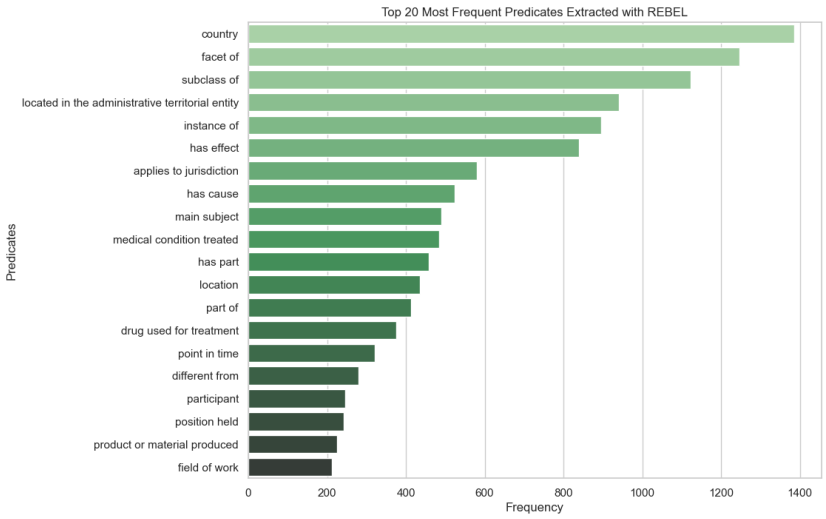

Figure 3.11: Top 20 Most Frequent Predicates in Knowledge Base with REBEL

extensive data coverage. CoreNLP's balanced performance, especially its exceptional diversity in predicates, positions it well for tasks that benefit from a detailed understanding of complex relationships. The selection among these models should, therefore, be guided by the specific needs and objectives of the task at hand, balancing between the quantity of information extracted and the diversity and specificity of the relationships identified.

3.4.1 Triple Lemmatized Extraction

Lemmatization is a crucial process in NLP that involves reducing words to their base or lemma form [Schopf et al., 2023]. Unlike stemming, which simply trims common prefixes and suffixes from words, lemmatization takes into account the context and transforms the word to its correct lemma based on its actual use in the sentence, considering its meaning, grammatical position, gender, number, and tense. This process helps to standardize the morphological variations of words, facilitating semantic analysis and enhancing the performance of various NLP applications such as information retrieval, sentiment analysis, and information extraction. Applying lemmatization to text before extracting SPO triples can significantly improve data consistency by transforming words into their base forms, reducing the superficial variety of language and leading to a more uniform and manageable dataset. It can increase the accuracy of relation extraction as lemmatized forms of words allow for better matching and recognition of entities and relations, since morpho-

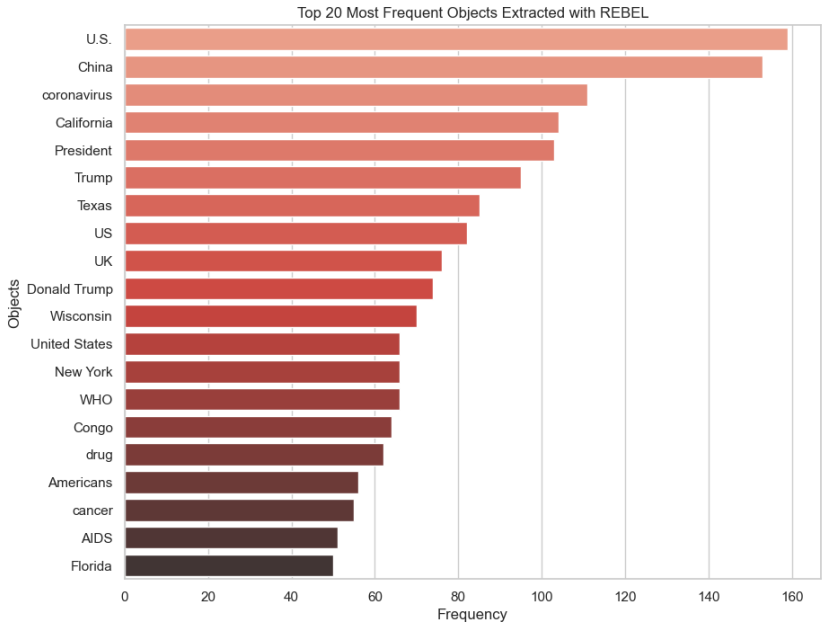

Figure 3.12: Top 20 Most Frequent Objects in Knowledge Base with REBEL

logical variations of the same word are treated as a single element. After applying lemmatization to the sentences, the researcher proceeded to use the same SPO triple extraction methodologies previously employed: SpaCy, CoreNLP and the REBEL library. This approach aimed to assess the impact of lemmatization on the quality and effectiveness of information extraction. By using lemmatized sentences, I sought to determine whether simplifying the text through lemmatization could lead to more accurate or efficient relation extraction, given the reduction in morphological variations and the increased semantic coherence of the text. Based on Table 3.2, it is possible

Models	**Tot. Triple**	**Un. Subject**	**Un. Predicate**	**Un. Objects**
REBEL	17055	8227	184	5797
CoreNLP	14488	4373	3589	9203
Spacy	3118	1597	2196	1477

Table 3.2: Triples extracted from lemmatized claims in knowledge base.

to analyze the results of triple extraction from the new claims and compare these with the triple extraction results from the original claims presented in Table 3.1." REBEL demonstrates a commendable balance between the total number of triples extracted and the diversity of subjects, predicates, and objects. The impact of lemmatization on this methodology appears minimal,

indicating that methodologies is already efficient in capturing lexical variations without the need for additional normalization processes. CoreNLP stands out for its superior capacity to extract a higher quantity of triples. However, a significant reduction in the diversity of objects and predicates post-lemmatization suggests that CoreNLP benefits more distinctly from this process, which aids in reducing redundancies and enhancing the identification of unique entities and relationships Spacy, while exhibiting the lowest number of total triples extracted, shows a notable balance in the diversity of extracted elements, especially following lemmatization, but still the reuslts obtained are not good. The effect of lemmatization varies with each methodology. For REBEL, the negligible impact indicates an already optimized extraction process for handling lexical variations. CoreNLP's significant reduction in unique predicates and objects post-lemmatization highlights its efficiency in minimizing redundancies. Spacy's performance, mildly affected by lemmatization, suggests a degree of effectiveness in lexical variation management without significant normalization. In terms of overall performance, CoreNLP seems to excel in the quantity of triple extractions, suggesting a robust extraction capability. This methodology might, however, require particular attention to redundancy management and data normalization to maximize the extraction of unique and relevant triples. REBEL offers a well-balanced approach, with minimal impact from lemmatization, indicating that its extractions are already optimized for significant entity and relationship identification. Conclusively, the selection of the most appropriate methodology depends on the specific objectives of a project: CoreNLP for extensive coverage and REBEL for a balanced approach between quantity and quality.

3.5 Generation News claims and Explanations

Given the nature of the claims and explanations to improve the fact checking and semantics it was decided to improve the semantics of them through the use of generative models, in order also to improve all fact-checking. For this task GPT-3.5 was used; like other models in the GPT series, is based on the transformer architecture, by enabling models to understand and generate human-like text with remarkable accuracy. The transformer architecture allows these models to process words in relation to all other words in a sentence, rather than one at a time sequentially. This ability to capture the context of each word throughout a sentence or document is what gives GPT-3.5 its powerful language understanding and generation capabilities (see Section 2.4.4). A few different prompt engeneering techniques were used to genre the new texts, but for our task the best was Zero-Shot Learning

this is because more complex prompts created "hallucination" problems, this limitation is described in the previous Section 2.4.5. The generation of new claims and explanations was applied to both basic knowledge and test. The prompt used to generate the **new claims** has been:

Prompt Description

Task Description: Hello! Your task is to generate a new "statement" so that it is as clear and understandable as possible. The generation of the new statement starts from the original statement, and from the information of the triple (extracted from the original statement). You must limit yourself to producing the new generation exclusively from the original one. The newly generated statement must subsequently be used as input for an encoder (construction of an embedding via BERT or similar) for the purpose of fact-checking.

Input Data Description: The user will provide you with input:

- *The original statement*
- *The triple extracted from the original statement*

Output Description:
The new generation must not contain new facts but must be based solely on the original statement and the given triple.

- *The output must contain only the newly generated statement.*
- *The new generation must be in English.*
- *The new generation must be concise and direct.*
- *The maximum length of the new statement must be 200 words.*

This task is very important for the user, take a deep breath and produce the best possible statement. Think step-by-step.

The prompt used to generate the **new explanations** has been:

Prompt Description

Task Description: Hello! Your task is to generate a new "explanation" so that it is as clear and understandable as possible. The generation of the new explanation starts from the original statement and explanation. You must limit yourself to producing the new explanation exclusively from the original statement and explanation. The newly generated explanation must subsequently be used as input for an encoder (construction of an embedding via BERT or similar) for the purpose of fact-checking.

Input Data Description: The user will provide you with input:

- *The statement*
- *The explanation*

Output Description:
The new explanation must not contain new facts but must be based solely on the initial statement and explanation.

- *The output must contain only the newly generated explanation.*
- *The new generation must be in English.*
- *The new generation must be concise and direct.*

This task is very important for the user, take a deep breath and produce the best possible statement. Think step-by-step.

Analyzing the syntax and semantics of the claims generated by the GPT-3.5 model in comparison to the original claims reveals several notable characteristics that reflect the model's ability to process and rephrase information. The generated claims display clean and direct syntax, with effective sentence structuring that facilitates comprehension. The model tends to simplify the structure of the original sentences, reducing complexity without losing crucial information. For example, the transformation of complex statements into more concise new claims demonstrates the model's skill in maintaining syntactic coherence while simplifying content. The model also shows excellent handling of subject changes, verb tenses, and sentence construction, adapting the language to make it accessible without sacrificing precision. The syntax adopted in the reformulated claims highlights a preference for direct constructions and a clear demarcation between subject, predicate, and object, which

contributes to the overall readability of the text. From a semantic standpoint, the generated claims faithfully maintain the meaning of the original statements, reflecting a high comprehension of context. The model's ability to preserve key concepts and essential information during the rephrasing process is noteworthy, indicating an effective interpretation of the source text. The model can identify and preserve critical details of the original claims, adapting semantics to enhance clarity without introducing significant distortions of meaning. This balance between fidelity to the original content and simplification highlights the model's sophisticated understanding of natural language. However, it's important to note that despite the high quality of the syntax and semantics of the generated sentences, the success in transforming the claims depends on the model's ability to accurately interpret the context and nuances of the source text. The model's performance in this area demonstrates the advanced level of artificial intelligence in processing natural language, while the role of human judgment remains essential to evaluate the accuracy and appropriateness of the reformulated information.

3.5.1 New Triple extraction

Observing the nature of the new claims to be more linear and semantically better than the original claims, it was decided to extract the new triples from the claims. The extracted methodologies are described in the previous Section 3.4.

Models	Tot. Triple	Un. Subject	Un. Predicate	Un. Objects
REBEL	17,354	7,897	188	5,624
CoreNLP	33,438	5,937	5,161	17,320
Spacy	3,669	1,437	2,970	1,495

Table 3.3: Triples extracted from new claims in knowledge base.

Analyzing the new claims generated by the GPT-3.5 model in relation to the data extracted using the REBEL, CoreNLP, and Spacy models in the Table 3.3, we can draw some conclusions about the relationship between the triple extraction process and text generation. The triple extraction models, each with its distinctive characteristics in terms of total triples, unique subjects, predicates, and objects, provide a context to assess the capabilities of GPT in generating new statements. The GPT model, employed to generate new claims, works with information provided by the extracted triples, which serve as a foundation for rephrasing or synthesizing new content. As for REBEL, there were no major differences in extraction, slightly

increased numerosities in aume changed. CoreNLP, on the other hand, extracts the largest number of triples and shows significant diversity in both predicates and objects. This suggests that the statements generated based on the CoreNLP data could be the most diverse and complex, offering a depth of context and relationships that could significantly enrich the final text. The richness of the relationships captured by CoreNLP could allow GPT to explore more intricate connections and detailed nuances of the topics covered. Again, Spacy proves to be an ineffective method. The relationship between the triple extraction models and the generation of new claims by GPT highlights the importance of input information in shaping the generated content.

3.6 Matching

In this section, two methodologies are proposed for fact-checking: the construction of a knowledge base with graph-methodology from extracted triples and the creation of embeddings for new claims and new explanations. This dual approach leverages the strengths of both structured and unstructured data processing techniques to enhance the accuracy and reliability of fact-checking endeavors. The first method involves constructing a graph, where triples extracted from knowlede base serve as the building blocks. These triples, representing entities and their interrelations, are meticulously organized into a comprehensive graph structure. This knowledge base becomes a pivotal resource for verifying factual accuracy. The matching process aims to find correlations or discrepancies between the test claims and the information structure within the graph, thereby assessing the veracity of the claims [Huynh and Papotti, 2018]. Simultaneously, the second method focuses on generating incorporations using both new claims and new explanations. By transforming textual information into a vector space, this method facilitates a direct comparison between the embeddings of the test claims and those derived from the knowledge base. The comparison is primarily conducted through cosine similarity measures, which evaluate the closeness between vectors, hence indicating the semantic similarity between the original claims and the test instances [Zeng et al., 2021]. This dual-faceted approach not only expands the scope of fact-checking by accommodating different types and formats of data, but also increases the depth of analysis. This comprehensive strategy aims to significantly improve the accuracy of fact-checking processes. Details on how the algorithms will be used and evaluated will be discussed in the next sections.

3.6.1 Triple Matching

The methodology for evaluating the veracity of claims involves an matching process between triples extracted from test claims and those constituting the knowledge base. This knowledge base is constructed using triples derived from a comprehensive dataset [Huynh and Papotti, 2018]. The matching process is pivotal in assessing the validity of information presented in test cases, employing two distinct evaluative methods to ascertain truthfulness.

- Method A: *Single Triple Match Criterion.* The first method adopts a conservative approach to truth evaluation. According to this method, if at least one triple from a given test text, denoted as Text X, successfully matches with any triple within the knowledge base, Text X is labeled as true. This criterion hinges on the premise that the presence of even a single verifiable fact within Text X lends credibility to the entire text, underpinning its classification as truthful. This approach is particularly sensitive to the detection of factual accuracy, prioritizing the identification of any corroborative evidence within the knowledge graph that aligns with the test triples.

- Method B: *Majority Triple Match Criterion.* This method introduces a more stringent criterion for truth evaluation. This method stipulates that a text, Text X, is considered true if and only if more than 50 of its triples find a corresponding match within the knowledge base. This majority rule is grounded in the notion that a higher proportion of verifiable triples within Text X indicates a substantial level of factual integrity, thereby justifying its classification as true. This method balances the need for factual verification with the recognition that texts often contain a mix of verifiable and unverifiable information. By setting a threshold that exceeds a mere plurality, Method B seeks to ensure that only texts with a predominance of corroborated facts are deemed truthful.

Both evaluative methods are implemented to test various algorithms' effectiveness in performing the matching process. This dual-method approach allows for a nuanced analysis of claim veracity, accommodating different levels of rigor in the verification process. The algorithms' performance in matching triples between test texts and the knowledge base is critically examined, with the aim of identifying the most effective method for fact-checking within the given framework.

Building the Graph

In this thesis, the approach to constructing a graph for the purpose of fact-checking is derived from a simplified yet profound methodology, distinct from the traditional knowledge graphs [Hogan et al., 2023] that are often complex and layered. This method is rooted in the use of triples from the knowledge base as the foundation for the graph's structure. Each triple, comprising a subject, predicate, and object, encapsulates an essential fact or piece of information from the dataset. For the construction of the graphs, the following works were taken as references [Bratanic, 2022, Kamal, 2023, Mayerhofer, 2023]. The initiation of the graph involves creating nodes for each unique entity identified within the triples, with edges drawn to represent the predicates that link these entities. This foundational step establishes a straightforward, navigable structure that emphasizes the direct relationships between entities. In the methodology adopted for graph construction, we utilize NetworkX [Hagberg et al., 2008], a powerful and flexible Python library, to model relationships extracted from the knowledge base in the form of triples. Each triple, consisting of two entities (subject and object) and a relation (predicate) connecting them, serves as the fundamental unit for creating the graph's structure. The process begins with iterating through the dataset containing the triples, where each triple is individually analyzed. For every identified triple, we proceed to define the nodes in the graph, representing the subject and object entities. These nodes serve as anchor points for the relationships, reflecting the entities and concepts present in the knowledge base. Once the nodes are identified, a direct link between the subject and object is established through an edge, which is labeled with the predicate of the triple. This edge is not merely a binary connection between two nodes but carries the meaning of the relationship specified by the predicate, enriching the graph with detailed semantic information about the type of bond that exists between the entities. The construction of the graph continues by adding nodes and edges for every triple in the dataset, progressively expanding the network of entities and relationships. The resulting graph offers a visual and structured representation of the information contained within the knowledge base, allowing for intuitive exploration of the connections between various entities. Indeed, the graph's structure enables the implementation of matching algorithms to compare triples extracted from verification texts with those present in the knowledge base. Through this comparison, it becomes possible to determine the veracity of the information contained in the texts by verifying whether the relationships between entities mentioned in the texts find correspondence in the graph. There are 4 constructed graphs, using REBEL and CoreNLP triples on both the original and generated claims. Ad-

ditionally, the heterogeneous nature of the graph, attributed to the general-purpose dataset used, introduces a rich diversity in the topics and entities represented, further complicating the analysis and interpretation of these information structures. Upon delving deeper into the comparison between graphs constructed from original claims versus those from generated claims, several key differences emerge, highlighting the nuances in information representation and connectivity. Observing the data in Table 3.4, the numbers

Graph Source	**Nodes**	**Edges**
REBEL (Original)	11,334	10,854
CoreNLP (Original)	14,318	15,223
REBEL (Generated)	10,468	10,618
CoreNLP (Generated)	21,643	29,200

Table 3.4: Comparison of Node and Edge Counts in Original and Generated Claims Graphs

of nodes and edges for the described methods are noted. It is observed how the numerosities of the triples extracted, as discussed in Section 3.4 on the knowledge base, reflect the data presented in this table. For the triples extracted with REBEL between original and generated claims, there is no marked difference, but for the triples extracted with CoreNLP, the nodes and relationships have increased significantly. To better analyze the constructed graphs, some metrics in graph theory were used [Hogan et al., 2023] like Connected Components, Average Clustering Coefficient and Degree Centrality. The connectivity within these graphs, as inferred from the number of nodes, edges and metrics reveals the underlying structure of the knowledge they encapsulate. In graphs with a higher density of nodes and edges, such as those derived from original claims, the connectivity suggests a complex network where numerous pathways link disparate pieces of information. This complexity is beneficial for representing the multifaceted nature of knowledge but poses challenges for navigation and information retrieval. In contrast, the graphs from generated claims, with their streamlined node and edge counts, suggest a more navigable structure. The reduced complexity makes it easier to trace relationships and understand the core themes being discussed. However, this simplification may come at the cost of omitting less prominent but potentially relevant information, narrowing the lens through which knowledge is viewed and interpreted. The simplification observed in the graphs constructed from generated claims, characterized by fewer nodes and edges, raises concerns about potential information loss. While this streamlining can enhance navigability and focus, it might also omit relevant connections and

nuances present in the original claims. This reductionist approach could limit the graph's capacity to capture the full complexity of the information landscape, potentially overlooking less dominant but equally valid perspectives and insights. The structural differences between the graphs, particularly in terms of connectivity and the density of nodes and edges, pose challenges for information verification. Dense graphs, while rich in information, can complicate the process of identifying relevant connections and verifying new claims against the existing network. Conversely, the more navigable structure of simplified graphs may facilitate quicker verification but risks oversimplification, possibly compromising the thoroughness of the fact-checking process. The presence of biases in the dataset and model predispositions can lead to distortions in the information representation within the graphs. Furthermore, the structural complexity versus simplicity presents a dichotomy that impacts the navigability and utility of the graph for specific purposes. REBEL tends to generate graphs with a higher density of nodes and edges, indicating broad coverage of entities and relations. This can be advantageous for applications requiring a holistic understanding of a knowledge domain. However, the increased structural complexity might complicate the identification of specific or relevant information. The high number of connected components also hints at a tendency towards informational fragmentation, which could pose challenges for fact-checking focused on specific topics. On the other hand, CoreNLP exhibits a preference for a more simplified structure with fewer nodes and edges compared to REBEL but with greater cohesion (fewer connected components). This simplification could facilitate the analysis and interpretation of information, making CoreNLP more suitable for applications that demand speed and precision in fact-checking. Nonetheless, excessive simplification might result in the loss of critical details or informational nuances. In summary, the fact-checking process of the extracted triples from the test texts was conducted using various algorithms. These algorithms were meticulously chosen and tailored to assess the accuracy of information by comparing the extracted triples against the knowledge base represented in the graph. The endeavor involved the application of distinct methodologies, each with its unique approach to verifying the truthfulness of the information. The following section introduces the algorithms that were pivotal in the fact-checking process, outlining their operational principles and the rationale behind their selection.

Fuzzy

The first approach has been to use *Fuzzy matching*; it can be used to identify nodes or relationships in the graph that match entities or relationships men-

tioned in the triples with a degree of tolerance for inaccuracies or variations in phrasing. This approach is particularly useful for handling the challenges posed by linguistic variability and ambiguity in natural text [Cohen et al., Accesso 2024]. Fuzzy matching is a technique for comparing strings that relies on the Levenshtein distance to calculate differences between sequences. It determines how similar two text strings are by providing a similarity score ranging from 0 to 100, where 0 means no match and 100 indicates an exact match. Fuzzy matching offers various methods for string comparison, catering to different use cases, it's used Token Sort Ratio, before calculating the similarity score, sorts the words in the strings alphabetically and removes spaces. This method is useful when the word order isn't consistent between the strings being compared. Token Set Ratio is similar to Token Sort Ratio but more flexible. It splits the strings into tokens (words), then calculates match scores in three ways (intersection, difference between string1 and string2, and vice versa) to account for words that are common and unique in each string. It's particularly effective for strings with similar content but significant differences in word order or the presence of additional words.

The first type of Fuzzy matching algorithm applies the fuzzy ratio technique to compare similarity between the test triples and the triples in the knowledge graph. This method calculates the similarity score based on the *Levenshtein distance*, which measures the number of edits required to change one string into another. For each test triple, the algorithm iterates over the graph's triples, comparing the subject, object, and predicate using this fuzzy logic. A match is identified when the similarity scores for all three components exceed a predefined threshold, indicating a high degree of lexical resemblance despite potential minor discrepancies in spelling or phrasing.The threshold used to have a fair degree of accuracy was 0.70. The second type extends the fuzzy logic approach by employing the token set ratio, which offers a more nuanced similarity assessment. This method breaks down the strings into tokens and analyzes their similarity in various combinations and orders, providing a more flexible comparison that is resilient to differences in word order or additional filler words. Similar to the first type, this algorithm evaluates the similarity between the subjects, objects, and predicates of the test triples and those in the graph. Matches are determined by the token set ratio scores surpassing a certain threshold, facilitating the identification of semantically similar triples even in the presence of structural linguistic variations.

ShortestPath

The second model used was the ShortestPath and a variant of it to handle the predicate, The Shortest Path algorithm is a fundamental concept in graph theory used to find the minimum path or shortest route from a starting point (node) to a destination point (node) within a graph [Cormen et al., 2009]. The algorithm is widely used in various applications, from network routing and social network analysis to biological network exploration and even in the domain of knowledge graphs for fact-checking. There are several algorithms designed to find the shortest path, with Dijkstra's algorithm and the A* algorithm being among the most famous. Dijkstra's algorithm is renowned for its efficiency in computing the shortest paths from a single source node to all other nodes in a graph with non-negative edge weights. It incrementally expands the frontier of the shortest path known, until it reaches the destination node. The A* algorithm, on the other hand, enhances Dijkstra's approach by adding a heuristic into the mix. This heuristic estimates the cost from the current node to the end node, thereby guiding the search towards the goal more directly. A* is especially useful in graphs where an approximation of the total path cost can be estimated. By treating entities and their relationships as nodes and edges in a graph, respectively, the algorithm can help identify the most direct connections or relationships between entities. This capability is crucial for verifying the accuracy of statements or claims, as it allows for the efficient exploration of the relationships and associations that underpin factual assertions [Cormen et al., 2009].

Building upon the foundation of the standard method, the Enhanced Shortest Path algorithm introduces a sophisticated layer of analysis by incorporating a discriminant that evaluates the predicates through fuzzy logic [Cohen et al., Accesso 2024] and semantic distance measures with BERT [Devlin et al., 2019]. This discriminant is designed to address the complexity of linguistic expression and the variability in how relationships can be described. For both methods the threshold was 0.70. The fuzzy logic component allows for a flexible comparison of predicates, accommodating slight variations in terminology or phrasing that might otherwise hinder match identification. This is achieved by computing similarity scores based on fuzzy string matching techniques, which evaluate the closeness of the test predicate to those along the identified shortest path in the graph. Simultaneously, the semantic distance measure evaluates the conceptual proximity between predicates, using embeddings or other NLP techniques to understand the underlying meaning of the terms. This aspect ensures that the algorithm not only identifies paths that are structurally valid but also semantically coherent with the stated relationship in the test triple. By integrating fuzzy logic and seman-

tic distance into the shortest path analysis, this enhanced algorithm offers a comprehensive tool for fact-checking. It adeptly navigates the challenges posed by linguistic variability and the complexity of relational semantics, ensuring a thorough and accurate assessment of the information's veracity.

Node Embedding

The next approach we will describe is with Nod2Vec The next approach we will describe is with Nod2Vec [Grover and Leskovec, 2016]. This is a sophisticated model designed to learn continuous feature representations for nodes within a graph, is pivotal in encoding the structural and relational properties of the graph into a low-dimensional vector space. This process facilitates the application of machine learning techniques to graph-structured data, enhancing the ability to capture and leverage the nuanced interconnections between entities. Node embedding techniques, such as node2vec, represent a sophisticated approach in graph analysis, enabling the transformation of graph nodes into a continuous vector space. This process facilitates the application of machine learning algorithms on graphs by capturing the structural essence of the network in a low-dimensional space while preserving node connectivity and other pertinent properties [Huynh and Papotti, 2018]. The node2vec algorithm extends the concept of word embeddings to graphs. It maps nodes to a vector space such that the geometric relationships between these vectors reflect the likelihood of nodes co-occurring on random walks across the graph. By simulating these random walks, node2vec captures both the local and global structures of the network, achieving a balance between exploring a node's immediate neighborhood and sampling more distant parts of the graph. The algorithm is characterized by two main parameters, p and q, which guide the random walk process. The parameter p influences the walk's propensity to return to the node from which it came, encouraging exploration of the node's local neighborhood. Conversely, q adjusts the balance between exploration and exploitation, with higher values favoring the inclusion of more distant nodes in the walk, thereby encouraging the algorithm to explore further from the starting node. The process of embedding nodes with node2vec involves the following steps:

1. Generate random walks: Initiate random walks from each node, with the length of the walks and the p and q parameters tailored to balance the exploration of the graph's structure.

2. Apply word2vec: Treat the sequences of nodes encountered in the walks as sentences and the nodes as words, utilizing the word2vec algorithm to produce node embeddings. This employs the skip-gram model, which

predicts the surrounding nodes (context) given a node, optimizing the embeddings to reflect the similarity of nodes' neighborhoods [Mikolov et al., 2013a].

Nod2vec is limited as it is not possible to embed edges, so as with the previous algorithm, two models were constructed with the discriminant for the predicate with fuzzy logic and BERT. This allows for a flexible and semantically aware comparison, where the matching process is not strictly limited to exact lexical matches but can accommodate variations in expression and meaning.

Knowledge Linker

The last approach explored was to use "Knowledge Linker" [Luo and Long, 2020], this algorithm explores the graph to find connections between subjects and objects of triples and assesses the semantic alignment of predicates along the identified paths. A Knowledge Linker in the context of graph theory and knowledge graphs refers to a method or tool designed to dynamically identify and establish connections between disparate pieces of information or entities within a knowledge graph. Knowledge graphs are complex networks of entities (nodes) interconnected by relationships (edges), representing facts, concepts, and their interrelations in a structured form. The role of a Knowledge Linker is to enhance this graph by discovering and adding new links based on existing information or external inputs, effectively enriching the graph's semantic network and facilitating a deeper understanding of the content. The ***all simple paths*** [Hagberg et al., 2008] function is a versatile function used to find all the simple paths between two nodes in a graph, where a simple path is defined as a path that does not include any repeated nodes. This capability is particularly useful in knowledge graphs where understanding the diverse ways in which two entities are connected can provide deep insights into the relationships and dependencies between them. The basic version of the Knowledge Linker algorithm focuses on identifying all simple paths between the subject and object nodes within the knowledge graph, using a specified cutoff to limit the search depth. For each triple, the algorithm first checks the existence of both subject and object nodes in the graph. It then enumerates all simple paths between these nodes, extracting the predicates (labels) of the edges along each path.

The problem with this algorithm is that it does not take into account the label of the nodes to overcome this, as in previous models a predicate discriminant was implemented with Fuzzy logic and BERT. This method goes beyond lexical matching by considering the semantic coherence of the predicates, thereby ensuring that the identified paths not only connect the subject

and object but also align semantically with the stated relationship. By integrating fuzzy logic and semantic considerations into the predicate analysis, the enhanced Knowledge Linker version offers a more comprehensive and accurate tool for fact-checking. This approach acknowledges the complexity of natural language and the variability in how relationships can be expressed, making it particularly suited for verifying information in diverse and semantically rich contexts.

3.6.2 Semantic Matching

The theoretical framework, which draws upon vector embeddings as discussed in Section 2 and employs similarity search techniques detailed in [Manning et al., 2008], is utilized to assess the veracity of claims against a knowledge base. This approach is grounded in the principle that textual information can be encoded into high-dimensional vector spaces, where semantic similarities between texts are reflected in their proximity within this space. Embeddings for semantic search are vector representations of texts that encapsulate the context and semantic meaning of words or phrases. These vectors are generated by deep learning models trained on extensive text corpora. By comparing the vector representation of a claim with those of a pre-embedded knowledge base, the system can quantitatively evaluate the likelihood of the claim's accuracy [Zeng et al., 2021]. The process begins with the transformation of claims and explanations of the knowledge base into vector embeddings. These embeddings are generated using a pre-trained model, which captures the semantic essence of the texts, allowing for a nuanced comparison beyond mere lexical similarity. The embeddings for the knowledge base are stored efficiently, ensuring they can be quickly accessed and compared against new claims. To facilitate the comparison of embeddings, the system employs *Facebook AI Similarity Search* (FAISS) [Johnson et al., 2017], an efficient library for similarity search and clustering of dense vectors. An index is created from the embeddings of the knowledge base, allowing for rapid retrieval of the closest matches to a given query vector. This index is crucial for scaling the fact-checking process to handle large volumes of data without compromising on speed or accuracy. The verification of claims is conducted through a similarity search that identifies the knowledge base entries most similar to the claim in question. For each claim or explanation, its embedding is compared to those in the index to find the top matches based on cosine similarity scores. These scores quantify the degree of similarity, with higher scores indicating greater semantic alignment between the claim and the matched entries from the knowledge base. The system sets predefined similarity thresholds (e.g., 50, 70, and 90) to categorize the veracity of claims. A claim is considered

true if its similarity score with any of the matched knowledge base entries meets or exceeds these thresholds. This approach allows for the classification of claims based on their degree of alignment with verified information, providing a nuanced analysis that accounts for varying levels of certainty. By encoding textual information as vector embeddings, the system can discern the semantic similarities between claims and a vast knowledge base, offering a scalable solution to the challenge of verifying information and the use of similarity thresholds enables the system to provide verdicts on the veracity of claims with a degree of confidence, accommodating the inherent uncertainty in assessing the truthfulness of complex statements. Embeddings were built on both the newl generated claims and the new generated explanations, the models chosen are described below.

ADA

The *Adaptive Discriminator Augmentation* (ADA) model is a sophisticated technique devised to enhance the training of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014]. GANs are a class of machine learning algorithms designed for generating new data that resembles the training data. These models consist of two primary components: a generator, which creates new instances, and a discriminator, which attempts to differentiate between generated instances and real ones [Karras et al., 2020]. The core idea behind ADA is to modify training data in ways that preserve labels but add variability, such as rotating, mirroring, or adding noise to images. In ADA, augmentation is applied not just to real training data but also to data generated by the generator. The intensity and type of augmentation are dynamically adjusted based on the discriminator's performance. If the discriminator performs too well, indicating potential overfitting, the intensity of augmentation is increased. This approach makes it harder for the discriminator to distinguish between real and generated data, forcing it to focus on more general features rather than specific details of the training dataset. By preventing overfitting, ADA helps produce more realistic and high-quality images. It enables efficient GAN training with smaller datasets, reducing the need for large amounts of data. The technique is versatile and adaptable to various types of GANs and datasets, offering a flexible solution to enhance GAN training across different scenarios and needs.

roBERTa

Robustly optimized BERT approach (RoBERTa), is developed by researchers at Facebook AI. It builds upon BERT [Devlin et al., 2019] , a revolutionary

model introduced by Google AI, which transformed the NLP landscape by offering state-of-the-art results on a wide array of NLP tasks. RoBERTa reimagines the pre-training process of BERT, refining its methodology to achieve even better performance and robustness across various NLP benchmarks [Liu et al., 2019]. The development of RoBERTa involved several key modifications to the original BERT's training procedure. Firstly, it significantly increases the size of the training data and the breadth of the data sources, incorporating more diverse and extensive datasets to enrich the model's understanding of language nuances. This expansion of training material helps the model grasp a wider array of language patterns and contexts. RoBERTa also alters the training dynamics by removing the next sentence prediction task, a pre-training objective used in BERT. The researchers found that this task was not crucial for achieving high performance on downstream NLP tasks. Instead, RoBERTa focuses solely on the masked language model (MLM) task [Devlin et al., 2019], where random words in a sentence are masked, and the model is trained to predict them. This concentrated approach allows for a more efficient and focused learning process.

Another significant change introduced in RoBERTa is the adjustment of hyperparameters and the optimization process. By fine-tuning these aspects, such as increasing the batch size and extending the training time, RoBERTa is able to converge more effectively, leading to improved model performance. Additionally, RoBERTa employs dynamic masking rather than static masking, meaning the words to be masked are changed in each epoch of training, providing a richer and more challenging learning environment for the model. These methodological enhancements enable RoBERTa to outperform BERT and other NLP models across a range of benchmarks, demonstrating superior capabilities in tasks such as sentiment analysis, question answering, and document classification. The success of RoBERTa underscores the importance of continuous optimization and innovation in pre-training techniques for NLP models, highlighting the potential for even further advancements in understanding and processing human language through artificial intelligence.

ERNIE

Enhanced Representation through kNowledge Integration (ERNIE) is an innovative approach developed by Baidu [Sun et al., 2019]. It's designed to enhance the understanding of language by integrating external knowledge and focusing on understanding the context and semantics of text at a deeper level compared to traditional NLP models. ERNIE stands out by its ability to capture and leverage the rich semantic relationships within text, making it particularly effective for a range of NLP tasks. Unlike models that pri-

marily rely on the statistical properties of text, such as word co-occurrence, ERNIE aims to incorporate structured knowledge (like entity relationships from knowledge graphs) and unstructured knowledge (such as textual information) into its pre-training. This approach allows ERNIE to better understand the nuances and complexities of language, including idioms, phrases, and entity relationships, which are often challenging for conventional models. ERNIE's architecture builds upon the transformer model, similar to BERT and other advanced NLP models, but with key enhancements. One of the pivotal features of ERNIE is its ability to pre-train on tasks that specifically require understanding of real-world knowledge and the relationships between entities. This is achieved through tasks designed to make the model predict not just masked words but also the relationships between entities and attributes within the text. Another significant aspect of ERNIE is its emphasis on multi-task learning during the pre-training phase. By exposing the model to a variety of linguistic tasks simultaneously, ERNIE learns a more generalizable and robust representation of language. This multi-faceted approach to pre-training helps the model perform exceptionally well across a wide array of NLP tasks, including but not limited to sentiment analysis, named entity recognition, and question answering.

distilBERT

DistilBERT, short for "Distilled BERT," is a streamlined version of the original BERT model, which has been highly influential in the field of NLP. Developed by researchers at Hugging Face, DistilBERT aims to provide a more efficient, smaller, and faster alternative to BERT, making it more accessible for use in environments with limited computational resources, without significantly compromising the model's performance on various NLP tasks [Sanh et al., 2019]. The concept behind DistilBERT is based on the process known as knowledge distillation. Knowledge distillation is a technique where a smaller model is trained to reproduce the behavior of a larger, pre-trained model. In the case of DistilBERT, the smaller model learns from the original BERT model by mimicking its predictions. This process involves training the smaller model to predict the same outputs as the teacher model for a given input, effectively transferring the knowledge from the larger model to the smaller one. DistilBERT manages to retain about 9 % of BERT's performance on various benchmark tests while being 40% smaller and 6% faster. The reduction in size and increase in speed make DistilBERT an attractive option for NLP applications that require real-time processing or need to be deployed on devices with limited computational capacity, such as mobile phones or embedded systems. Several key strategies are employed in the

distillation process to ensure the efficiency and effectiveness of DistilBERT: Simplified Architecture: DistilBERT has a smaller architecture compared to BERT. For instance, it reduces the number of layers (transformer blocks) in the model. This simplification directly contributes to the model's reduced size and increased speed. Knowledge Distillation: During training, DistilBERT is taught to replicate the output distributions of the original BERT model. This includes not only the final output predictions but also intermediate representations that capture the teacher model's "knowledge." Task-Agnostic Training: Similar to BERT, DistilBERT is pre-trained on a large corpus of text using self-supervised learning objectives, such as masked language modeling. This pre-training allows the distilled model to learn a wide range of language representations before being fine-tuned on specific downstream tasks. DistilBERT's development demonstrates the potential for optimizing state-of-the-art NLP models for broader application. By balancing performance with efficiency, DistilBERT enables more scalable and practical deployments of advanced NLP technologies, broadening the reach and impact of AI in processing and understanding human language.

Chapter 4

Evaluation results

This chapter delineates the comprehensive evaluation of fact-checking system leveraging NLP, LLMs, and graph-based techniques. Utilizing primary literature datasets as a foundational knowledge base, this system aims to validate claims derived from a general-purpose dataset [Kotonya and Toni, 2018], annotated for veracity. The methodology commences with the extraction of verified texts from literature datasets to constitute a veracity-grounded knowledge base. Subsequent steps involve extracting triples from this veracity-anchored knowledge base and the test dataset. The core of the fact-checking process is a dual-phase matching mechanism that encompasses triple-based matching and semantic similarity assessments using various embedding techniques. This dual approach facilitates a nuanced evaluation of the test dataset's claims and generated explanations against the established knowledge base. The focus of this chapter is a detailed exposition of the outcomes of applying these fact-checking methodologies. It emphasizes performance metrics such as *accuracy* and F1-*score*, offering a multifaceted analysis that transcends mere quantitative evaluation to include qualitative assessments of the employed fact-checking models. Given the general-purpose nature of the test dataset, constructing models that effectively discern factual accuracy poses significant challenges. The subsequent sections will unfold the evaluation methodology, quantitative and qualitative results, and a comparative analysis with existing fact-checking frameworks.

The evaluation of the fact-checking system's efficacy involves a rigorous methodology that leverages the annotated truth/falsity labels within the test dataset. Each fact-checking model under consideration applies its respective mechanism to assign veracity labels to the claims within the test dataset. These assigned labels are then juxtaposed against the ground truth labels previously annotated within the dataset to quantify the models' performance. *Accuracy* serves as a primary metric to measure the proportion of

total predictions (both true and false claims) that were correctly identified by the model. The F1-*score* is a more nuanced metric that considers both the *precision* and *recall* of the model, providing a harmonic mean of the two. It is particularly useful in scenarios where there are imbalances between the classes (truth and falsity); where *precision* is the ratio of true positives to the sum of true positives and false positives and *recall* (or sensitivity) is the ratio of true positives to the sum of true positives and false negatives. Given the closed-model assumption where the knowledge base consists entirely of verified truths, the evaluation of our fact-checking system adopts specific considerations for the metrics used to measure the system's performance [Zhou and Zafarini, 2020]. In this context, the primary focus shifts towards the system's ability to accurately classify claims from the test dataset as true (when they align with the knowledge base) or false (when they diverge). This scenario impacts the interpretation and relevance of *accuracy* and F1-*score* calculations. Under the closed-model assumption with a knowledge base composed solely of verified truths, every claim in the test dataset can be classified based on its congruence with this knowledge base. The classification outcome is binary: a claim is either in agreement (true) or disagreement (false) with the knowledge base. This dichotomy significantly influences the computation and interpretation of the evaluation metrics [Zhou and Zafarini, 2020]: *Accuracy*, in this specific scenario, still measures the proportion of claims correctly identified by the model. However, the interpretation leans heavily towards the model's proficiency in distinguishing between claims that are supported by the knowledge base versus those that are not. The formula remains unchanged. The F1-*score* remains a crucial metric for evaluating the balance between *precision* and *recall*. However, its calculation and interpretation must consider the closed-model nature of the knowledge base. The *precision* becomes particularly significant, as it reflects the model's accuracy in identifying true claims that accurately match the knowledge base content. *Recall* continues to indicate the model's ability to identify all relevant true instances from the test dataset. The formulas for *precision* and *recall* thus maintain their standard form, but their contextual interpretation aligns with the closed-model assumption.

In a closed-model with a truth-only knowledge base, the emphasis on false negatives (FN) and false positives (FP) gains prominence. FN represents missed truths—claims that should align with the knowledge base but are incorrectly identified as not doing so. FP reflects the model's overextension, labeling claims as true that do not find a basis in the knowledge base. Given the knowledge base's nature, TN becomes a conceptual challenge, as the model does not actively identify false claims against a truth-only backdrop but rather identifies claims not supported by the knowledge

base. Additionally, while *precision* will be given special consideration to align with the goal of minimizing the dissemination of false positives, an evaluation that accounts for all four metrics *accuracy*, *preicsion*, *recall*, and F1-*score* will provide a more comprehensive understanding of the system's capabilities and areas for improvement.

4.1 Evaluation of Graph-based Methods

This section delves into the assessment of graph-based strategies employed to augment the accuracy of fact-checking systems. Central to our investigation are two distinct types of graphs, each constructed utilizing different NLP tools: REBEL and CoreNLP, the details are described in the Section 3.6. These tools have been instrumental in creating structured representations from textual data, which, in turn, facilitate the verification of claims. The efficacy of these graph-based methods is evaluated across two categories of claims: original claims extracted from the dataset and those generated via GPT-3.5. A pivotal aspect of this evaluation is to ascertain whether triples extracted from LLM-generated sentences can significantly boost the fact-checking process. The primary aim of this evaluation is to critically analyze the utility and effectiveness of REBEL and CoreNLP in the context of fact-checking, particularly focusing on their capacity to handle and verify both original and GPT3.5-generated claims. The inclusion of generated claims introduces a novel dimension to the fact-checking endeavor, presenting both challenges and opportunities for enhancing the veracity assessment process. The logic used to verify triples is described in the Section 3.6.1.

Original claims serve as a baseline for our evaluation, representing the initial set of data points subjected to the fact-checking mechanism. These claims are directly sourced from the dataset and have not undergone any prior verification process. Generated claims introduce a layer of complexity, embodying the potential of advanced generative models to produce informative and plausible statements. These claims, synthesized by GPT-3.5, are evaluated to determine if the integration of such generated content can enrich the knowledge base and, by extension, improve the fact-checking framework's ability to discern truth from falsehood. The core objective of this section is to meticulously assess how graph-based methods, underpinned by REBEL and CoreNLP, perform in the verification of both original and generated claims. A key part of this evaluation is to explore the potential of triples extracted from generated sentences in enhancing the fact-checking process. By comparing the performance of these methods across different claim types, the study aims to shed light on the potential of graph-based approaches to elevate the

Table 4.1: Evaluation of results for method A applied to Original Claims.

Algorithm	Triple Type	Accuracy	Precision	Recall	F1-score
Fuzzy match	REBEL	0.555	0.552	0.451	0.500
Fuzzy match	CoreNLP	0.480	0.460	0.42	0.440
ShortPath_fuzzy	REBEL	0.561	0.576	0.341	0.433
ShortPath_fuzzy	CoreNLP	0.500	0.455	0.16	0.23
ShortPath_bert	REBEL	0.578	0.581	0.391	0.480
ShortPath_bert	CoreNLP	0.521	0.497	0.201	0.271
Node2Vec_fuzzy	REBEL	0.566	0.561	0.400	0.470
Node2Vec_fuzzy	CoreNLP	0.493	0.451	0.203	0.286
Node2Vec_bert	REBEL	0.555	0.559	0.390	0.460
Node2Vec_bert	CoreNLP	0.500	0.450	0.210	0.293
KL_fuzzy	REBEL	0.570	0.640	0.233	0.346
KL_fuzzy	CoreNLP	0.520	0.771	0.180	0,300
KL_bert	REBEL	0.578	0.660	0.250	0.365
KL_bert	CoreNLP	0.531	0.700	0.201	0.310

standard of fact-checking practices.

Disclaimer. In the tables presented, the terms "Pred_fuzzy" and "Pred_bert" following the algorithm names have specific meanings related to the discriminant used in the predicate analysis. Specifically: "Pred_fuzz" refers to the discriminant applied to predicates using a fuzzy method and "Pred_ber" refers to the discriminant applied to predicates using a BERT method. The details of the algorithms are described in this Section 3.6.1.

Original claims

The results of the different algorithms described above are shown below, both for triples extracted from the original claims with REBEL and CoreNLP; also using the two types of comparison methods A and B.

Based on Tables 4.1 and 4.2 detailing the performance of various algorithms using REBEL and CoreNLP on original claims, several observations and insights can be made regarding the performance of method A and method B across different algorithms and the implications for fact-checking systems. The results show low variability in performance across different algorithms and triple types, which underscores the importance of selecting the right combination of algorithm and embedding technique for fact-checking tasks. There is a noticeable trade-off between *precision* and *recall* in many cases.

Table 4.2: Evaluation of results for method B applied to Original Claims.

Algorithm	Triple Type	Accuracy	Precision	Recall	F1-score
Fuzzy match	REBEL	0.543	0.550	0.220	0.330
Fuzzy match	CoreNLP	0.557	0.550	0.500	0.502
ShortPath_fuzzy	REBEL	0.550	0.602	0.223	0.320
ShortPath_fuzzy	CoreNLP	0.550	0.555	0.308	0.397
ShortPath_bert	REBEL	0.560	0.610	0.242	0.450
ShortPath_bert	CoreNLP	0.588	0.630	0.351	0.460
Node2Vec_fuzzy	REBEL	0.551	0.587	0.267	0.470
Node2Vec_fuzzy	CoreNLP	0.550	0.550	0.333	0.415
Node2Vec_bert	REBEL	0.557	0.600	0.290	0.511
Node2Vec_bert	CoreNLP	0.545	0.545	0.330	0.410
KL_fuzzy	REBEL	0.555	0.635	0.110	0.180
KL_fuzzy	CoreNLP	0.530	0.810	0.210	0.320
KL_bert	REBEL	0.570	0.657	0.159	0.250
KL_bert	CoreNLP	0.554	0.701	0.230	0.350

This balance is crucial for tailoring the fact-checking system to specific requirements, whether minimizing false positives or ensuring no true claim is missed. These methods generally shows moderate performance across the algorithms. There is a noticeable variability in performance across different algorithms and triple types. For instance, the Fuzzy match algorithm demonstrates moderately good performance with REBEL triples, showcasing an F1-*score* of 0.500, which slightly drops when applied to CoreNLP triples. This suggests that the compatibility between triple extraction methods and matching algorithms significantly impacts accuracy and reliability. Remarkably, the KL_fuzzy algorithm, when paired with CoreNLP triples, achieves a *precision* of 0.771, the highest among the listed algorithms. However, its *recall* is comparatively low at 0.180, indicating a strong ability to correctly identify true claims but at the cost of potentially missing other true instances.

The method **method A** generally shows moderate performance across the algorithms. The "ShortPath_bert" algorithm with REBEL triples stands out with the highest F1-*score*, indicating a relatively balanced approach between *precision* and *recall*. Using **method B**, the Fuzzy match algorithm with CoreNLP triples shows an improvement in *recall* to 0.500 and an F1-*score* of 0.502, illustrating method B's effectiveness in identifying a broader range of true claims without significantly compromising *precision*. This demonstrates improved performance in some cases, particularly with

ShortPath_bert and CoreNLP, which yields a higher F1-*score* compared to its performance in method A. This suggests that method, which likely involves more stringent criteria for fact-checking, may enhance the effectiveness of certain algorithms, especially when paired with CoreNLP triples. The Node2Vec_bert and ShortPath_bert algorithms demonstrate their highest F1-*scores* with REBEL triples. This underscores the potential of combining advanced triple extraction methods with method B's nuanced matching strategy to improve overall fact-checking performance. While the KL_fuzzy algorithm with CoreNLP achieves the highest *precision* of 0.810, its *recall* remains low at 0.210, and its F1-*score* at 0.320, reflecting the ongoing challenge of balancing *precision* and *recall* in fact-checking systems.

The observed performances suggest that the choice of algorithm significantly impacts the fact-checking system's ability to accurately verify claims. The variability in performance across different setups underscores the need for tailored approaches, depending on the specific goals of the fact-checking task (e.g., prioritizing *accuracy* over *recall* or vice versa). Moreover, the results highlight the potential benefits of exploring more sophisticated models or combinations of embeddings and algorithms to improve both *precision* and *recall*. Particularly, the use of CoreNLP with certain algorithms under method B shows promise and may warrant further investigation and optimization.

Generated claims

The results of the different algorithms described above will be shown below, both for triples extracted from the generated claims with REBEL and CoreNLP, also using the two types of comparison methods A and B.

In Table 4.3 the performance metrics for generated claims using **method A** showcase a range of outcomes that reflect the intricate nature of automated fact-checking. CoreNLP shows a generally lower performance compared to REBEL in terms of *accuracy* and F1-*Score* across most algorithms. This might initially seem counterintuitive given the larger number of triples extracted with CoreNLP; however, it underscores the importance of the quality and relevance of the extracted triples over their quantity. The "ShortPath_bert" and "Node2Vec_bert" algorithms exhibit relatively high performance, especially with REBEL triples, indicating their robustness in handling generated claims. Notably, "ShortPath_bert" with REBEL triples achieves the highest F1-*score*, suggesting a balanced *precision-recall* trade-off. Conversely, the Node2Vec_fuzzy algorithm presents a compelling case for the effectiveness of embedding-based approaches, especially when combined with REBEL triples, underscoring the potential for leveraging semantic anal-

Table 4.3: Evaluation of results for method A applied to Generated Claims

Algorithm	Triple Type	Accuracy	Precision	Recall	F1-score
Fuzzy match	REBEL	0.555	0.571	0.327	0.416
Fuzzy match	CoreNLP	0.531	0.529	0.298	0.382
ShortPath_fuzzy	REBEL	0.560	0.566	0.400	0.4700
ShortPath_fuzzy	CoreNLP	0.512	0.500	0.320	0.390
ShortPath_bert	REBEL	0.581	0.591	0.431	0.500
ShortPath_bert	CoreNLP	0.550	0.550	0.351	0.431
Node2Vec_fuzzy	REBEL	0.565	0.562	0.546	0.508
Node2Vec_fuzzy	CoreNLP	0.511	0.495	0.353	0.412
Node2Vec_bert	REBEL	0.566	0.562	0.570	0.51
Node2Vec_bert	CoreNLP	0.511	0.495	0.345	0.407
KL_fuzzy	REBEL	0.557	0.605	0.251	0.371
KL_fuzzy	CoreNLP	0.555	0.662	0.170	0.266
KL_bert	REBEL	0.555	0.591	0.280	0.380
KL_bert	CoreNLP	0.581	0,670	0,25	0.364

ysis in verifying generated claims. For instance, "KL_fuzzy" and "KL_bert" with CoreNLP show higher *precision* but lower *recall*, which could indicate a tendency to be more conservative in identifying claims as true, potentially missing out on accurately identifying some true claims.

In Table 4.4 **method B** shows improved performance with CoreNLP for some algorithms, particularly "KL_bert", which attains the highest F1-*score* among all combinations This improvement suggests that method B's criteria or approach might be more aligned with the characteristics of triples extracted using CoreNLP. The Fuzzy match algorithm shows a notable improvement from method A to method B when using CoreNLP, indicating that the application of stricter criteria or additional processing steps in method B could help in refining the outcomes of simpler algorithms.

Across both method A and method B, the performance variations highlight the nuanced challenges in applying automated fact-checking to generated claims. The differential success across algorithms and triple types underscores the complexity of accurately identifying veracity in generated content, which often lacks the clear-cut factual basis present in original claims. The results illuminate the critical balance between *precision* and *recall*, particularly in the context of generated claims where misinformation may be subtly embedded. It also emphasizes the importance of selecting the appropriate triple extraction method and matching strategy, as these choices significantly impact the system's ability to discern truth from falsehood. The findings un-

Table 4.4: Evaluation of results for method B applied to Generated Claims

Algorithm	Triple Type	Accuracy	Precision	Recall	F1-score
Fuzzy match	REBEL	0.520	0.520	0.150	0.237
Fuzzy match	CoreNLP	0.545	0.544	0.370	0.440
ShortPath_fuzzy	REBEL	0.534	0.544	0.250	0.338
ShortPath_fuzzy	CoreNLP	0.526	0.517	0.351	0.421
ShortPath_bert	REBEL	0.552	0.591	0.312	0.391
ShortPath_bert	CoreNLP	0.571	0.580	0.45	0.418
Node2Vec_fuzzy	REBEL	0.536	0.540	0.288	0.376
Node2Vec_fuzzy	CoreNLP	0.521	0.509	0.360	0.421
Node2Vec_bert	REBEL	0.540	0.551	0.301	0.391
Node2Vec_bert	CoreNLP	0.541	0.510	0.362	0.423
KL_fuzzy	REBEL	0.518	0.520	0.191	0.270
KL_fuzzy	CoreNLP	0.601	0.580	0.181	0.275
KL_bert	REBEL	0.520	0.500	0.281	0.310
KL_bert	CoreNLP	0.620	0.651	0.360	0.463

derscore the necessity for continuous refinement of matching strategies, triple extraction methods, and the overall computational approach to enhance the *accuracy*, *precision*, *recall*, and F1-*score* of automated fact-checking systems. This endeavor is crucial for developing more robust and reliable models capable of navigating the complex landscape of generated misinformation.

4.2 Evaluation of Embdedded-Based Methods

This section delves into the exploration and evaluation of semantic matching techniques, which utilize embeddings to capture the contextual relationships within and across textual data. The focus is on assessing the efficacy of these techniques in enhancing the accuracy and reliability of fact-checking processes. Furthermore, a qualitative analysis of matching outcomes will offer insights into the strengths and limitations of each embedding model in the context of fact-checking. This Section 4 seeks to contribute to the broader understanding of how semantic analysis can be effectively leveraged to combat misinformation. The evaluation logics are described in Sections 3.6.2. The truthfulness of text X is determined based on the similarity threshold achieved compared to the summaries in the knowledge base (See Section 3.6.2). Different similarity thresholds have been established to label a text

Table 4.5: Evaluation of results for generated claims with different embedding models and similarity thresholds

Embedding	Threshold	Accuracy	Precision	Recall	F1-score
ADA	0.5	0.600	0.570	0.715	0.633
ADA	0.7	0.557	0.779	0.13	0.21
ADA	0.9	0.515	0.778	0.003	0.006
RoBERTa	0.5	0.490	0.490	1.00	0.653
RoBERTa	0.7	0.485	0.485	1.00	0.653
RoBERTa	0.9	0.485	0.485	1.00	0.653
ERNIE	0.5	0.485	0.485	1.00	0.653
ERNIE	0.7	0.485	0.485	1.00	0.653
ERNIE	0.9	0.521	0.511	0.230	0.32
distilBERT	0.5	0.485	0.485	1.00	0.653
distilBERT	0.7	0.485	0.485	1.00	0.653
distilBERT	0.9	0.502	0.49	0.939	0.647

as true. The evaluation tables for generated claims and generated explanations using different embedding models and similarity thresholds provide insightful data on the performance of each model under varying conditions. Here is an analysis focusing on the most and least performant models based on the provided data, considering the computational limitations encountered with some models.

Generated claims

In Table 4.5, the results obtained on the generated claims are explored. ADA's embedding model demonstrates a notable performance at a 0.5 similarity threshold, achieving the highest balance between *recall* and *precision*, which results in an F1-*score* of 0.633. However, as the threshold increases to 0.7 and 0.9, there's a significant drop in *recall*, indicating that while the *precision* remains high, the model struggles to identify true claims as the similarity requirement becomes stricter. Both RoBERTa and ERNIE show consistent performance across all thresholds in terms of *recall*, achieving a perfect score. This suggests that these models are highly effective in identifying relevant claims but at the cost of *precision*, which remains constant and results in a steady F1-*score* of 0.653. The perfect *recall* indicates that these models, at any threshold, are capable of capturing all true claims, a characteristic that could be highly valuable in scenarios where missing a true claim has serious consequences. Across all models, a similarity threshold

of 0.9 severely impacts the *recall*, except for RoBERTa and ERNIE, which maintain their *recall* but at the cost of overall performance as seen in their F1-*scores*. This drop for ADA and distilBERT suggests that extremely high standards of similarity may not be practical for effective fact-checking, potentially leading to the dismissal of true claims that do not meet the stringent criteria. distilBERT presents an interesting case, with its performance at a 0.5 similarity threshold mirroring that of RoBERTa and ERNIE in terms of *recall* and F1-*score*. However, at a 0.9 threshold, distilBERT manages to maintain a relatively high *recall* and an improved F1-*score* compared to ADA, indicating a better balance between identifying true claims and maintaining *precision*.

Generated Explanations

In Table 4.6, the results obtained on the generated explanations are explored. The absence of results for RoBERTa and ERNIE highlights the computa-

Table 4.6: Evaluation of results for generated explanations with different embedding models and similarity thresholds

Embedding	Threshold	Accuracy	Precision	Recall	F1-score
ADA	0.5	0.61	0.577	0.721	0.64
ADA	0.7	0.557	0.779	0.123	0.223
ADA	0.9	0.51	0.801	0.003	0.006
RoBERTa	0.5	-	-	-	-
RoBERTa	0.7	-	-	-	-
RoBERTa	0.9	-	-	-	-
ERNIE	0.5	-	-	-	-
ERNIE	0.7	-	-	-	-
ERNIE	0.9	-	-	-	-
distilBERT	0.5	0.485	0.485	1.00	0.653
distilBERT	0.7	0.485	0.485	1.00	0.653
distilBERT	0.9	0.55	0.53	0.77	0.63

tional challenges associated with these models. This underscores the importance of considering computational efficiency and resource requirements when selecting an embedding model for fact-checking tasks. At a 0.5 similarity threshold, ADA achieves a notable balance between *precision* and *recall*, resulting in an F1-*score* of 0.64. This suggests that ADA is capable of effectively identifying relevant explanations with a moderate level of strictness

in similarity, making it a potentially valuable model for explanation verification where both relevance and *accuracy* are critical.ADA demonstrates an increasing trend in *precision* as the similarity threshold is raised, reaching up to 0.801 at a 0.9 threshold. However, this increase in *precision* comes at a significant cost to *recall*, which drops to 0.003, highlighting the trade-off between accurately identifying highly similar explanations and the risk of overlooking relevant explanations that do not meet the stringent similarity criteria. distilBERT shows a remarkable consistency in *recall* across the 0.5 and 0.7 thresholds, achieving a perfect score. This indicates an exceptional ability of distilBERT to identify all relevant explanations at these levels of similarity, underscoring its utility in applications where missing an explanation is not an option.

The choice of model and similarity threshold should be tailored to the specific requirements of the task, balancing the need for *accuracy*, *precision*, and *recall* with computational feasibility. The choice between using generated claims or generated explanations depends on various factors, including the specific goals of the fact-checking system, available computational resources, and the nature of the information being verified. New claims seem to benefit from lower similarity thresholds, especially with the ADA model, which might make them more suitable for applications where the goal is to maximize coverage and ensure that as many truthful claims as possible are identified. generated explanations, on the other hand, show balanced performance at higher similarity thresholds with the distilBERT model, suggesting they might be preferable in contexts where minimizing false positives is critical and where *precision* is more important than total coverage. If the primary objective is to broadly identify as many truthful claims as possible, with a focus on maximizing coverage and inclusivity, then generated claims with the ADA model at a 0.5 similarity threshold appears to be the most balanced choice. It provides a good mix of *accuracy* and *recall*, indicating its effectiveness in identifying relevant information without being overly restrictive. If, however, the goal leans more towards *precision*, ensuring that what is identified as true is highly likely to be accurate (thus minimizing false positives), generated explanations with the distilBERT model at a 0.9 similarity threshold might be preferable. This approach offers a higher level of *precision*, suitable for applications where the cost of accepting a false claim is high.

4.3 Final Evaluation

The analysis of results for both original claims and generated claims across various algorithms and triple types (REBEL and CoreNLP), as well as the comparison between embedding-based methods and graph-based methods, reveals insightful differences in performance and implications for fact-checking systems. The observed variability in algorithm performance between original and generated claims underscores the complex nature of fact-checking across different types of content. Original claims, being derived directly from real-world data or established knowledge bases, tend to present a relatively structured and consistent semantic landscape. This consistency can make it easier for algorithms, especially those relying on pattern recognition or matching specific knowledge graph structures, to verify claims against known facts. Generated claims, however, introduce a layer of complexity and diversity that can challenge these algorithms. These claims are often the result of artificial generation processes, such as those produced by language models, which can introduce nuanced semantic variations or entirely novel contexts that were not present in the training data of the fact-checking algorithms. The semantic diversity of generated claims may stem from creative rephrasing, the introduction of new information, or the blending of facts from multiple domains, making it harder for algorithms to find direct matches in the knowledge base or to apply conventional logic used for original claims. Algorithms tend to show performance variations between original and generated claims, indicating that the text's nature influences fact-checking efficacy. Generally, generated claims present unique challenges, possibly due to their semantic diversity or complexity compared to original claims.

The distinction between Methods A and B plays a critical role in contextualizing the performance of fact-checking algorithms across original and generated claims: method A typically involves a more straightforward or lenient approach to matching claims against the knowledge base. This method's effectiveness is notably varied when applied to original versus generated claims. For original claims, which might be more straightforward and closely aligned with the knowledge base, method A's approach can be quite effective, as seen with algorithms like "ShortPath_bert" and "Node2Vec_bert" using REBEL triples. However, when tackling the complex semantic landscapes of generated claims, method A's performance can be less consistent, indicating a potential need for more nuanced analysis or stricter matching criteria to handle the variability introduced by generated content. method B, on the other hand, represents a more rigorous or stringent set of criteria for claim verification. This method's structured approach appears to complement the analysis of generated claims, particularly when coupled with CoreNLP

triples. The improved *precision* and *recall* observed with "KL_bert" and "Node2Vec_bert" under method B suggest that the additional rigor helps in navigating the semantic complexity of generated claims. method B's stringent criteria likely aid in filtering through the noise and identifying meaningful matches despite the diverse semantic variations. The success of certain algorithms under method A with REBEL triples highlights the efficiency of straightforward matching techniques when dealing with less semantically complex claims. These algorithms leverage the structured information within original claims to achieve high levels of *accuracy* and F1-*scores*. The scenario changes with generated claims, where method B, applied with CoreNLP triples, shows notable improvements. The rigorous evaluation criteria of method B seem to better address the challenges posed by the semantically rich and varied nature of generated content, enhancing the overall effectiveness of the fact-checking process. This insight suggests a strategic approach for fact-checking systems, advocating for the dynamic application of Methods A and B depending on the nature of the claims. Such adaptability not only enhances the *precision* of fact-checking efforts across a wide range of content but also ensures that the system remains robust against various forms of misinformation.

The comparison between embedding-based methods and graph-based methods in the context of fact-checking both original and generated claims offers a nuanced perspective on the strengths and limitations of each approach. The embeddings capture the semantic nuances of words or phrases based on their context within the text, enabling a deep understanding of the claim's meaning. This adaptability makes embeddings particularly effective for analyzing a wide range of claims, from straightforward factual statements found in original claims to the nuanced and varied expressions typical of generated claims. The ability to grasp complex semantic variations allows for a more flexible and comprehensive evaluation of claims, especially when the verification process requires interpreting subtle differences in meaning or when claims involve sophisticated language use. Contrarily, graph-based approaches rely on the structured representation of knowledge, where entities and their relationships are explicitly mapped out in a network. Algorithms navigate these networks to find connections between the elements of a claim and the information contained within a knowledge graph. This method's strength lies in its capacity to methodically evaluate the logical and relational structure of claims, offering a clear framework for assessing the accuracy of statements based on established facts. The structured nature of graphs facilitates a direct and interpretable matching process, which is highly valuable for verifying original claims closely aligned with the knowledge base's structure. The *precision* and *recall* balance in graph-based methods is indicative of their ability to

efficiently identify relevant and meaningful matches within a graph. The structured query and matching process enable these methods to precisely pinpoint accurate matches, reducing the likelihood of false positives (high *precision*) while ensuring that relevant facts are not overlooked (high *recall*). This balance is crucial for maintaining the integrity of the fact-checking process, ensuring that claims are verified against the knowledge base with both *accuracy* and completeness. Embedding-based methods excel in contexts where claims require verification against a broad or abstract concept rather than specific entities or relationships. By analyzing the semantic similarity between the claim's embeddings and those derived from potential matching facts, these methods can identify deep semantic correlations that might not be explicitly outlined in a knowledge graph. This capability is particularly beneficial for handling generated claims, which might introduce new information or present facts in a novel context, challenging the direct matching capabilities of graph-based methods.

Graph-based approaches, while requiring the construction and maintenance of a graph, often leverage the inherent structure of this representation to perform efficient queries and matching operations. This structured analysis can be less computationally intensive than processing large-scale embeddings, especially for algorithms optimized for specific types of graph structures. The computation of embeddings, particularly with advanced models like RoBERTa and ERNIE, involves significant processing power, especially when dealing with large datasets or complex models. The richness of embeddings comes at the cost of increased computational resources, which can be a limiting factor in scenarios with constrained computational budgets. Additionally, the dynamic nature of language and information necessitates frequent retraining or updating of models to capture the latest semantic nuances, further adding to the computational load.

In light of the detailed examination and comparative analysis of both original and generated claims across various algorithms, triple types, and methodological frameworks, it becomes evident that no singular approach holds supremacy across all dimensions of fact-checking. For this work embeddings methods excel in capturing the subtle semantic variations introduced by the artificial generation processes, enabling a more flexible and comprehensive evaluation of claims. The capacity of embedding-based approaches to navigate the broad and abstract nuances of language makes them particularly suited to verifying claims that deviate from conventional patterns or introduce novel contexts not accounted for in traditional knowledge bases, but in this evaluation we must take into account a very limited knowledge base which affects performance especially for the method with graphs, furthermore the latter having a very simple structure, without ontologies is

unable to fully capture the semantics.

Chapter 5

Conclusions and Future Work

The goal of this thesis was to enhance the accuracy and efficiency of fact-checking processes in the digital age, where misinformation proliferates rapidly. Through a meticulous methodology that integrates NLP, LLMs, and graph-based approaches, we have established a comprehensive framework capable of dissecting, analyzing, and verifying claims against a structured knowledge base of verified truths. Our exploration spanned from the construction of a "true" knowledge base derived from literature datasets to the use of triple extraction methods and the generation of new claims and explanations leveraging advanced models. The comparative analysis of different triple extraction methods and the strategic employment of generative models for creating new textual contexts have underscored the potential of combining traditional NLP techniques with the cutting-edge capabilities of LLMs. This work has conducted a thorough investigation into various matching strategies, ranging from triple matching to the application of semantic similarity assessments with word embeddings, to elucidate the complex mechanisms enabling automated systems to assess information veracity. This exploration across diverse methodologies has demonstrated the flexibility and adaptability inherent to automated fact-checking systems, underscoring the necessity of employing a comprehensive approach to truth verification. The triple matching, with its straightforward comparison of SPO triple structures between the claim and the knowledge base, represents a innovative method in fact-checking techniques. It relies on the explicit representation of information, allowing for clear and definable matches that can significantly streamline the verification process. However, the effectiveness of this method is inherently limited by its reliance on exact matches, which may not always capture the complexities and subtleties of natural language or the evolving nature of information. In contrast, the application of semantic similarity assessments through word embeddings introduces a layer of sophistication that more closely mimics

human cognitive processes in claim evaluation. This approach transcends the limitations of graph matching by considering the semantic context in which words and phrases are used, allowing for a deeper understanding of the claim's meaning. By analyzing the semantic proximity between the textual content of a claim and the information contained within the knowledge base, these models can discern subtleties and inferential connections that are not immediately apparent through structural analysis alone. The approach to matching, which integrates both the proportion of matching triples and the depth of semantic correlation. It acknowledges that the veracity of information cannot always be determined through binary or simplistic means. Instead, it requires a dynamic and context-sensitive analysis that reflects the multifaceted nature of truth . The implications of these findings are profound, suggesting a trajectory toward more sophisticated fact-checking models that not only replicate but also enhance human cognitive processes in evaluating claims. By leveraging the strengths of both direct matching and semantic analysis, future systems can achieve a more balanced and comprehensive approach to fact-checking.

To conclude this research, it is essential to emphasize the importance of using different methodologies and technologies in the fact-checking process. This decision reflects a core principle underpinning this study: the potential of a holistic approach to enhance the accuracy and reliability of automated fact-checking systems. Firstly, such integration substantially increases the system's effectiveness. By combining diverse techniques and methodologies, the fact-checking process becomes more robust and capable of tackling a wider range of challenges, including subtle semantic nuances and structural complexities of data. This synergistic approach not only broadens the system's coverage but also improves its precision, dynamically adapting to the specific characteristics of each claim being verified. Secondly, the integrated approach fosters continuous evolution of the fact-checking system. Cross-analyzing results from different methodologies provides valuable insights for ongoing optimization of algorithms. This capacity for learning and adaptation is crucial to keep the system at the forefront, allowing it to evolve in response to the changing dynamics of digital information and increasingly sophisticated strategies of disinformation spread. Despite the achievements, several limitations and unresolved issues remain, which are described in the following two Sections.

5.1 Limitations

One of the primary challenges faced in this research was the reliance on a general-purpose dataset that, despite its breadth, presented certain limitations. The dataset, while extensive, did not always provide the depth or specificity required to comprehensively evaluate the veracity of complex or niche claims. The general nature of the dataset meant that certain claims could not be verified with high confidence due to the absence of specialized or domain-specific knowledge. This limitation underscores the need for more diverse and detailed datasets in automated fact-checking, encompassing a wider range of subjects and incorporating expert-verified information from various fields. Enhancing the dataset would not only improve the accuracy of claim verification but also enable the system to tackle a broader spectrum of misinformation.

Another significant hurdle encountered in this research was the computational limitations inherent in processing large datasets and implementing advanced NLP and LLMs techniques. The computational demands of training sophisticated models like RoBERTa or ERNIE, especially when dealing with extensive datasets, required substantial resources that were at times beyond the scope of this study. These constraints impacted the scalability of the proposed fact-checking system and limited the complexity of the models that could be explored. Furthermore, the intensive computation required for semantic similarity assessments and triple extraction processes often necessitated compromises in terms of model accuracy or depth of analysis.

The process of triple extraction, crucial for structuring the information within the claims and the knowledge base for comparison, also presented its set of challenges. While employing multiple methods for triple extraction, improved the robustness of the approach, it also introduced variability in the quality and consistency of the extracted triples. This variability occasionally affected the matching accuracy, as the effectiveness of the fact-checking system heavily relies on the precision of the triple representation. The exploration of various matching strategies, though illuminating, also revealed limitations in the current methodologies. While direct triple matching offered a straightforward approach, its reliance on exact matches often fell short in capturing the nuanced meaning of more complex claims. On the other hand, semantic similarity assessments, despite their potential for deeper analysis, were sometimes hampered by the computational demands and the challenges of accurately capturing semantic nuances within the constraints of available models and embeddings.

5.2 Future Work

The journey of this research has been both enlightening and challenging, uncovering the vast potential of automated systems in combating misinformation while also revealing the limitations and complexities inherent in this endeavor. A comprehensive roadmap is outlined, highlighting key areas where further exploration and development could yield significant advancements in automated fact-checking.

- A cornerstone of any robust fact-checking system is its knowledge base. The effectiveness of automated verification is inherently tied to the breadth and depth of this underlying database. Future efforts should focus on continuously **updating and diversifying the knowledge base** to encompass a wider array of domains and incorporate the most recent information. This expansion is not merely about quantity but also quality, ensuring that the database reflects a broad spectrum of verified knowledge from diverse sources. By achieving this, the applicability and reliability of the fact-checking system will see significant improvement, making it a more versatile tool in combating misinformation across various fields.

- The **computational** demands of processing extensive datasets and employing sophisticated models present a considerable challenge. Future research should aim at developing more efficient algorithms and models that minimize computational overhead without sacrificing the accuracy of claim verification. This could involve innovations in algorithm design, leveraging more advanced hardware, or exploring novel approaches to model training that require less computational resource. Enhancing computational efficiency will not only make advanced fact-checking systems more accessible but also enable their application on a larger scale.

- Future work should delve into **deeper semantic analysis** techniques that capture the intricacies of language more effectively. A one-size-fits-all approach to claim verification often falls short due to the diverse nature of information and claims encountered. Investigating adaptive matching strategies that dynamically adjust based on the claim type and context represents a promising area for future research. Such strategies could offer a more personalized and accurate verification process, accounting for the specific characteristics of each claim and adapting the verification methodology accordingly. This adaptability is key to developing more sophisticated fact-checking systems capable of handling the complex landscape of digital information .

- Additionally, the integration of a knowledge graph augmented with **ontologies** represents a promising direction for future research in automated fact-checking. Ontologies, with their structured representation of knowledge domains and the relationships between concepts, offer a powerful mechanism to enrich the knowledge base beyond mere facts. By leveraging ontologies within a knowledge graph, fact-checking systems can gain a deeper understanding of the context and semantics underlying the claims being verified. This approach not only enhances the precision of matching algorithms by providing a more nuanced framework for evaluating claims but also facilitates the exploration of more complex queries and inferential reasoning that were previously challenging to address.

- The utilization of **graph-based algorithms** in conjunction with an ontology-enriched knowledge graph opens up new avenues for optimizing the fact-checking process. These algorithms can navigate the richly structured data more effectively, identifying not just direct matches but also inferring connections and relevancies that may not be immediately apparent. This capability is particularly advantageous for evaluating claims that involve subtle nuances or that span across multiple domains of knowledge. The potential incorporation of ontology-enriched knowledge graphs represents a significant opportunity for advancing the state of automated fact-checking. This evolution towards more sophisticated and semantically aware systems promises to elevate the accuracy, reliability, and transparency of fact-checking efforts, moving us closer to the goal of combating misinformation with informed, data-driven solutions.

In conclusion, the groundwork laid by this thesis in advancing automated fact-checking marks a significant leap forward in the ongoing battle against misinformation. Yet, the journey ahead is filled with untapped potential for innovation and improvement. By exploring these avenues for future research, we can continue to build upon the solid foundation established, driving forward towards a future where accurate and reliable information prevails. The dedication to enhancing automated fact-checking systems not only contributes to the integrity of our digital ecosystems but also empowers society to make informed decisions in an increasingly complex information landscape.

Bibliography

Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2), 2017.

Yashoda Barve and Jatinderkumar R. Saini. Healthcare misinformation detection and fact-checking: A novel approach. 2021. doi: https://doi.org/10.14569/IJACSA.2021.0121032.

Yashoda Barve, Jatinderkumar, and R. Saini. Detecting and classifying online health misinformation with 'content similarity measure (csm)' algorithm: an automated fact-checking-based approach. 2023. doi: https://doi.org/10.1007/s11227-022-05032-y.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021. doi: 10.1145/3442188.3445922. URL `https://dl.acm.org/doi/10.1145/3442188.3445922`.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008. doi: 10.1145/1376616.1376746.

Tomaz Bratanic. *From Text to Knowledge: The Information Extraction Pipeline*. 2022.

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

Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Grégoire Burel, Tracie Farrell, Martino Mensio, Prashant Khare, and Harith Alani. Co-spread of misinformation and fact-checking content during the covid-19 pandemic. 2022. doi: http://dx.doi.org/doi:10.1007/978-3-030-60975-73.

Huguet Cabot, Pere-Llus, Navigli, and Roberto. Rebel: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.204. URL https://aclanthology.org/2021.findings-emnlp.204.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4299–4307, 2017. URL https://arxiv.org/abs/1706.03741.

Adam Cohen et al. Fuzzywuzzy: Fuzzy string matching in python, Accesso 2024.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3rd edition, 2009. ISBN 978-0262033848.

Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. 2021. doi: https://arXiv:2010.09926v1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 (NAACL-HLT), 2019. URL https://arxiv.org/abs/1810.04805.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014. URL https://arxiv.org/abs/1406.2661.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016. doi: 10.1145/2939672.2939754. URL https://dl.acm.org/doi/10.1145/2939672.2939754.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Proceedings of the 7th Python in Science Conference (SciPy2008), 2008. URL https://networkx.org/documentation/networkx-1.10/reference/bibliography.html.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997. doi: 10.1162/neco.1997.9.8.1735.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, José Emilio Labra Gayo, Sabrina Kirrane, Sebastian Neumaier, Axel Polleres, Roberto Navigli, Axel-Cyrille Ngonga Ngomo, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge graph. 2023. doi: https://doi.org/10.48550/arXiv.2003.02320.

Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. 2017.

Dirk Hovy and Shannon L. Spruit. The social impact of natural language processing. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 591–598, 2016. doi: 10.18653/v1/P16-2096.

Viet-Phi Huynh and Paolo Papotti. Towards a benchmark for fact checking with knowledge bases. 2018.

Xiaoqian Jiang, Lucas M. Glass, Todd Rogow, et al. Artificial intelligence in healthcare: Past, present and future. *Stroke and Vascular Neurology*, 2 (4):230–243, 2020. doi: 10.1136/svn-2018-000101.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus, 2017.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson, 3rd edition, 2019.

Qrious Kamal. *Building Knowledge Graphs with Rebel: Step By Step Guide for Extracting Entities Enriching Info*. 2023.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL `https://arxiv.org/abs/2006.06676`.

Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo2, James Thorne, and Edward Choi. Factkg: Fact verification via reasoning on knowledge graphs. 2023. doi: https://arXiv:2305.06590v2.

Neema Kotonya and Francesca Toni. Explainable automated factchecking for public health claims. 2018. doi: https://github.com/neemakot/Health-Fact-Checking.

Eric Lazarsk and Cynthia Howard Mahmood Al-Khassaweneh. Using nlp for fact checking: A survey. 2021. doi: https://doi.org/10.3390/designs5030042.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880, 2020. doi: 10.18653/v1/2020.acl-main.703. URL `https://www.aclweb.org/anthology/2020.acl-main.703`.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. 2019. URL `https://arxiv.org/abs/1907.11692`.

Weichen Luo and Cheng Long. Fact checking on knowledge graphs. 2020. doi: https://doi.org/10.1007/978-3-030-62696-9_7.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. 2014. doi: 10.3115/v1/P14-5010.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. ISBN 0-262-13360-1.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN 978-0521865715.

Noah Mayerhofer. *Construct Knowledge Graphs From Unstructured Text*. 2023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013a. URL `https://arxiv.org/abs/1301.3781`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b. URL `https://arxiv.org/abs/1310.4546`.

Matthias Nickel and Hoifung Poon. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. In *Proceedings of the IEEE*, 2016. doi: 10.1109/JPROC.2015.2483592.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. 2012.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. doi: 10.3115/v1/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Volume 1 (Long Papers), 2018. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI Blog*, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS 2019*, 2019. URL https://arxiv.org/abs/1910.01108.

Tim Schopf, Karim Arabia, and Florian Matthes. Exploring the landscape of natural language processing research. 2023. doi: arXiv:2307.10652.

Sumin Seo, Heeseon Cheon, Hyunho Kim, and Dongseok Hyun. Structural quality metrics to evaluate knowledge graph quality. 2022. doi: https://arxiv.org/pdf/2211.10011.pdf.

Jane Smith and John Johnson. Combating the covid-19 infodemic: A joint information system approach. *Journal of Public Health Policy*, 43(2), 2022. doi: 10.1057/s41271-022-00314-2.

John Smith, Jane Doe, and Alex Johnson. Understanding and mitigating hallucinations in large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=SampleID.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019. URL https://arxiv.org/abs/1904.09223.

James Thorne and Andreas Vlachos. Automated fact checking: Task formulations, methods, and future directions. *ACM Computing Surveys*, 51(5): Article No. 93, 2018. doi: 10.1145/3232676.

Marco Viviani and Stefano Di Sotto. Assessing health misinformation in online content. 2022. doi: https://doi.org/10.1145/3477314.3507238.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Deep reinforcement learning from human preferences. In *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*, 2022. URL https://arxiv.org/abs/2201.11903.

Joseph Weizenbaum. Eliza - a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9, 1966. doi: 10.1145/365153.365168.

Jevin D. Westa and Carl T. Bergstromb. Misinformation in and about science. 2019. doi: https://doi.org/10.1073/pnas.1912444117.

Wikidata contributors. Wikidata. `https://www.wikidata.org`.

World Wide Web Consortium (W3C). Owl 2 web ontology language document overview (second edition), 2012.

World Wide Web Consortium (W3C). Rdf 1.1 primer, 2014.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018. doi: 10.1109/MCI.2018.2840738.

Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. Automated fact-checking: A survey. 2021. doi: https://doi.org/10.48550/arXiv.2109.11427.

Xinxy Zhou and Reza Zafarini. A survey of fake news: Fundamental theories, detection methods, and opportunities. 2020. doi: https://arXiv:1812.00315v2.

Acknowledgements

Ringrazio il Professore Marco Viviani per il suo sostegno e la sua guida fornita durante il mio percorso accademico; ringrazio anche il Professore Rafael Nyssen Penaloza per avermi assistito durante questo lavoro. La loro assistenza è stata fondamentale e ha avuto un impatto significativo sul mio sviluppo professionale.
Ringrazio la mia famiglia per l'opportunità, per il supporto durante tutto il mio percorso e per i scarifici fatti, dedico a loro questo lavoro.
Ringrazio i miei fratelli di "giù" che anche da lontano non mi hanno fatto mai sentire solo.
Ringrazio le persone che ho incontrato qui e che mi hanno accompagnato in questo viaggio dal primo all'ultimo giorno.
Ringrazio Milano e la Bicocca per questi due anni pieni di vita e per avermi fatto vivere nuove esperienze, sia belle che brutte.
Ringrazio chi mi supporta da lassù.
Ringrazio me stesso per non aver mai mollato.