Author:
Bindris, Nouf M

Title:
Natural language claims consistency checking using probabilistic reasoning with explanation

Natural Language Claims Consistency Checking Using Probabilistic Reasoning with Explanation

By

NOUF MANSOUR BINDRIS

Intelligent Systems Laboratory
Department of Computer Science
Faculty of Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

MAY, 2021

Abstract

Due to the increasing amount of information that is available on the Internet (and sometimes, only on the Internet), especially on social media sites, it can even be difficult for experts to differentiate between what is true, false, or deliberately falsified. False claim detection has therefore become a vast research area, with extensive work on finding a concrete and robust solution to the deliberate deception of others with incomplete and/or fabricated information. While manual fact-checking is clearly possible, it is extremely time-consuming and unmanageable on a large scale. The automated analysis of online news and other digital content is important for many reasons, leading to calls for automated fact-checking systems that can verify the truthfulness of statements.

Previous work in this area has largely focused on analysing the characteristics of the natural language used, for example, writing style and grammar rules. In contrast, this study looks at automated reasoning, with statements in online texts being checked for consistency with a knowledge base. Natural language processing is used to support this. The author considers that such analysis is best achieved by modelling false claims as claims that are inconsistent with a trusted knowledge base.

Consequently, the fact automated consistency testing (FACT) approach was introduced as a system of checking the consistency of facts in information that is not explicitly mentioned in an extracted text corpus. This approach is based on the following elements: information extraction (from natural language texts); checking claims against a trusted knowledge base, including against relations inferred via a logic system; the incremental building of a trusted knowledge base via a continuous learning technique, and the generation of explanations to facilitate users' understanding of the system's decisions. Consistency is checked using probabilistic soft logic (PSL) and a Markov logical network (MLN) for the purpose of comparison.

The results of this study indicate that automatic fact-checking algorithms offer a means of facilitating the detection of false claims, including what is commonly referred to as 'fake news'. Here, FACT contributes to new knowledge through its ability to check the consistency of claims that are not explicitly mentioned in a text corpus. Thus, FACT permits the veracity of information found on the Web to be checked. The approach was demonstrated by checking facts about family-tree relationships against a corpus of Web resources, relating to the UK royal family and political relations.

Dedication

I dedicate this thesis to...

The memory of my dad, whose greatest wish for me was to obtain my PhD,
My great mum,
My beloved husband,
My always supportive girl Shouq,
My little man Saad,
My lovely boy Mansour who came into the world during the early stages of this work.

ACKNOWLEDGMENTS

Firstly, thanks be to Allah (God) for giving me the strength, courage, determination and drive to complete this PhD thesis. The research and writing up of this project has been a very interesting and valuable experience: a transformative learning journey. In addition, I would like to express my sincerest thanks to all those who made this thesis possible, whether directly or indirectly. In particular, I include King Saud University, to whom I owe my utmost gratitude for granting me a scholarship, thereby providing me with the financial support to undertake this programme.

Aside from the above, I wish to thank my first supervisor, Professor Nello Cristianini, who shared his passion for the research field over the course of my project. Professor Cristianini was consistently patient and ready with constructive advice for me. His guidance, supervision, and time spent reviewing and critiquing my work helped me improve my research and thesis greatly. In short, he taught me the true meaning of research. My deepest appreciation is also due to my second supervisor, Professor Jonathan Lawry, who always found the time to help and support me with my study. Neither must I forget my co-worker and friend, Saatviga Sudhahar, for whose continuous support I am so grateful, and with whom I enjoyed so many productive meetings and intellectual exchanges.

Closer to home, special thanks go to my devoted mother, Asma Aleissa, whose love and affection have been the source of motivation and encouragement for my studies. Likewise, a big 'thank you' to my wonderful husband, Ahmed Aleissa, for his patience, support, and helpfulness every step of the way. No less important have been the understanding and patience shown to my by my young daughter, Shouq, and little boys, Saad and Mansour, especially during the final critical stages of completing this work. Moreover, a huge thank you goes out to my sisters, Mashael and Alanoud, and my brother Nawaf, simply for being there when I needed them. Likewise, a special thank you is due to my beloved little brother and flatmate, Mishal, for his unconditional love and encouragement.

Last but by no means least, a very special thank you is due to my friends, Asma Alzaid and Nora Alkhamees for their unconditional love, encouragement, endless support, and the unforgettable days we spent together in Bristol, Colchester, London, and Leeds. I am deeply grateful to all my friends in Bristol for their continuous presence my life, and for making my time in Bristol such a wonderful and memorable experience. Finally, I would like to thank everyone who generously gave of their time and effort to help me throughout my doctoral research; there are too many of you to name here, but I will never forget your friendship, professionalism, and contribution.

Nouf Bindris
May 2021, Bristol

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED:.......NOUF BINDRIS....................... DATE: ....28-05-2021.......

Table of Contents

LIST OF TABLES

List of Figures

Chapter 1

Introduction

1.1 Introduction

The development of the Internet in the 1960s was a significant breakthrough, which will continue to change lives for generations into the future. Through this significant infrastructure, numerous and diverse methods of sharing knowledge, information, and data have emerged [1, 2]. The Internet was initially a US military development, evolving for uses within the commercial and public sectors. Since then, daily human activities, methods of communication, and lifestyles have been drastically transformed. For example, the versatility of Internet services has birthed a new renaissance, in which the prospect of a decline in ignorance, hunger, disease, and poverty appears to be closer than ever before [3]. Furthermore, cultural exchange is taking place worldwide, thereby enabling greater understanding and empathy between nations and individuals, as well as the global distribution of products and goods. Indeed, the Internet is one of the key inventions of the 20th-century [4].

Another important feature of the Internet is the increased social interaction that it facilitates, and the continually expanding content of the World Wide Web. Aside from this, there is an urgent demand for data analysis and utilisation in data-driven decisions to enhance all fields, including education [5]. However, this becomes complex and confusing, due to the huge volume of data that exists on the Internet – an estimated 2.5 million terabytes of data generated daily. This is called ‘big data’ and it requires especially sophisticated processing methods and algorithms, such as semantic analysis, to produce valuable meaningful information. This information can be employed to assist real-time decisions, offer insights to individuals in all sectors and roles, enhance business revenue, and manage risk [6]. This problem of complexity and data volume is now further exacerbated by the advent of the Internet of things (IoT) and cloud computing, because vast amounts of data are thereby generated by devices, objects, and sensors [7, 8].

To date, this massive amount of data has brought new challenges, such as potential violations of privacy, and the need for accountability and authenticity. In sum, there is no guarantee that a user's private data will not be shared. Moreover, it is impossible to determine the author of content on every website. In addition, there is no way of determining whether the information being presented is true or false [1, 2, 9, 10]. To elucidate, it is proven that social bots are deployed to disseminate articles very effectively from inauthentic sources. Fake content is established in the first few seconds of being published, before an article gets spread widely. In particular, bots target users who have many active followers and who keep adding replies and mentions. Users may be vulnerable to this manipulation and unable to recognise such misinformation; consequently, they continue to share false information [11].

For this reason, there are various techniques for checking the veracity of information and identifying whether information published on the Internet is false or accurate. One of the earliest techniques is the manual process, which is used to distinguish fake from genuine news. With this technique, a journalist conducts a manual examination of the text to ascertain its trustworthiness. He or she will be regarded as a fact-checker, giving an expert opinion. On this basis, decisions may be made concerning the validity of a claim, or the level of credibility that should be attributed to a particular item of information. However, this technique cannot yield prompt results or be applied to large data sets and data streams. Instead, other methods are preferred for their effectiveness in handling big data, with less subjectivity. Thus, the use of modern technologies such as machine learning and deep learning, known as automated fact-checking techniques, are gaining attention and winning trust.

Consequently, nowadays, automated fact-checking systems are employed by companies, groups of individuals, and decision-makers to conduct research and analysis, while also ensuring a consistent level of truthfulness in the information used. Moreover, there is an urgent need to further develop those systems to eliminate the human factor and achieve a higher percentage of accuracy and efficiency [7, 12–14].

1.2 Research Questions

Research questions bring forward the necessary arguments that are critical for building the structure of a thesis. This includes further discussion, elaboration, and the development of concepts.

The fundamental research question addressed in this study is worded as follows:

- **How can probabilistic reasoning be used to check the internal consistency of a set of claims?**

. This question entails the following questions:

- How can the construction and dynamic expansion of a knowledge base enhance checking the validity of claims?
- What is gained by employing probabilistic reasoning in such systems?
- What measures should be used to quantify the efficiency of the proposed solution?

1.3 Research Aims and Objectives

1.3.1 Research Aims

As set out above, this thesis aims to answer the question: *How can probabilistic reasoning be used to check the internal consistency of a set of claims?*. In other words, this study seeks to discover how probabilistic reasoning can be utilised to establish the validity of fact-checking systems to achieve better performance.

For this purpose, fact-automated consistency testing (FACT) was formulated in this study, this being an automated fact-checking approach, which employs probabilistic reasoning to assess the consistency of various claims, grounded in a trusted knowledge base. In addition, a claim explanation technique was developed to help anyone adopting the approach outlined in this study to understand the rationale for the decisions underpinning each claim.

1.3.2 Research Objectives

Based on the research aim delineated above, the following research objectives were defined to obtain a clear description of a set of achievable tasks. The following tasks are objectively measurable in terms of their performance, thereby representing the desired research outcomes. These research objectives are listed below:

- To gather articles related to a specific claim and to extract the relevant relationships. This would be implemented using a Web crawling technique to browse Web indexes, wherein the identified links would be highlighted and then subjected to Web scraping techniques. Web scraping would clean the extraneous data from the relevant articles obtained during the previous stage, collecting them in an appropriate format. Finally, information extraction tools would be used to extract relations and properties for building the knowledge base, which then can be used to check for fact-consistency.
- To expand the knowledge base through a process of continuous learning, carried out via incremental building.
- To use a family tree graphic as a means of displaying a human-readable visualisation of the knowledge base contents.

- To apply a novel fact-checking approach to the assignment of truth values to claims, using both the probabilistic soft logic (PSL) and Markov logic network (MLN) frameworks.
- To generate an automated explanation of the fact-checking process for a particular claim, in order to provide a humanly comprehensible description of the automated decision.
- To evaluate the explanations generated by human evaluators in relation to the decision and the fact of being comprehensible.

1.4 Key Findings

Based on the results of the experiments, it can be concluded that:

- Generating a knowledge base related to a given claim, which will be used to validate the authenticity of that claim, can help produce an effective consistency-validation method.
- Such a knowledge base can be generated by employing information extraction techniques to grasp the relationship between two named entities. In this research, the General Architecture for Text Engineering (GATE) software toolkit using its processing resource A Nearly-New IE system (ANNIE) and the Java Annotation Patterns Engine (JAPE) was utilised and proved to be very efficient.
- To enhance the performance of a system for consistency checking, founded on a knowledge base, the knowledge base needs to be expanded continuously. This can be achieved by crawling the Web to source related articles, while also employing information extraction techniques.
- Probabilistic logical reasoning frameworks, such as MLN and PSL, associated through a corpus, are effective techniques for fact consistency checking. These frameworks can check the consistency of facts that are not explicitly mentioned in the text, which is an advantage over the linguistic methods used in previous fact-checking methods.
- Probabilistic soft logic (PSL) performed better than Markov Logic Networks (MLN) in fact consistency checking.
- The explanation of a decision related to a fact-checking system is crucial for the end-user.
- The results of the fact-checking systems need to be explained; this can be achieved by employing probabilistic logical reasoning.
- An effective method of measuring the worthiness of the explanations generated is to employ human evaluators.

- Human evaluation can be designed in a way that reduces bias. This can be achieved by employing a number of human evaluators from different backgrounds, giving them the same set of claims, along with their automatically generated decisions and explanations for those decisions. If the human evaluators agree on their worthiness, it means that their decision is free of bias, and the system may be deemed to be an effective tool for fact consistency checking.

- The proposed framework pipeline FACT employed probabilistic soft logic (PSL) and an incremental dynamically generated knowledge base to create a fact-checking system that is interpreted by design. It can generate decisions explanations as text and graphs. It is compared to the state-of-the-art CredEye and performed significantly better in terms of truthfulness, understanding, and satisfying. These results are based on human evaluators using a bias-free setup.

- The proposed approach gives proper justification for positive decisions, but for negative decisions, it shows that there is no evidence to support the claim, and an alternative relationship supporting this claim is given if available.

- The proposed approach performed equally well on true and false claims in terms of true and understandable metrics.

1.5 Thesis Structure

The thesis comprises seven, chronologically interconnected chapters, each made up of sections and subsections that are linked by logical relationships. This structure allows for a detailed presentation, as well as a general overview of the research rationale, as outlined below:

Chapter One introduces the thesis and presents an overview of the background to the study, especially with respect to core research in the field of fact consistency checking in relation to natural language processing. In this Chapter, the research motivation and core research questions are defined. This includes the reasons for conducting the study, and a description of the strategy adopted in its execution. The responses to these concerns have paved the way for developing comprehensive solutions, such as planning the research structure and implementing important deliverables. In particular, it should be mentioned that the research questions provide the key insights that are necessary to develop the research aims and objectives. Subsequently, the key questions were broken down into simpler and more manageable questions that could be answered more effectively. Furthermore, the anticipated research contributions and main achievements of this study are identified in this introductory Chapter. Thus, it was possible to foreshadow some of the study's core successes and demonstrate that these are perfectly aligned with the current field of research.

Chapter Two details the background to this study, with an extensive and detailed review of the main research studies that have already been conducted in the field of fact consistency checking. Chapter Two therefore begins with an in-depth explanation of fact-checking problems. Next, it examines information extraction and all its related topics, followed by a presentation of the different techniques and frameworks applied in this area. Furthermore, the Chapter dissects the research topics that have hitherto been addressed, specifically notions of reasoning and inference. Hence, the relationship between these two notions are established for greater clarification. Finally, explainable artificial intelligence (AI) is discussed in some detail.

Chapter Three then begins with an explanation of the research methodology, accompanied by a detailed description of the proposed research framework, whereupon FACT is presented. This includes the notion, components, structure, and algorithm. The main steps in FACT are explained in detail, illustrated by examples such as Web crawling, pre-processing, post-processing, and the expansion of the knowledge base by re-querying the Web. Claim-consistency checking via probabilistic reasoning is subsequently presented. Finally, FACT approach validation is introduced, which includes validation of the information extraction and fact-consistency decision.

Chapter Four describes the different stages of information extraction that were used to build the knowledge base, with a case study on the UK's Royal Family. It begins by defining the primary data-set. The use of Java Annotation Patterns Engine (JAPE) grammar to extract the relationship between two named entities is then described. Next, the algorithm of solving named entity ambiguity is introduced to unify the named entity and storage them is outlined. Finally, the UK Royal Family tree was visualised using a GEDCOM file, allowing key relations in this family to be evaluated as a case study.

Chapter Five presents the fact-checking methodology, where two kinds of probabilistic reasoning are applied: PSL and Tuffy – the latter being an MLN inference engine. These are the two main reasoning frameworks in the field. The Chapter starts by outlining the syntactic structures of the PSL and MLN models, followed by a comparison between the two. The various components of the model language and the syntactic structure involved in this setup are presented, including the predicates, observation, descriptions of targets, logical rules, and different priors. Finally, a case study on the UK Royal Family was utilised to evaluate the performance of PSL and MLN in the field of fact-checking. In addition, a political relations case study was selected to evaluate the performance of PSL, purely in the field of fact-checking.

Chapter Six demonstrates the way in which automatic explanations are produced in relation to automated decisions. This Chapter begins with an introduction together with the means through which automatic explanations of these decisions were developed. This explanation includes an examination of both the text and tree representations. Next, following this theoretical explanation of the decision methodology, experiments were performed, wherein human evaluators were engaged in evaluating the claims explanation. Finally, the framework result proposed in this study was compared with the results of CredEye evidence, in order to benchmark the performance of the proposed FACT system.

Chapter Seven concludes this thesis, consisting of an a general summary of the study, and a conclusion to the entire thesis. The research findings and contributions to knowledge are presented therein, as well as the research limitations. In light of these limitations, potential areas of future research are recommended, specifically in relation to fact-consistency checking via NLP.

1.6 Publications

A number of papers were published in the course of writing this thesis. These are referred to extensively in Chapters Three, Four, and Five, and are cited below:

- Bindris, N., Sudhahar, S. and Cristianini, N. (2018) 'Fact checking from natural text with probabilistic soft logic', in Duivesteijn, W., Siebes, A. and Ukkonen, A. (Eds.), Advances in Intelligent Data Analysis, XVII. Cham: Springer International Publishing, pp. 52-61.
- Bindris, N., Cristianini, N. and Lawry, J. (2020) 'Claim consistency checking using soft logic', Machine Learning and Knowledge Extraction, 2(3), pp. 147-71.

CHAPTER 2

LITERATURE REVIEW

While attempting to develop an approach for fact checking claims in NLP, in this chapter we present a background and a literature review of related topics. The literature review chapter covers five main topics, in sequence: truth discovery; fact checking; information extraction; probabilistic reasoning; and finally, linking financial eXplainable artificial intelligence (XAI).

2.1 Truth Discovery

2.1.1 What Is Truth Discovery?

When there are conflicting values, or there is information from multiple sources, regarding a particular instance in a given data item or dataset; finding the actual or true value in that given data is known as the process of truth discovery or truth finding. Truth finding involves finding the actual value in conflicting and noisy data from multiple sources.

In this age of information overload, a great deal of information is accessed on a daily basis. Thus, it is possible that the sources of this information are prone to error, whether due to a lack of expertise in this regard, out of malicious intent, or because of malicious content, staleness, etc. Other reasons include missing records or the provision of partial information. In such a situation, accurate and complete information should be assessed from multiple sources, all claiming to be authentic. Identifying accurate information is sometimes referred to as truth discovery. In this current study, it is called fact-checking [15]. Here, a method of supporting the fact-checking of statements from natural text is demonstrated. These sources include online news, encyclopaedias, and academic repositories, and it is ascertained whether they violate the knowledge that is implicitly present in a reference corpus

This type of information can take the form of fake news, misinformation, disinformation, and rumour detection. Spreading false information and fake news can generate revenue for those who are spreading it, for example, politicians and the producers of such content. This is a matter of concern for journalists. Without fact-checking, the flood of disinformation cannot be stopped. Therefore, journalists and fact-checkers work on the prevention of fake news and false data, although more efforts are still urequired to stop this dissemination of falsified and misleading information [16]

2.1.1.1 Fake News

Across the globe, a high volume of falsified information is available on almost every topic. Instead of original content, there are statements that could be categorised as falsified or overstated, as well as false claims, fake videos, etc. This kind of false information exists simply because of its popularity and the uncontrolled flood of information on the Internet, especially on social media networks. It is an unprecedented situation, posing a threat to society and affecting populations all over the world. One example of fake news is when the Twitter account of the US new agency, Associated Press, was hacked. Fake news was subsequently posted from this account, falsely reporting an explosion in the White House, in which US President, Barack Obama, was allegedly injured. This news wiped out USD139 billion in a flash. To combat this kind of fake news, fact-checking systems are put in operation, ranging from manual systems that involve human personnel, to machine learning and deep learning methods [17].

2.1.1.2 Misinformation

Misinformation includes missing records and the provision of incomplete information, leading to the compromised accuracy of reports from those sources. This is why misinformation presents a task for fact-checking [17].

2.1.1.3 Disinformation

Disinformation includes the intentional dissemination of fake news to mislead news consumers. It also includes so-called ‘clickbait’, or the false incentivisation of users to click on links, thereby generating revenue for the source (the source asks users to click on a misleading link, which fails to correspond to the source’s claims, and does not give the user what they anticipate). The other type of disinformation consists of manipulating public opinion via false news and falsified information. This may be for political gain in election campaigns, or can be intentionally divisive, as in pitting one ethnic group against another [17].

2.1.1.4 Rumour Detection

Rumours consist of less formal news items or statements of questionable accuracy. Rumour detection therefore differentiates between false and genuine information. Rumour detection is defined in diverse ways. For some, rumour detection consists of the classification of posts into the credible and incredible; for others, it represents the link between the truthfulness of a claim and its expression in a post. Rumours are also classified according to true-rumour, false-rumour, non-rumour, and unverified-rumour by some scholars. All these different definitions mean that rumour can be treated according to its various definitions [17].

In the next section, fact-checking problems are discussed, together with their potential solutions, whether automatic or manual.

2.1.2 Fact-checking

Recent reference has been made to the problem of published false information. Several automated fact-checking systems [18–21] have been developed and used in real-world scenarios, such as in the monitoring of false claims during the primary and general election debates throughout the 2016 US elections. A claim is first checked by collecting supporting or opposing evidence from knowledge bases and the Web, thereby generating questions/queries related to the claim and extracting a final answer. This is then presented to the user, based on discrepancies between the returned answers and the claim.

Hassan et al. [20] refer to half-truths, hyperbole, and falsehoods, while Thorne and Vlachos [22] mention that false information can now be very easily disseminated to millions of people. Mihaylova et al. [23] examined the issue of potentially false information, specifically in relation to community question-and-answer (Q&A) forums. The above authors [23] devised a system that checked the validity of answers to specific questions, similar to the approach described in [24]. The system was proposed as a solution to a hitherto neglected issue. The novel contribution of their work was a multi-faceted model, which captured the diverse information that can be gleaned from the answers posted in these forums, not only comprising what is said, but also how, where, and by whom. In [24, 25], the contribution involved checking the consistency of a set of claims that were not explicitly mentioned in the text in reference to the knowledge base. This method combines the use of information extraction (IE) techniques with probabilistic reasoning, allowing for inferences to be made, starting with natural text. A different approach to fact-checking was adopted by Rashkin et. al. [26], who determined the presence or otherwise of fake information by comparing the language of real news with that of satire, propaganda, and hoaxes. Meanwhile, Rashkin et al. [26] used a corpus from PolitiFact, arguing that while fact-checking remains an open research topic, the truthfulness of a text can be determined from stylistic cues in the language. A similar approach, also utilising the properties of the language itself, was adopted by Nakashole and Mitchell [27], who deployed linguistic features such as those represented in a subjectivity lexicon of strong and weak subjective words, and a sentiment lexicon of positive and negative words – these being bias lexicons derived from Wikipedia. In addition, parts-of-speech (POS) tags were used to establish whether a source of information represented facts objectively, or was opinionated and/or speculative in nature.

2.1.2.1 Fact-checking Using Various Techniques

In response to the fact that over the past few years, there appears to have been an increase in the quantity of false information published on the Internet. Mohtarami et al. [28] presented an automatic false information detection system, using end-to-end memory networks. This system can check whether a particular statement agrees, disagrees, discusses, or is unrelated to a target claim. Mohtarami et al. employed convolutional and recurrent neural networks, as well as a similarity matrix. They then applied the Fake News Challenge benchmark to test their proposed system, which could be considered as feature-light. Nevertheless, the system's performance was found to be equal to that of more complex systems. Furthermore, automatic fact-checking was considered by Vlachos and Riedel [29] as a process that must include the identification of check-worthy statements, while Hassan et al. [19, 20] considered the creation of questions relating to these statements, with a focus on finding information that was pertinent to the construction of a knowledge base (also relating to these statements), followed by inferring the validity of the check-worthy statements. Conversely, Thorne et al. [30] investigated automatic fact-checking, using surface-level linguistic patterns. Specifically, Thorne et al. [30] adopted a hybrid convolutional neural network, which integrated text with metadata, arguing that it would improve text-only (rather than knowledge-focused) deep learning. In contrast, Popat et al. [31] produced the CredEye user interface for fact-checking, where the user enters a claim that needs to be checked, and the output is the probability of it being true or false, conditional on the information obtained from a Web search focused on this claim. CredEye also checks the trustworthiness of information sources and gives evidence in the form of a screenshot of texts containing the relevant material. Nevertheless, Thorne and Vlachos [22] noted that the interdisciplinary nature of fact-checking research has resulted in terminological inconsistencies. In order to respond to this issue, Thorne and Vlachos [32] surveyed the various research efforts concerning automated fact-checking, and based on natural language processing.

Conversely, Hassan et al. [20] centred their fact-checking approach on an analysis of the meanings and characteristics of natural language sentences. Thus, they introduced ClaimBuster: a system involving the analysis of natural language statements within a particular political discourse for the purpose of checking facts. This system works by first classifying and ranking sentences into non-factual, insignificantly factual, and check-worthy sentences. The latter are then appropriately labelled by human coders, after which, feature extraction is applied to produce a training dataset [33]. However, ClaimBuster is limited by discrepancies between the classifications determined by the human checkers employed, and the classifications generated by the software system. Similar to the present study, however, Karadzhov et al. [34] presented a fact-checking system in which the claims must be verified via information that needs to have been constructed or identified outside the system. The verification of claims is also the focus of the method adopted by Baly et al. [35], who developed an approach that involves determining the stance of a text, with regard to a claim, and then determining whether the claim is factual on this basis.

In response to the problems that are associated with using traditional methods for fact-checking the ever-increasing volume of information published online, Ciampaglia et al. [36] offered a computational approach that uses knowledge graphs which represent semantic proximity from transitive closure, exploiting the smallest path between two nodes. Thus, their approach leverages existing bodies of expert knowledge to assess the truth of statements. This is similar to our approach, which is also based on known truths. Another related approach is Shi and Weninger's [37], which uses knowledge graphs that incorporate predicate interactions and connectivity. In the present study, we use entities and the relationships between them in the fact-checking processes. Again, similarly, Ciampaglia et al. [36] and Shi and Weninger [37] both used statements of fact in the form of a subject, a predicate, and an object, whereby there is a stated relationship between them. A different approach to fact-checking has been proposed by Popat et al. 2016 [12] which, although using a domain setting to check the credibility of claims made in natural language, also employed inferences based on a joint interaction between the kind of language employed by the claim and the reliability of the web source. The same authors extended their work [38] to include the claim's temporal footprint on the web; this was shown to be effective for the early detection of emerging claims.

The fact checking of numerical claims has also been studied in recent times. For example, Vlachos and Riedel [39] focused on the fact-checking of simple numerical claims, such as the 'population of Germany in 2015 was 80 million'. The above authors used distant supervision to identify and verify claims, in order to fact-check 16 numerical properties of countries (for example, population). Input claims were matched with entries in a knowledge base, and verdicts were deduced. In the follow-up work, the system was extended to include temporal expressions, so that the temporal context of the claim could be taken into account [32].

Recent work has adopted MLNs to apply reason to the world under conditions of uncertainty, answering questions such as 'According to sources A and B, is Mr. Doe euro-sceptic?' [40, 41]. The algorithms in question support the task of extracting information about the facts from various sources, and fact-checking claims against background data, although this was not tested on real-world data. On the other hand, work by Patwari et al. [42] discusses a system of identifying check-worthy statements in political debates, where these need to be fact-checked using a multi-classifier system that models latent groupings in data. These statements might not be explicitly mentioned in the text, but they are check-worthy. From the statement, 'We need the private sector's help, because government is not innovating', Patwari et al. identified a check-worthy claim such as 'the U.S. government is not innovating'. Natural language summaries of relational databases have also been semi-automatically fact-checked using probabilistic modelling to identify erroneous claims in articles from major newspapers [43]. Nevertheless, the limitation of this work is that it requires humans to check the interpretations of the system and correct them if they are wrong.

In contrast, there are claims checked in this current study, which are not explicitly stated in the text corpus. Using a knowledge base of extracted facts from various sources and first-order logic rules, information is inferred that is implicit in the text. The focus is on detecting claims that might be considered implausible because they implicitly contradict background knowledge, assumptions, or other claims contained within a reference corpus.

Here, an approach is proposed to search for an interpretation of language that is consistent with a given knowledge base. The focus is on looking for interpretations that are highly consistent, or which are maximally consistent with the knowledge base, given a set of rules. Verification of the consistency of statements is performed using PSL to conduct the necessary logical inference [24, 25]. This method begins with the extraction of relations and constants from texts, using an IE tool that can work across any domain to build a trusted knowledge base. Once this is built, the consistency of given statements can be checked against the knowledge base, according to pre-specified rules. In addition, there is a re-querying mechanism that enables the knowledge base to be expanded to achieve continuous learning. This research uses PSL as a rule-based probabilistic framework for inferring facts, which are not explicitly mentioned in the text, and to check their consistency with a trusted knowledge base. This paper shows how PSL can be used to assess the consistency of one fact with another.

2.2 Information Extraction (IE)

An IE system generates the kind of structured data that is essential for various applications, searching tools, and Web-mining processes. Information extraction may be defined as the automatic or computerised extraction of events, entities, and concepts (along with their associated attributes and relations) from free and unstructured text. The majority of IE-based systems are expert in recognising the patterns that present semantic, syntactic, and lexical constraints [44]. Furthermore, in the general domain, the application of IE commonly focuses on one or more subtasks, such as named entity recognition (NER). The latter is responsible for identifying named entity mentions and relationship extraction. It discovers the relationship between attributes, entities, and concepts, and performs co-reference resolution: a task that links names or mentions in reference to the same entity.

2.2.1 Information Extraction (IE) Tasks

Information extraction is applied to text documents, in order to create a structured view of information, which is understandable and easy to navigate. For this purpose, three major IE tasks are performed: event extraction, relationship extraction, and fact extraction. Event extraction may be defined as discovering or identifying the various events that are represented in unstructured free text, while simultaneously deriving structured and detailed information in relation to these events [45]. In other words, event extraction extracts multiple entities and identifies the relationship between them. The development of event extraction has been fuelled by the continuous advancements in natural language processing (NLP) and text mining that have taken place over recent years; the availability of annotated datasets, and the increasing popularity of big data [46]. Moreover, event extraction synthesises both experience and knowledge from multiple domains, such as knowledge modelling, AI, data mining, linguistics, and computer science, thereby producing more detailed and complex outputs. These outputs then play a vital role in expanding knowledge bases, monitoring and analysing public affairs for governments, and discovering market responses [47]. Therefore, event extraction is one of the core components of IE, not only extracting corresponding phrases, words and characters, but also detecting a wide variety of events.

In contrast, relationship extraction classifies and detects pre-defined correlations between different entities within a text document. For example, in the case of the names, 'Steve Jobs' and 'Apple', relationship extraction would derive the relationship between the organisation and its founder [45]. The task of relationship extraction can be divided into two simple steps: first, it detects whether an entity of interest lies within a sentence or text, and then classifies the detected entity into a pre-defined group. Several techniques have been proposed for the implementation of relationship extraction. In the most popular of these techniques, the task is considered to be a classification problem [48], whereupon more than one entity occurring in a sentence can be categorised into a single relation type. Put more simply, relation extraction reveals a relationship between attributes, entities, and concepts, while also storing the association between these and the relation [44]. Such attributes or entities and relations could include organisation-location, person-affiliation, and so on. Similarly, [49] explains that relation extraction obtains various instances of semantic relation that are present in texts. This information is then converted into a more readily manipulable format, which is of great use in applications that require semantic knowledge.

The third IE task is fact extraction. Here, logical reasoning is combined with facts to extract fact-representing structures from unstructured text [50]. However, this latter task is prone to various constraints, relating to the trade-off between scalability, recall, and precision. The methods that perform best in terms of scalability tend to become vulnerable to noise in patterns, which lowers the precision of the extracted information. Correspondingly, the application of high-precision techniques, which utilise deep reasoning negatively, affects scalability in relation to Web-scale data. Nevertheless, regardless of its limitations, a fact-extraction system can perform multiple syntactic transformations on free and unstructured texts, in order to obtain factual information. This syntactic transformation utilises English syntactic rules, along with pattern-matching and linguistic technologies (particularly text annotation) to break text into base tokens [51], which are subsequently analysed to extract the factual information. In this research, both relations and fact extraction are used.

2.2.2 How Different Information Extraction (IE) Tasks are Performed

The IE process assists, as its name suggests, with extracting useful pieces of information from semi-structured or unstructured data that is present in different forms, primarily online. Such information will relate, for instance, to events, objects, relations, and entities. The main objective of IE is to provide information to knowledge bases, so that their ability to allow access to and to organise useful information (i.e. knowledge) is improved. This process proceeds by first collecting documents to be used as input, and then by producing diverse representations of such information as will satisfy the information requirement [52]. Information extraction techniques effectively and efficiently examine unstructured text in its unrestricted form, and then extract from it the relevant and valuable information. Thus, the ultimate objective of an IE technique is to recognise the factual details that are likely to enhance the knowledge base or database. Furthermore, [53] argues that by adopting machine learning for IE, practitioners will not only be able to render their IE systems trainable and adaptable, but also useful for reducing manual effort. The most popular of these techniques include NLP and syntactic rules. While NLP examines the grammatical structure of sentences to construct grammar-based rules for extracting useful information, syntactic rules analyse word patterns directly to extract this useful information. In addition, name entity recognition (NER), sentiment analysis, text summary, aspect mining, and topic modelling are a few of the most widely used IE-related techniques. In this research, NER is applied, which will also be discussed in the proceeding sub-section.

2.2.2.1 Name Entity Recognition (NER)

Name entity recognition is one of the most commonly adopted IE techniques. It offers exceptional results in terms of extracting descriptive entities. In particular, NER identifies domain-independent or generic entities, like organisations, persons and locations, by semantically categorising information into pre-distinguished classes [54]. Conventionally, NER techniques have been constructed on learning-based methods (LBM), rule-based methods (RBM), or a hybrid of the two. However, with the introduction of deep-learning technologies like NLP, NER systems based on these have begun to play a significant role in contextual IE and language modelling where semantic analysis, phonetic, syntactic, and morphological linguistics are used. Besides, due to the greater accuracy and efficiency of NER techniques, they have been widely adopted across various domains, including knowledge-base, population, opinion mining, information retrieval, text mining, and machine translation [55]. In this research, RBM is used to extract facts and relations to build the knowledge base.

Name entity recognition is an essential IE method or process that can extract information from both structured and unstructured data. Information extraction techniques that are centred on NER can be further divided into machine learning-based, dictionary-based, and rule-based approaches. Commonly, NER applies rules that are produced according to contextual or grammatical features, simple dependencies, or parts-of-speech tags [56]. Although these systems have over-fitting limitations, they perform excellently in F-measure evaluations. Conversely, the contrasting dictionary-based NER technique is considered as a very effective high-tech approach for large-scale indexing and literature annotation. It not only identifies names, but also distinguishes between unique identity concepts [57]. However, similar to the rule-based approach, dictionary-based approaches also lack the ability to detect new terminologies. To overcome this limitation, machine-learning-based approaches have recently received a great deal of attention. Among the wide range of machine-learning techniques, conditional random fields (CRFs) have become popular, since they integrate the use of multiple features to process sequence labelling.

2.2.3 Evaluation of Information Extraction (IE)

The Message Understanding Conference (MUC) originally encouraged early developments in IE. Moreover, the MUC inspired the development of evaluation metrics for IE systems. Even though these evaluation metrics are similar to the standard precision and recall metrics that are used in information retrieval systems, the definitions have been changed so that they are specific to IE [50]. In the evaluation of IE tasks, precision is interpreted as the amount of authentic and accurate information extracted. In contrast, recall measures the amount of correctly extracted information. In other words, precision signifies the reliability and authenticity of the extracted information, while recall represents the amount of information that is correctly extracted. Likewise, [51] presented three essential structures for IE systems. The first structure consists of the means by which the following are handled: extraction from non-HTML sources, Sarawagi's three-level extraction tasks, and the evaluation of free and unstructured text documents. The second structure should include a taxonomy that is implemented via Prolog-like logic rules, regular expressions, and probabilistic hidden models. Finally, the third structure should incorporate annotation-free, learning-based, or programmer-involved approaches, depending on the degree of automation. The dimensions associated with these structures are first used to evaluate the difficulty of the IE task, followed by making comparisons between the methods adopted in IE, and then, finally, evaluating the efficacy of the training process.

In this account, [58], there is a description of the MUC evaluation process, in which system developers are asked to extract a particular template from a set of documents in a given scenario. The developers are given 1-6 months to ensure that their system has adapted to the scenario. After this period, each system developer is provided with another new document set (the test corpus), using their IE system to extract the required information, and then forwarding the template to the organiser. The organiser proceeds to compare the answer key to the extracted information, and assigns a score for precision and recall. However, comparing two such templates is not straightforward, and multiple methods have been proposed to counter this problem. One such method is the F-score, which is the harmonic mean of the precision and recall [59]. Using the F-score, the performance of an IE system can readily be evaluated.

2.2.4 How Fact-Checking Uses Information Extraction (IE) to Resolve Issues

Of late, fact-checking has grabbed the attention of both the public and the media. In addition, the computer science community has shown keen interest, especially from the IE, NLP, knowledge, and data-management perspective. Fact-checking could be described as the process of examining a piece of information, verifying its credibility, checking its accuracy, and then enriching it with connections, comments, and nuances to determine the correctness and veracity of the factual claims [60]. Due to its ability to combat misinformation and false claims, fact-checking now plays a key role in the news environment. Traditional fact-checking first looks at the factual claim to be checked, and then at the claimant (the person making the claim). A fact-checking process may also analyse the integrity and authenticity of a claim and its source, while providing strong evidence to either counter or support the claim [61]. Despite its significance, the rapid development of IT and parallel growth in the use of digital media has limited the ability of human fact-checkers to keep up with the increased spread of falsehoods and misinformation. Nevertheless, in general terms, new forms of journalism, social media, and the use of the Internet have also made it easier and faster to apply technology to the task of exposing half-truths and misinformation, compared to human fact-checkers [19]. Various computerised IE and data storage tools, as well as NLP, visualisation, querying, and indexing, can all be exploited to assist fact-checkers. Moreover, fact-checking is open-domain in nature. This, along with recent progress in IE and NLP, has provided journalists with powerful tools that they can use to assess the truthfulness of claims.

Traditional fact-checking by journalists lacks the ability to authenticate the vast amount of data that is available online. For this purpose, [36] propose a computational fact-checking technique that is based on the K-nearest neighbours and random forest methods, which produce a knowledge graph. The above study claimed that with this IE technique, it was possible to analyse thousands of claims relating to biographical information, geography, entertainment, and history. Likewise, [62] proposed a framework that used structured data to model claims as parameterised. The latter study stated that satisfactory results were achieved in identifying questionable and vague claims, as a result of the fact-checking tasks that were undertaken. Besides, concerning fact-checking, it has been identified that IE has been of significant assistance to journalists in the extraction of factual claims from both structured and unstructured texts, like articles, speeches, and, for instance, lists of dates. The most widely-used approaches depend on a combination of machine learning and NLP algorithms. These are ideal for identifying and verifying claims [63]. However, even though NLP and machine-learning techniques can recognise multiple versions of a particular statement, they are still vulnerable errors due to paraphrasing. As a result, a potential trade-off between precision and recall exists: having to deal with similar but not identical instances of a factual claim can result in a system with lower accuracy producing a large number of false positives. For this purpose, automatic fact-checking techniques are proposed, which not only reduce the burden on humans in assessing the authenticity of claims, but also discern the reasoning behind a claim [22]. Thus, it could be affirmed that fact-checking is an area of concern that is coming into increasing prominence, and by integrating various IE techniques with each other, a fact-checking process can be produced that will combat misinformation.

Most automated fact-checking methods depend on supervised learning. Many studies have suggested using supervised learning models that analyse and learn from previously annotated statements [29]. This approach was adopted by [64], who proposed the design of a novel, hybrid and convolutional neural network (CNN) for the purpose of detecting fake news. The above study reported that its model achieved accuracy of 0.270, thereby outperforming standard fact-checking algorithms in the hold-out test set. Likewise, [26] presented a fact-checking model based on long-short-term-memory (LSTM). This model used word sequences to generate a feature vector, which was further employed to perform lexical analysis on various types of misinformation. The study concluded that the model achieved greater accuracy than Naïve Bayes, when text alone was used as input. Consequently, it may be seen that the primary limitation of IE techniques is that they usually require extra knowledge of the world (that is, knowledge that is external to the text they are examining), and this extra knowledge is usually absent from a claim [27]. Although language can help establish whether a claim is factual; credible sentences can sometimes (or often) be inherently wrong. To counter this problem, [39] adopted a distantly supervised relation extraction system, which examined the surface patterns in text to establish relationships between different entities on a knowledge graph. The above study stated that the model achieved an accuracy of 60% in identifying false claims. Likewise, [32] proposed a fact-checking system based on claim identification and semantic analysis, using generated knowledge bases and temporal expressions to automate the system. The above study reported that an accuracy of 68% was achieved in relation to simple numerical claims. In light of all these findings, it is evident that many different IE techniques have been adopted to solve the problem of fact-checking, with varying degrees of success.

2.3 Probabilistic Reasoning

Probabilistic reasoning, also known as probability logic, is defined as the ability to handle uncertainty using probability theory, combined with formal arguments. Probability logics attempt to find the probable truth of a situation, but also act as a natural bridge between building and finding truth, with the final result being built through probabilistic expressions. Probabilistic logic statements can be classified into two main groups: one is the class of logics that represent a continuation of logical entailment, as present in MLNs, while the second deals with issues relating to uncertainty and a lack of evidence, wherein the truth of a statement is separated from the trust assigned to this truth [65]. In this section, two probabilistic reasoning frameworks will be defined, as well as how they work. Next, the various applications of the two frameworks will be shown, together with the ways in which they may be used to solve the fact-checking problem.

2.3.1 Markov Logic Networks (MLN)

Richardson and Domingos [66] presented MLNs as a first-order knowledge representation, with a weight for each formula. However, this can also be seen as a framework for constructing Markov networks. Markov logic networks use discrete Markov random field (MRF) [66] graphical models, which are applied to discrete variables with values of 0 or 1. The above authors add that this kind of framework, based on probability theory, has the ability to incorporate a wide range of domain knowledge into the compact language of a Markov network. Conversely, Beltagy et al. [67] present an MLN as a probabilistic logic framework, using first-order logic to decode complex probabilistic graphical models. The idea supporting the application of MLNs is that this allows a level of flexibility, where the model is not required to satisfy all the formulae. Hence, weights are assigned according to the potential levels of satisfaction of each formula [66]. Lee and Wang [68] emphasise that the probability distribution of weights over possible words, and a formula that should be satisfied by a given word, are important in the development of an MLN framework. Therefore, the importance of a word is correlated with the level of satisfaction achieved by the formula, with a weight for each word. All this finally leads to decisions being made about the truth values of statements. Furthermore, Richardson and Domingos [66] affirm that MLNs, in terms of being first-order logic, enable uncertainty to be handled effectively, while also permitting imperfect and contradictory knowledge. Additionally, MLN can support several methods and techniques that are commonly applied in modern statistics and modelling, including clustering, classification, and predictive analysis [66].

2.3.2 Probabilistic Soft Logic (PSL)

Probabilistic soft logic (PSL) is a many-valued logic that uses an inference mechanism based on hinge–loss Markov random fields (HL–MRFs) [65, 69]. Moreover, HL–MRFs represent a graphical model, which is analogous to discrete (MRFs) [66], but applied to continuous variables in the unit interval [0, 1]. Kimmig et al.[69] present a PSL that exists as relational and logical rule definitions within a set, wherein the single entities are viewed as logical atoms. The dependency structure within a set is established through the application of first-order logic rules. This enables a joint probabilistic model to be created over all the atoms, thereby giving a positive weight to each rule in the set. In this manner, the relative importance of each rule is established. The view underlying this methodology is shared by Deng and Wiebe, who affirm that in relation to first-order logic, probabilistic logic with weights for predicates allows each predicate's importance to be established within the inference process [70]. Therefore, this gives a PSL the characteristics of an optimisation model, which can be used effectively to solve optimisation problems [69]. Additionally, the authors state that a PSL can consequently be used to create and establish inference and learning rules. Probabilistic soft logic inference methods are based on providing the most likely values to be given a set of propositions, where the remaining (i.e. non-zero probability) proposition values are presented as evidence in the form of what is known to be the most probable explanation (MPE) or MPE inference [69]. However, Fakhraei et al. [71] maintain that PSL rule weights should be set so that an association is penalised when a rule is not satisfied; again, these weights measure the importance of each rule.

2.3.3 Continuous and Discrete Reasoning

MLN and PSL have been presented as discrete and continuous reasoning frameworks, respectively [65, 66]. Indeed, Kimmig et al. [69] affirm that one of the main features of a PSL is the use of continuous soft truth values in the interval [0, 1]. This is believed to give a PSL greater flexibility over the type of relation that can be established between entities. These relations include belonging to a family or circle of friends and being a classmate. However, these relations are also based on notions of similarity of influences, for example, hobbies, opinions on specific topics, and beliefs. In addition, the above authors reaffirm that PSL is a framework for collective and probabilistic reasoning in relational domains, and it has been applied in a variety of areas, including collective classification [72], drugs [71, 73], opinion diffusion [74], and trust in social networks [75, 76]. Moreover, Huang et al. [76] undertook a study based on social interaction, with the aim of investigating notions such as trust, fondness, and respect. According to the above researchers, trust is the key to social interaction. Therefore, the strength of the tie between people was presented as a weighted graph, portraying their social interaction. Furthermore, Huang et al. maintained that this form of flexible framework is generally well-suited for social interaction and network representatio, using PSL [76].

Tushar et al. [77] used knowledge derived from textbooks to answer science questions, apparently based on scalable knowledge. However, Clark et al. [78] argued that the automatic extraction of scalable knowledge is often associated with incomplete knowledge, with a high level of noise. Markov logic networks are believed to be the best framework for dealing with this type of uncertainty in the knowledge acquisition process. The reasoning approach applied in this study improves on the conventional methods of knowledge acquisition that may be found in textbooks and other text. Therefore, Tushar et al. [77] used three different forms of MLN-based formulation in knowledge acquisition, so as to determine the most effective of these models. The first approach applied rules based on fundamental science as an input into the MLN framework, and deployed the outcome for the purposes of tractability. The second method applied an interpretation of scientific rules, which led to a simpler network. The third technique applied Praline, which enabled more control in the application of MLN. The result of this latter approach was a boost to the framework's accuracy and a tenfold reduction in runtime, as compared to baseline implementation of the same model [77]. In the application, a quality assurance (QA) system was in fact used to handle noisy and incomplete knowledge, so that the reasoning could include uncertainty. However, a difficulty arose with regard to how a QA task could be performed as an MLN reasoning problem. It is important to mention here that Belgaty et al. [79], and Beltagy and Mooney [80] demonstrated MLN as effective for reasoning with rules extracted from natural language. Consequently, Tushar et al. [77] only managed to produce one model (among the three tested) that was more effective than a baseline approach for handling knowledge acquired from text, demonstrating 15

Kimmig et al. [69] reported that a PSL semantics program, represented as a set of first-order logic rules with positive weights, enabled the creation of probabilistic relations between individuals. It predicted their behaviours via certain relationships or links that were established as existing between these entities. For example, voting behaviour was predicted based on the links between two given individuals, A and B. This type of approach also takes into consideration the notion of trust, especially trust within relationships. Therefore, Huang et al. [76] argue that PSL and its soft logic nature is perfectly adapted to non-discrete models, especially representing the strength of social trust. Thus, PSL was used for trust prediction in social networks, where both a structural balance model and social status model were necessary. The authors established that social balance relations and structural balance behave differently, with social balance following the rule of triadic closure, similar to transitive relationships in mathematics. Conversely, Farnadi et al. [81] used a mechanism to infer a set of user characteristics, which worked by mining within a larger source of information, provided by the users. However, it is important to note that the users' status model type was mostly based on the notion of status or reputation. In another study, West et al. [75] investigated the person-to-person evaluation of discourse that represents the key to establishing reputations and social bonds, as well as creating public reputations. In fact, this evaluation can be scrutinised by textual sentiment analysis via social networks. However, it is a form of assessment that lacks the range of variables that should be included, such as the interaction between languages and the social context of the time. In order to predict individuals' opinions of each other, a model was constructed, using information shared between individual I_1 and individual I_2, through techniques such as sentiment analysis and text evaluation. This formulation of individuals' evaluation of each other does lead to some kind of resolution of the matter, but it is a very complex problem and therefore, difficult to address [75]. The above author also mentions that the way in which people evaluate each other can be predicted according to the social networks (a network of contacts) in which they are embedded. The resulting model, which combines information from both text and network structures, proved to be superior to the text-only and network-only versions, in terms of resolving the issue of community-level decision-making [75].

Thus, Farnadi et al. [81] succeeded in constructing a user profile in their model by inferring users' characteristics through the mining of broader sources of user-generated content (UGC). This approach contrasts with the more conventional method of utilising a single source of information. Farnadi et al. were able to build profiles of social media users using UGC and the different social relations that they could establish or which were available. The above-mentioned statistical relational framework employed HL-MRFs. Hence, it was based on first-order logical rules, validated with data from well-known social platforms such as Facebook. It was found to perform better than models where a single source of information was deployed [81]. Farnadi et al. [81] emphasised that a variety of different techniques were used for the profiling of users, in order to infer their gender and age, and build their 'portrait'. This was done for various reasons, including targeted advertisement, but also to improve the user experience on the Web. However, the above authors indicated that due to different platforms collecting different types of information, and different users possibly providing different forms of information on the same platforms, the main challenge was to build a system that would unify these sets of available information to profile a given user. Farnadi et al.'s model was developed under a flexible framework. It can therefore provide user profiles based on several different user traits, including age, gender, use of UGC, social relations, and traits relating to/generated from PSL. Probability soft logic-weighted attributes inputted into first-order logic rules are very effective for modelling a social-relational dataset, with the applications that adopt this form of PSL ranging from sentiment analysis, to social trust propagation and span detection, all in social networks [76, 81, 82]. In this work, the authors combined diverse sources of UGC and social-relational content (SRC). The result of the study shows that except in relation to predicting gender, the most reliable data source is SRC.

In a further study, the researchers combined the advantages of MLNs and PSL to handle natural language semantics with proficiency. The findings supported the argument that textual entailment systems are effective for conducting quality reasoning using logical inferences, but poor at using soft logic inferences [67, 83, 84]. Conversely, distributional semantics-based systems perform well when applying lexical and soft reasoning, but perform poorly when faced with negation or quantifiers. For example, distributed models can use word co-occurrences in texts to identify and predict similarities in semantics [67, 83, 84]. It is important to note that in relation to higher-dimensional spaces, words can be represented by vectors. In this manner, they may be subject to computation when dealing with long phrases and deducing distributional and asymmetric similarity to predict both hypernymy and lexical entailment [67, 85, 86]. The above researchers showed that an MLN could be used as a suitable framework for probabilistic logics that employ weights on first-order logics, in order to 'soften' the rules. Thus, probability distributions can be defined over words, so that words obeying clauses will have exponentially increased probability [66]. Probabilistic soft logic was also presented as a framework that can perform efficient probabilistic inference, where grounded atoms have continuous soft truth values in the interval [0, 1]. In contrast to MLN, PSL defines a probability distribution over a continuous space, with PSL being able to lead to an optimum interpretation, using its most probable explanation (MPE) inference [69].

Bos [87] used a logical representation based on Boxer, a software component that performs semantic analysis by applying combinatory categorial grammar (CCG). Boxer maps input sentences into logical forms, wherein the predicates are words in each sentence [87]. On the other hand, the necessary distributional information is presented as weighted inference rules, which establish a relationship between words and sentences in the input domain. Beltagy et al. [67] defined two vectors from two input sentences, also applying a 'cosiness' estimate to those two vectors and thereby indicating a similarity between the distribution of the noted sim of the vectors. Additionally, Beltagy et al. defined asymmetricity as a difference. However, the recognition of the textual entailment (RTE) method was selected, employing MLN. However, because of issues such as overhead costs and unpredictability, the MLN needed to be adjusted before it could play a practical role in this context. Therefore, extra constants were used to overcome this difficulty. In contrast, PSL was adopted for semantic text similarity (STS) because PSL is particularly effective in computing similarity within structured environments. Nevertheless, [67] demonstrated that the use of MLN, augmented with extra constants, can lead to significantly enhanced performance of this method in terms of semantic text similarity. Conversely, an additive regressive model was built according to an option selected from the permutations created by $E = S_1$ and $Q = S_2$ for each sentence pair: S_1 and S_2. The results of the combined MLN/PSL model, using either similarity or asymmetry with RTE, enabled the researchers to conclude that the asymmetric measure for predicting lexical and textual entailment were of great significance [67]. The authors used Boxer, an item of software that enabled them to map the two sentences. The facilities included CCG, employed as a logical representation of distributed semantic knowledge and to add weight to the MLN's inference rules, thereby determining the entailment probability [87, 88]. The sim measure of the two sentences was obtained by averaging the probability entailment of these sentences. In addition, STS was used to compute the entailment of two textual entailments, deploying a system with a combiner. This proved to be computationally more effective for shorter as opposed to longer sentences [89]. However, the use of short sentences introduced some loss of text structure, but accelerated inference while reducing overall accuracy [89]. Beltagy et al. [79] extended this work but replaced MLN with PSL and used the 'lazy' grounding approach, which avoids an explosion of the knowledge base by refusing to establish irrelevant grounding [89]. Using Pearson's correlation, the results confirmed that PSL offered better probabilistic logic than MLN for measuring semantic similarity, and the importance of probabilistic logic in the analysis of logical information [89].

A similar investigation was conducted on STS, using PSL combined with a distributional framework. Probabilistic soft logic was presented in the study as being computationally less expensive and more accurate than MLN. Indeed, for various reasons, researchers have found PSL to be a more appropriate probabilistic logic than MLN when studying STS; PSL is a framework that was originally conceived and developed to support better inference and scaling on more structured and longer sentences. In addition, PSL was designed more for computing similarity between entities, such as sentences and words, than for determining probabilistic logical entailment. It has even been referred to as probabilistic similarity logic [67, 83, 89]. Beltagy et al. used PSL and MLN to detect semantic text similarity, and MLN to determine entailment probability, demonstrating that the performance of MLN could be improved when augmented by extra constants. Betalgy et al. carried out two related studies, the first of which employed PSL+cosine (sim) and PSL+asym to highlight the importance of asymmetry in semantic similarity. The second study used PSL instead of MLN, subsequently combining PSL with vector addition (PSL+vec-add) and vector multiplication (PSL+vec-mul), both based on Pearson's correlation. This was to demonstrate the importance of probabilistic logic in integrating logic and distributed information [67]. Nevertheless, MLN has always exhibited some limitations in the analysis of long sentences. Hence, the authors set it to employ short sentences (resulting in lower accuracy) [69, 87, 90]. Semantic textual similarity is measured via a judgment on a scale of 0-5. Scores on this scale were yielded by the SemEval function.

Ramesh et al.[91] conducted a study where PSL was used in an attempt to understand students' patterns of interaction with massive open online courses (MOOCs). A general analysis revealed that after mass registration on the course website, whereby students were firmly committed to completing the course, there was a very high drop-out rate, such that very students actually completed the course to attain a grade or certificate. The Coursera online course model, used as a case study, demonstrated the importance of understanding students' interactions with the learning material [91]. Therefore, it was considered crucial to comprehend learner engagement, and observe their online behaviour and levels of interaction with certain activities. These activities included posting on discussion boards, viewing or voting on posts, following course materials, and completing course assessments. The importance of each of these activities was established, and 'OpinionFinder' was used as a tool to label and measure student engagement or disengagement with the online course [91, 92]. This enabled learners to be classified into different categories: passive, active, and disengaged. For example, those who showed less enthusiasm in taking quizzes during the virtual classes (therefore, not 'active') might nevertheless have interacted with the material, such as by following the recorded lectures (therefore, 'passive'). In addition, the researchers tried to correlate this notion of engagement with students' levels of performance and achievement in terms of the course received online. Two PSLs were adopted to model the students' behaviour. The results showed that the PSL model where latent engagement was applied, predicted student performance more accurately than the simple PSL model, which only employed receiver operator criteria (ROC) and area under the curve (AUC) [91] .

Nonetheless, Deng and Wiebe [70] claim that when using PSL, it is important to consider users' sentiments, which can go some way towards explaining students' engagement with an online course. Indeed, a parallel could be drawn with their work, where the PSL applied logical rules, and developed and integrated the students' sentiments, which would ultimately either positively or negatively affect the students. Probabilistic soft logic is known to be efficient in inferring and adopting probabilistic models that are specific at the first-order logic. Finally, in the above-mentioned work, the authors successfully achieved various aims, first detecting explicit or implicit sentiments in the entity-entity or entity-events relations; second, inferring sentiments by exploiting first-order logic rules, and third, jointly resolving both explicit and implicit sentiment ambiguities [70]. Deng and Wiebe then further developed their study, defining eTarget as an electronic target that would enable the author to identify an aim, such as understanding students' level of interaction. In addition, eTarget would allow the introduction of rules to help aggregate opinions into positive and negative pairs, without the need to apply the reference. Finally, Deng and Wiebe defined a PSL model that was combined with the +/− effects events, as well as building three different methods for eTarget candidates, and three PSL models for sentiment analysis [70]. In addition, a value was assigned to the ground atoms, using a support vector machine (SVM) classifier [70]. The result showed that SVM could not achieve good ranking performance in measuring the importance of eTarget, while the PSL model was found to be more efficient in the same ranking test. This could be explained as SVM using what could be regarded as a hard constraint in the selection of eTarget candidates, compared to PSL, which uses score values as a soft constraint. Furthermore, Deng and Wiebe [70] supported that the calculation of accuracy also favours PSL, which can identify true negative eTarget candidates and remove them from the set, and thus, from the calculus.

In contrast, Samadi et al. [93] built the ClaimEval system, which adopts an integrated approach, wherein pro and con arguments from the Web are taken into account, in conjunction with PSL, to validate a given claim. This method could be considered as a more flexible approach, allowing the incorporation of different forms of prior knowledge [93]. ClaimEval is proposed as an improved form of credibility assessment. It operates with a proposed claim, C_1, which has already been verified as true by the user. The pro and con arguments are then extracted from the Web and become associated with the credibility of the domain. In turn, this enables the C_2 claim to be evaluated. Nevertheless, information credibility is far too subjective a notion. For example, an item of information that is presented a credible by Person A might not be credible to Person B. It demonstrates another level of difficulty in this evaluation process. To overcome this limitation, ClaimEval uses as input a prior credibility assessment knowledge set. In addition, it adopts PSL to add more flexibility. The PSL application possesses the following properties: ease of use of prior knowledge, interpretable credibility scores, and guaranteed convergence [69, 93]. On the other hand, constructing the credibility assessment is more about extracting a set of objects of evidence and classifying as either pros or cons. This is the ultimate objective of ClaimEval, which primarily involves examining a set of claims, using relevant sources of evidence. It classifies these claims as either pros or cons [93]. ClaimEval is similar to the approach adopted in this current study, in that it uses PSL to fact-check problems. However, the difference lies in the fact that the current approach checks claims that are not explicitly mentioned in the text, whereas for ClaimEval, the claims need to be explicit in the text, in order to be checked.

After showing how logical reasoning can solve various problems and how it was adopted in this current thesis to help check the consistency of claims, another AI topic is explored, namely, explainable AI to promote trust in machine-learning algorithm decisions.

2.4 eXplainable Artificial Intelligence (XAI)

Artificial Intelligence is used as the basis for a number of applications that have become an integral part of everyday life, such as image processing, facial recognition, and machine-learning powered predictive analysis. These are applied across various industries like healthcare, education, retail, finance, and construction. The origins of AI can be traced back many decades, and the paramount value of empowering smart machines with intelligence, reasoning, and adaptation capabilities is now widely accepted. However, there is little transparency in terms of how AI systems make decisions, and how these decisions are implemented across different fields. Explainable AI represents an attempt to address this issue and shed light on the entire machine-learning pipeline: i.e. beyond the point of algorithm development [94]. A systematic review of the literature was conducted with a focus on analysing the role of explainable AI in fact-checking.

2.4.1 What is eXplainable Artificial Intelligence (XAI)?

XAI may be considered as an AI system with actions that can be easily interpreted by humans. Additionally, as stated by [95], it involves dealing with interpretable machine-learning models, the predictions and behaviours of which are understandable to humans. Interestingly, it also contrasts very strongly with the black-box concept in machine learning, where even the programmers are incapable of explaining the reasoning behind a model's decision. Moreover, it is worth mentioning that there is a difference between generating explanations and making these explanations readable for humans. As stated by [96], humans are significantly bad at explaining their actions, particularly where experience and skill are involved.

Contrarily, in some ways, XAI generates AI agents that utilise behavioural cloning effectively to explain their actions/decisions in a human-readable format. Despite their increasing adoption, machine-learning models are still considered as black boxes (i.e. a model that makes little use of prior knowledge or physical insights, but is known to have been successful in the past) [97]. This is due to a lack of transparency in the steps that these models perform to make decisions [98]. However, it is very important to understand the reasons behind a model's forecasts/predictions, so that the user will trust the system. This is especially crucial when someone is planning to take action in response to making an AI prediction, or when deciding to implement a new model. Solutions to the issue of trusting the accuracy of a validation set have been proposed by [99], who was mainly concerned with adopting local interpretable model-agnostic explanations (LIMEs).

In a paper-based survey on XAI [100], the researchers argued that this type of AI is essential for providing information that will justify to users the potentially unexpected decisions of machine-learning algorithms. The presence of explainable AI systems also means that a proven and auditable method exists for defending algorithmic decisions as ethical and fair. This fosters trust while ensuring compliance with legislation, such as the 'right to explanation', as per the requirements of the General Data Protection Regulation (GDPR) [100]. Explainable models can provide significant benefits across various domains. For instance, due to misclassification problems, a self-driving car can suddenly act abnormally, posing a serious threat to the passengers' lives. In such cases, only an explainable system can elaborate on the behaviour of a self-driving vehicle. A system of this nature can clarify ambiguous circumstances, quite possibly preventing the occurrence of future life-threatening incidents [100]. Likewise, in medical diagnosis, it would be highly irresponsible to blindly place trust in the predictions of black-box systems. Instead, each decision must be easily accessible to a human expert for appropriate validation [101]. Explainable AI systems are crucial in this domain, for example, AI systems that are trained to predict the risk of disease, with a high possibility of drawing the wrong conclusion. In fact, [102] cite an example of this phenomenon, where an AI system arrived at wrong conclusions regarding pneumonia risks. As the use of this machine-learning system was not reduced, it increased pneumonia-related deaths overall. In this thesis, an attempt is made to explain decisions generated through an approach to checking consistency for specific claims. This approach is interpretable by design, using theological reasoning frameworks.

Thus, it is evident from the literature that explainable AI is gaining widespread recognition, and the machine-learning literature contains a long and continuous history of explanatory work, providing researchers with a pool of ideas that can be used to address explanatory (or justification) issues that involve machine-learning algorithms. However, various challenges remain in relation to explainable AI, such as a lack of agreement on vocabulary and standard terminology. For instance, feature relevance and feature importance are interpreted as the same concepts in explainable AI. In contrast, they are treated as two different concepts when models are built in Python or any other programming language [103]. Additionally, it is difficult to establish objective metrics for what constitutes a 'good' explanation (i.e. interpretation of model results). From the social science perspective, such systems would need to be developed and expanded to have effective and efficient explanatory agents, thereby enabling researchers and experts in explainable AI to collaborate closely with researchers from other fields, such as cognitive science, human-computer interaction, psychology, and philosophy [104].

2.4.2 eXplainable Artificial Intelligence (XAI) and Fact-checking

Fact-checking has become an increasingly critical social issue and many technical problems (such as in credibility scoring) arise when attempts are made to automate it. Previous approaches, such as credibility analysis using subject-predicate-object statement grammar, have provided black-box solutions. However, researchers have accepted that there is a need to clearly explain the reasons behind classifying a particular statement as either true or false. Such issues have been addressed in CredEye [31] by taking natural language statements — provided by users — into consideration and using them to create a system of automatic credibility assessment. This system takes a natural language argument as user feedback, and analyses its credibility automatically in a consideration of related Web posts. The various stages involved are as follows: retrieval of articles, stance detection, content analysis, credibility aggregation, and evidence extraction. The system operates by considering language style as a key element in the evaluation of articles [31]. CredEye successfully addresses the limitations of similar systems (such as ClaimVerif and ClaimBuster), specifically, their failure to consider the language style of articles, and the mere listing of claim-related online articles without recourse to evaluation scores for contextualisation. CredEye applies a snapshot of the articles that support or reject a claim as evidence for decisions. For the fact-checking of explanation, the approach adopted in this thesis is to apply logical rules and the corresponding knowledge base, rather than explicitly taking snapshots of text in articles.

XAI is undoubtedly important for fact-checking. However, this line of research potentially faces two major obstacles. First, according to [63], while NLP algorithms can generally capture close statement variants, they are still vulnerable when presented with paraphrasing. Thus, there is a trade-off between precision and recall; finding more claim instances significantly lowers accuracy, meaning high numbers of false positives. Second, any slight change in the context, timing, or wording of a statement can confuse the algorithms. It is therefore evident that these issues could create significant problems when analysing statements. Hence, fully automatic text verification using explainable AI is a very limited research domain.

2.5 Summary

This chapter was introduced by presenting the key reasons for conducting this research, surrounding the investigation of fact-checking in NLP. It was established that this is a question of trust in online information, and consequently, the level of credibility that should be assigned to that information, as outlined in the sections summarised below. 1) First, the problem of truth discovery was defined and discussed. 2) The extant research on fact-checking was then reviewed, starting with the generalities of the subject and followed by the various techniques and frameworks that support the investigation of fact-checking. 3) Next, the prior research on IE was introduced, presented the generality on this topic. Other studies of specific relevance to the present research were subsequently examined, such as research on IE tasks, how they are performed, and the usefulness and importance of the information items extracted in this process. Name entity recognition and sentiment analysis were also explored and highlighted as some of the most widely used forms of IE. In addition, many other techniques were dissected, such as unsupervised learning techniques, while also analysing relevant research studies on the topic of IE applications. 4) In this fourth section, prior research on reasoning and inference was discussed, especially studies evolving around MLN and PSL. Extensive reviews were performed of these two frameworks, which constitute the main reasoning frameworks that support this current study. 5) Finally, research on XAI was highlighted as an alternative to the opacity that surrounds AI decision-making methods, before closing the chapter with a summary.

CHAPTER 3

Research Methodology

3.1 Introduction

Two main approaches are generally considered when determining a research strategy: qualitative and quantitative. The qualitative research approach focuses on the fundamentals of a phenomenon, endeavouring to 'understand' it through experiences and logical explanation. The objective is to express the experience with limited or even no numerical data [105]. In contrast, the quantitative approach is based on empiricism, reliable measurements that control the experimental processes, and the way in which the experiment can be replicated [106, 107]. In terms of research that focuses on software development, a quantitative approach is identified as an attempt to estimate the performance achieved by the created algorithm(s) [108–110].

One well-known strategy is referred to as the combined quantitative, experimental, and synthetic strategy. This strategy involves developing software that will be the subject of a series of experiments. The results are then evaluated on a fundamentally quantitative basis, together with certain qualitative aspects [111]. This combined approach was adopted in the current study. More details on this point are provided in the section entitled: 'The Notion of the FACT Approach'.

In this chapter, fact automated consistency testing (FACT) is described [24, 25], this being a novel and integrated framework that was designed to take a set of claims and check their authenticity. The main sections comprised in this chapter provide: 1) an overview of the FACT approach, wherein the ideas behind the approach are reviewed; 2) a description of the structure of the FACT approach; 3) clarification of the development of the algorithm on which the FACT approach is based; 4) a description of the way in which the FACT approach was validated using well-known statistical methods, and finally, 5) a summary of the chapter.

3.2 The Fact Automated Consistency Testing (FACT) Approach

The FACT approach [25] principally employs information extraction and probabilistic reasoning to check the authenticity of information. Applying these paradigms results in a flexible, concept-based approach, which makes it easy to use and incorporates various forms of prior knowledge. In order to perform this checking process, a knowledge base is included as a point of reference. This knowledge base is built from the facts that have been gathered from various Web sources, based on previous attempts (using the same approach) to check claims. For experimental purposes, a set of claims in the form of natural language statements were created using a variety of sources. These claims were then checked for consistency via the proposed approach. The results of this process constitute a level of truthfulness that was returned for every claim checked.

3.2.1 The Notion of the FACT Approach

Fact Automated Consistency Testing (FACT) has already been defined in two previous papers by the current author, published in the Machine Learning and Knowledge Extraction journal and the Advances in Intelligent Data Analysis XVII Conference Proceedings [24, 25] as **Fact Automated Consistency Testing**. However, each word represented in the acronym carries a different weight in the present research. These weightings are described below.

First, the word *fact* relates to the notion of a claim in natural language, usually consisting of a sentence describing a relationship between two entities: for example, *London is a city in the UK*. This sentence expresses the connection between London and the UK. Second, the word *automated* conveys automatic assignment of a relative truth value to a given claim. Third, this relative truth value is estimated based on the *consistency* of the claim with a trusted knowledge base, calculated using probabilistic reasoning techniques. Therefore, the truth value reveals the claim's degree of coherence or compatibility with the knowledge base. Hence, the knowledge base represents the actuality of the investigated claim. Finally, the idea of *testing* is fundamental, as the entire approach aims to test the level of a claim's truthfulness. Therefore, the FACT approach is well structured, meets several appropriate criteria, and follows certain relevant rationales. This makes it highly suitable to be the subject of this research methodology and part of a clear and well-structured research strategy.

Accordingly, the approach adopted in this study represents a mix of quantitative and qualitative methods. It is constrained to a number of criteria relating to quantitative measures, which include experiments to measure the detailed operations of a phenomenon, such as detecting the level of a claim's credibility using algorithms and a knowledge base. Conversely, this study also applies a qualitative approach, using Java Annotation Patterns Engine (JAPE) grammar to create patterns that build relations from annotated text. Furthermore, claims are used to explain the consistency of decisions concerning claims, in order to evaluate the authenticity of those claims, which likewise corresponds to a qualitative approach. An example of a claim and explanation would be ‘Prince William and Prince Harry are siblings because Prince Charles is their parent’.

3.2.2 Components of the FACT Approach

The pipeline of the FACT approach is made up of several components, which are explained in the following sub-section.

- **Claims:** The primary input in the approach is the ‘claim’, referred to as C. In terms of natural language-processing to check facts, claims are statements, and their consistency is checked using a trusted knowledge base. This process is built into the FACT approach.

- **Keywords:** Referred to as K, this is a list of words that are significant for fact-checking. Keywords are employed in the FACT approach to index and search a relevant set of documents. With respect to the FACT approach, keywords are considered as entities that are extracted from claims, and then employed as search query words. Therefore, they are utilised by the search engine.

- **Web pages:** In this study, Web pages are pages that are found on the World Wide Web and contain the information that is relevant to a claim. They are referred to in this study as L.

- **Articles:** Called A in this study, articles are accessed by scraping the pages accessed via Web links, found as a result of a query related to a claim. These articles represent one of the key elements of extracting constants and relations in the process of building a knowledge base.

- **Relations:** A relation, in this context, can be defined as an existing connection or important association between items. Relations can exist in the form of unary or binary predicates, and can be extracted by employing JAPE grammar, as discussed in more detail in the next chapter (Chapter 4). The JAPE grammar processor is a tool that can be used for pattern extraction and annotation.

- **Constants:** In this study, a constant is a string that denotes an element in the universe upon which a PSL and MLN framework have been grounded. Moreover, in the natural language universe, constants are named entities – more specifically, person entities that have been extracted using a named entity recognition system and disambiguation. For example, in this work, ‘Princess Elizabeth’ will be considered as a constant.

- **Number of constants:** This is the size of the list of entities to be considered after removing duplication. This number increases as a result of querying and re-querying the Web and the incremental construction of the knowledge base, subsequent to checking claims.

- **Number of relations:** This refers to the size of the dataset containing the extracted relations after removing any duplicates. This number increases as a result of querying and re-querying the Web and the consequent incremental construction of the knowledge base.

- **Size of the Knowledge Base:** This is the total number of unique relations and unique constants extracted.

- **Predicate:** In this study, a predicate is said to be a particular relation, defined by a unique identifier and a positive integer, known as its arity [65, 112]. Only binary predicates are considered here, for example, siblings/2, which is a predicate that takes two arguments. This is discussed in more detail in Chapter 5.

- **An atom:** . In this study, an atom is defined as a predicate. This is combined with a sequence of terms. The length of this sequence is equal to the predicate's arity, wherein each term may be a variable, for example, siblings(x,y) [65, 112].

- **Ground atom:** In this study, a ground atom is an atom wherein the arguments are all constants, for example, siblings(Prince Charles, Princess Anne) [65, 112]. Here, siblings is the ground atom, while Prince Charles and Princess Anne are the arguments.

- **Herbrand base (HB):** This is the group containing all the ground atoms plus their negation. Therefore, for each ground atom included, there is also its literal opposite. For example, 'not siblings' is the negative ground atom of "siblings's".

- **Logical rules or clauses:** Logical rules, or R, are disjunctive clauses, which only involve literals. These may be either weighted or unweighted, for example,
  $1.0 : Parent(X,B) \wedge Parent(X,A) \Rightarrow Siblings(A,B)$

- **Ground logical rules:**These are the expressions used to connect ground atoms, using logical operators, for example,
  $1.0 : Parent(Elizabeth\,II, Princess\,Anne) \wedge Parent(Elizabeth\,II,$
  $Prince\,Charles) \Rightarrow Siblings(Prince\,Charles, Princess\,Anne)$

- **Interpretation:** Interpretation is performed with regard to assigning truth values to each ground atom. More details of this are given in Chapter 5, which explains the MLN and PSL algorithms.

- **Consistency:** In this study, interest is vested in the interpretation of the claim, subsequently checking its consistency based on the knowledge base.

3.3 Structure of the FACT Approach

This section describes the structure of the FACT approach, which is best presented as a framework pipeline. The pipeline in question was built to check the consistency of claims against the trusted facts represented in the knowledge base. The basic action of the pipeline is to allow the input of a claim about a relation between two entities, and then to generate a corpus by querying the Web, specifically with respect to that claim. Subsequently, some widely used information extraction tools are employed to extract named entities (constants) and the relations between them from the text corpus. This process is responsible for both checking the claim in hand and the incremental construction of the knowledge base. Thus, relevant constants and relations are added to the knowledge base, thereby enabling the claim's consistency with the knowledge base to be checked, using a probabilistic reasoning model. In its construction, the approach assumes that the claims are concerned with binary (Boolean) relations and are written in natural language, which can be parsed with an NLP tool. The results of this process consist of the relations required to search (across the Web) and extract, using a parsing tool. The experiments are described in detail in Chapter 4, for building the knowledge base, and Chapter 5 for checking consistency with probabilistic logic. The pipeline was implemented via the extant tools, wherever possible. An example of the pipeline input and output is shown below:

Claim: Prince William is the father of Prince George.
Assessment: 0.9 consistent with the knowledge base.

The following Figure (Figure 3.1) gives an overview of the FACT approach pipeline. Each stage of the pipeline is described in detail below. In order to clarify this further, we will present, in turn, the coupled stages of the pipeline.

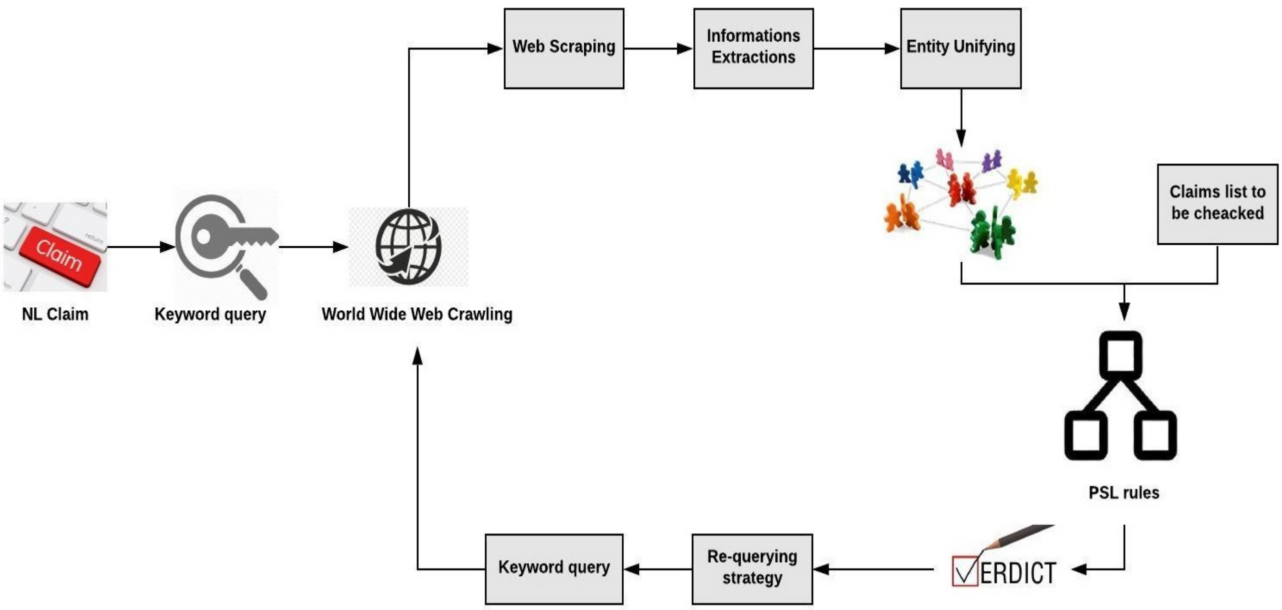


Figure 3.1: The Fact Automated Consistency Testing (FACT) pipeline

3.3.1 Text Collection: Web Query for a Claim in Natural Language

The process is initiated by inputting a set of claims (C), written in English natural language text. One claim is processed at a time, until all the claims have been reviewed. From each of these claims $c_i \in C$, two named entities are extracted, which can be of any recognised type, including persons, organisations, locations, etc. These two named entities are then used to compile the search keywords (K). Here, it is important to query the Web, which begins by crawling the Web to look for links relating to the search query. Finally, the Web pages indicated by the links are scraped with a Web scraper: a tool that returns the relevant text on a page in a consistent format. This tool facilitates the building of the corpus and subsequently, the knowledge base. An example of a claim is shown below:

$$Prince\ Edward\ is\ the\ child\ of\ Elizabeth\ II.$$

From this claim, the approach will extract two entities of the 'person' type, using the General Architecture for Text Engineering (GATE) toolkit [113] and its A Nearly-New Information Extraction (ANNIE) plugin. The example illustrated below will help clarify these steps. Concerning the above claim, the two relevant entities are:

$$Prince\ Edward$$
$$Elizabeth\ II$$

The Web-crawling tool used for the subsequent process is Jsoup, which is a Java library that contains facilities for enabling work with real-world Hypertext Markup Language (HTML) [1]. This is considered to be an effective and efficient application programming interface (API) for URL-based extraction and data manipulation.

Furthermore, it should be noted that each Web page is built using a text-based mark-up language (HTML or XHTML). Hypertext Markup Language is the markup language for documents designed to be shown on a Web page. An HTML document consists of a number of main parts, each introduced with a sequence of characters, or 'tag', to indicate the data type of the text that follows. In the approach adopted in this current research, the Boilerpipe API [2] is used to scrape the Web. Such a Web-scraping tool will extract relevant articles from the links and then perform some pre-processing of the Web page data to remove noise and yield, thereby 'cleaning' articles for use in building the corpus.

3.3.2 Pre-processing: Constant and Relations Extraction from Texts

In the next stage of the pipeline to be discussed, the knowledge base is built using an IE process that extracts named entities (constants), via its Named Entity Recognition (NER) sub process, and the relations associated with these all from the unstructured text corpus created in the

[1] jsoup: Java HTML Parser,in link
[2] API for the Boilerpipe Java library, in link

previous stage. The knowledge base consists of constants and relations. The constants are named entities without duplicates and the relations consist of relations between any two of these entities. Co-reference, Anaphora, and ambiguity resolutions are all carried out in the IE process, as outlined next. Information extraction is performed on the scraped articles via a number of steps, as follows:

- Pre-processing: This is performed on the data intended for the corpus. Before initiating fact and relation extraction, the pipeline applies a tokeniser, a part-of-speech tagger, and a syntactic parser to the data.

  For this pre-processing, the ANNIE plugin from GATE was used. ANNIE combines the resources of a sentence splitter, tokeniser, part-of-speech tagger, gazetteer, and JAPE transducer [114, 115]. In general, ANNIE adds annotations to text, in order to indicate the positions of the elements that are identified by these processing resources. In brief, ANNIE is a Named Entity Recognition tool that extracts named entities such as persons, organisations, etc. The core processing package resources for this study were:

  - Tokenizer:The tokeniser splits the text into very simple tokens, such as numbers, punctuation marks, and words of different types.
  - Sentence splitter: This consists of a cascade of finite-state transducers, which segment the text into sentences. This processing is required for further tasks, such as part-of-speech analysis, amongst others.
  - Part-of-speech: This adds a tag to each word. Such a tag represents an annotation, specifying the grammatical property/purpose of each word or symbol.
  - Gazetteer list: This is a look-up list of entities that are stored across various files. The GATE Gazetteer can use this list to help detect the roles and types of words for the initial annotation phases.
  - JAPE Grammar rule: : The JAPE grammar processor, which is a component of ANNIE, can be employed to help perform a variety of tasks. In this case, it finds or annotates named entities.

- Co-reference Resolution: The text of each article is processed for named entity co-reference resolution. This is a process that will determine whether two distinct natural language expressions in a text actually refer to the same entity in the real world [114, 116]. Through this process, a list of entities without duplication is created. To perform this task, the Orthomatcher module of the ANNIE plugin in the GATE toolkit [114, 116] is applied.

- Anaphora Resolution: Once the co-reference has been resolved, the pronominal resolution module in ANNIE is employed to perform anaphora resolution. The system can resolve any pronoun that presents itself in any form that is recognised in GATE.

- AIDA resolution: Accurate Online Disambiguation of Entities (AIDA) framework is used as an online solution, which offers co-reference resolution and disambiguated named entities in texts and tables [117], demonstrating high accuracy.

- Grammar rule: The JAPE grammar processor was applied in this study to extract patterns. This grammar processing is carried out across a set of phases, each of which employs a specific set of pattern rules. The phases are executed sequentially, and the whole constitutes a cascade of finite state transducers over annotations (the latter are the results produced by ANNIE). In a JAPE grammar rule, the left-hand side (LHS) consists of a description of an annotation pattern, which must be found in the file for the rule to be triggered. Meanwhile, the right-hand side (RHS) of each rule consists of a set of annotation manipulation statements. Annotations (for example, representing persons, organisations, etc.), matched by the LHS of a rule, may then be referred to on the RHS by means of labels attached to the pattern's elements [118]. Once an LHS has been matched, a new annotation may be added to the file by the RHS, for example, [119]:
  LHS (regular expression for an annotation pattern), i.e. this pattern is looking for the annotation, 'teacher'. The RHS (manipulation of the annotation pattern from the LHS) will, for instance, find the teacher's gender and include it in a further annotation.

The following Table (Table 3.1) illustrates a simplified example of applying the Date of Birth JAPE grammar pattern rule (for more detailed JAPE grammar, see [120]), followed by an explanation of the grammar rule, as shown in (Table 3.2). This second Table (Table 3.2) is a line-by-line explanation of the simple grammar rule example, established above (Table 3.1) and providing the meaning of each line involved in the rule's construction.

The following example clarifies the steps outlined above in Figure 3.2. Consider the following section of an article extracted from the BBC News website [3]:
"Prince William is the eldest son of the Prince of Wales and Diana, Princess of Wales, and is second in line to the throne. The Duke was 15 years old when his mother died. He went on to study at St Andrews University, where he met his future wife, Kate Middleton. The couple were married in 2011. On his 21st birthday, he was appointed a Counsellor of State – standing in for the Queen on official occasions. He and his wife had their first child, George, in July 2013, their second, Charlotte, in 2015 and third, Louis, in 2018."

[3] Royal Family tree and line of succession, `https://www.bbc.com/news/uk-23272491`

Table 3.1: 'Date of birth' in JAPE grammar

A JAPE grammar example
Phase: DateOfBirth Input: Person Token Date Split Options: control = brill Rule: Date_Of_Birth Priority: 20 ({Person} {Token}[0,3] {Token.string == "born" ?} {Token}[0,3] {Date}): label → :label.DOB = {rule= "Date_Of_Birth"}

Table 3.2: Explanation of grammar established for 'Date of birth'

Grammar Lines	**Grammar Line explanations**
Phase:DateOfBirth	This is a unique name for the JAPE grammar. The Java Annotation Patterns Engine can process either a single grammar item or a number of such items. Each grammar rule must have a unique notation, for example, 'Date of birth', as used here.
Input: Person Token Date Split	The annotations used in the relevant grammar patterns must be defined here, otherwise the JAPE Grammar will adopt the default annotation. These annotations are: ”Token", ”SpaceToken" and ”Lookup". Any undefined annotations taken up in the grammar will be ignored.
Options: control = brill	Several options can be set to define the pattern-matching rules, for example, the control options, including Appelt, Brill, All, and Once.
Rule: Date_Of_Birth	This is the name of a rule.
Priority: 20	If there are several different patterns for one particular thing, then all the patterns are defined with the highest priority assigned to the most common ones.
(	This starts the LHS.
Person	The pattern matching starts with the finding of a person annotation.

Token[0,3]	This indicates the range of the number of tokens that should be found between a person token and the string "born".
{Token.string == "born" ?}	In this line, the question mark "?" means that the “born” string is optional (may or may not be there).
Token[0,3]	This indicates the range of the numbers of tokens which should be found between the string "born" and a date
Date	The last match is that representing the Date annotation.
): label	The end boundary of the rule’s LHS (representing the pattern that must be found). Whatever follows the selected word (Label) will constitute the RHS of the rule. ‘Label’ is the temporary name used for new annotation patterns.
→	Start of RHS.
:label.DOB = {rule= "Date_Of_Birth"}	The transducer is informed that the temporary name or ‘Label’ annotation will be replaced with ‘DOB’, and the rule that discovers such entities is ‘Date_of_Birth’.

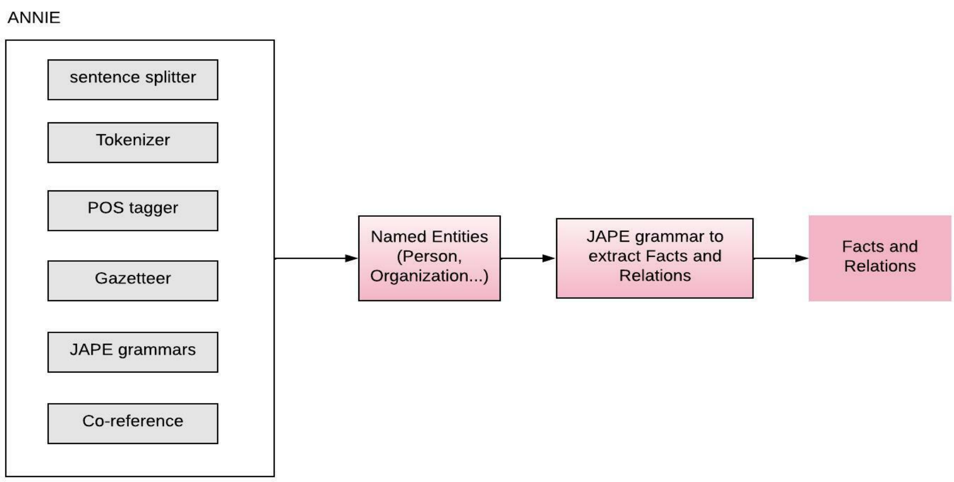


Figure 3.2: Relations and facts extraction pipeline using ANNIE and JAPE grammar

Sentence splitter:
Prince William is the eldest son of the Prince of Wales and Diana, Princess of Wales, and is second in line to the throne.
The Duke was 15 years old when his mother died.
He went on to study at St Andrews University, where he met his future wife, Kate Middleton.

The couple were married in 2011.
On his 21st birthday, he was appointed a Counsellor of State standing in for the Queen on official occasions.
He and his wife had their first child, George, in July 2013, their second, Charlotte, in 2015, and third, Louis, in 2018.
It is important to note that the role of the sentence splitter is to, quite literally, split the text, sentence by sentence

Tokeniser step:
Prince, William, is, the, eldest, son, of, the, Prince, of, Wales, and, Diana, Princess, of, Wales, and, is, second, in, line, to, the, throne, The, Duke, was, 15, years, old, when, his, mother, died, He, went, on, to, study, at, St, Andrews, University, where, he, met, his, future, wife, Kate, Middleton, The, couple, were, married, in, 2011, On, his, 21st, birthday, he, was, appointed, a, Counsellor, of, State, standing, in, for, the, Queen, on, official, occasions, He, and, his, wife, had, their, first, child, George, in, July, 2013, their, second, Charlotte, in, 2015, and, third, Louis, in, 2018.

NB: The tokenizer step consists of separating each word to be tokenized by a comma; henceforth, these words are considered tokens and will be used as such by the next processing resource.

Part-of-speech:
Each token, sentence, or other element of the text is given an appropriate tag. All the available taggers can be found in the GATE Part-of-speech tagger list [4]. The part-of-speech tags are added to each word or symbol, indicating their grammatical properties or roles to ensure greater clarity.

Co-reference and Anaphora resolution:
As mentioned above, co-reference resolution is the process that determines whether or not two distinct natural language expressions found in a text actually refer to the same entity. Meanwhile, anaphora resolution enables the system to correctly process any pronoun in a form that is recognised in GATE. Thus, after co-referencing and anaphora resolution were performed in this study, the article became:

"Prince William is the eldest son of the Prince of Wales and Diana, Princess of Wales, and is second in line to the throne. The Duke was 15 years old when *Prince William* mother died. *Prince William* went on to study at St Andrew's University, where *Prince William* met *Prince William* future wife, Kate Middleton. The couple was married in 2011. On *Prince William* 21st birthday *Prince William* was appointed a Counsellor of State - standing in for the Queen on official occasions. *Prince William* and *Prince William* wife had their first child, George, in July 2013, their second, Charlotte, in 2015 and third, Louis, in 2018."

NB: The changes made to the original text, as a consequence of the co-referencing and anaphora resolution, are highlighted above in italics.

Grammar rule:
Employing the date-of-birth grammar rule, as previously examined, the date of birth was ex-

[4] Gate Part-of-speech tagger list

tracted: "George, in July 2013" was annotated as a DOB.

3.3.3 Post-processing: Named Entities Unifying and Storing the Relations in Knowledge Base

- For each entity in the document, the co-referencing chains (using GATE) are retrieved and set as the local co-referencing chains.
- The features are extracted from the knowledge base. For this extraction, the Ambiverse Natural Language Understanding (AmbiverseNLU) tool [121] was used. The features extracted were of the types: person or location, male or female, Wikipedia link, and name.
- The next process to be performed is similarity-based for each entity, wherein the co-references for each entity and the features from the knowledge base are checked.
- Lastly, the high similarity entities are combined using co-referencing chains, so that the standard names can then be used as the only names for all the entities with high similarity. The standard name is the longest name found for the entity.

Once every instance of each entity is represented, using its standard name, no duplicate names will exist in the texts; this will eliminate all duplicated relations that are extracted.

Lastly, the relations are stored in the knowledge base. The number of times each relation has been extracted from the trawled data is calculated to decide which relations are to be considered as trusted, and therefore fit to be added to the knowledge base. A threshold was set for the number of times that a relation must be repeated for it to become trusted. In this thesis, we set the threshold to to be three. This criterion was combined with restrictions on trusted website selection, in order to determine whether a relationship should be stored in the database, along with the information found in the original article.

3.3.4 Expanding the Knowledge Base by Re-Querying the Web

The last stage in the procedure is to expand the knowledge base. This can be done in such a way that more is learned virtually every time a claim is processed. It should be noted that when new information (relations) regarding a claim is encountered in the text from the Web, it will be added to the knowledge base. After all the user's claims have been processed, the Web can be queried with new queries, specifically constructed for the purpose of further expanding the knowledge base. In relation to this activity, the procedure followed is outlined below. As mentioned above (section 3.3.1), one claim is selected from the set of claims representing the user's input.

Next, a new claim is selected from the set of claims, and the process is repeated iteratively (text collection, pre-processing, post-processing), until all claims have been subjected to this process. Finally, these claims will undergo the different stages of evaluating the consistency of a claim, as described in the next section.

Ultimately, there are a number of strategies that can be used to inform the re-querying of the Web, specifically to expand the knowledge base. One option is to re-query the claims that could not be answered, meaning that those entities/relations belonging to statements that could not be interpreted, due to insufficient information stored in the knowledge base when they were processed, may be used to form the basis of further queries. Another option is to simply re-query all the claims, one by one, until the user's set of claims has been reviewed in full. This second option was selected in this study, given that at the start of the iterative process of checking consistency and thereby expanding the knowledge base, this knowledge base would be quite sparse. Therefore, the Web was re-queried with all the claims that had just been processed, in order to produce a more accurate result for each claim and further expand the knowledge base [122].

3.3.5 Claim Consistency Checking Using Probabilistic Reasoning

The main goal of this proposed approach is to build a system that can check the consistency of any given natural language claim in relation to an iteratively constructed knowledge base. To achieve this, probabilistic soft logic inference (PSL) was used. Probabilistic soft logic inference [69] is a technique that enables developers to specify rich probabilistic models, employing a number of continuous-valued random variables. These models apply first-order logic to describe features that, when combined, define the Markov network type, known as a Markov random field. The statistical relational learning language, Markov Logic Networks (MLN), also performs this role [66]. It is important to note that MLN was likewise used in this study to check for consistency . This is discussed in more detail in Chapter 5.

3.4 FACT Approach Algorithm Design and Implementation

The development of the FACT approach algorithm design and implementation is described in the following two sub-sections.

3.4.1 Introduction

Before designing and implementing the FACT approach algorithm to check for fact consistency, light will first be shed on the methodology adopted to provide definitions of the main acronyms used in these algorithms (algorithm 1). Indeed, the algorithm designed follows a created methodological path. Therefore, it should ultimately be consistent with the algorithmic pipeline illustrated below to support the methodology. Furthermore, these acronyms are detailed as follows: let $C = c_1, c_2, c_3 .....c_{n-1}, c_n$ be a set of claims, where consistency needs to be checked against the trusted knowledge base (KB) created in this study. The KB knowledge base, its construction being reported in more detail in the next chapter (chapter 4), contains relations that were extracted from articles obtained by crawling and scraping Web pages.

Nevertheless, each extracted relation was assigned a truth value (TV), with the function of determining the level of trust to be awarded to that relation within the knowledge base. This truth value belonged to the interval [0, 1], when using PSL, and 1, 0 when using Tuffy as the MLN inference engine – wherein the value of 1 is always assigned to the relation, because the source of the relation is thoroughly investigated before creating the relation. Consequently, this higher level of trust in each relation was traduced by the highest truth value of 1. After building the KB, Tuffy or PSL was run. In turn, this would infer a new relation, to which a new Tuffy or PSL truth value, noted as MLN/PSL_{TV}, would be assigned. Both Tuffy and PSL are explained in more detail in Chapter 5. The Tuffy or PSL truth value was associated with each target or query that corresponded to the claims. The latter was also assigned a certain level of consistency with the KB (observations/evidence) and rules (R). The notion of the truth value was extensively developed and employed in this chapter 5.

3.4.2 FACT Algorithm Design and Details of the Implementation

Algorithm 1: FACT algorithm

Input : Set of Claim C & Logical Rules R

Output : Knowledge Base created KB with Truth Value TV. Infer Relation I with Truth Value MLN/PSL_{TV}.

1 **for** *each* c_i *in* C
2 **do**
3   Run ANNIE for c_i (annotate c_i with named entities)
4   K = Extract keyword (two named entities form c_i) for web Crawling
5   Crawl the web with K for extracting all related links L
6   **for** *each* l_j *in* L
7   **do**
8   A_i = Scraping link l_j
9   Clean A_j
10   Apply AIDA on A_j
11   Annotate A_j using ANNIE
12   Apply Co-referencing and Anaphora resolution for A_j
13   Apply JAPE Grammar for patterns extraction from A_j
14   Store result Fact and Relations extracted from A_j
15   **end**
16   Unify named entities in Fact and Relations
17   Store Unify Relations and the Facts in KB
18 **end**
19 Run Logical Inference Framework (PSL or Tuffy models) using R, C and KB

Algorithm 1 takes as input a set of claims (C) in natural language (English), with a relation between two constants. Lines 3-5 for each given c_i claim will apply the ANNIE plugin in the GATE toolkit to extract the named entities, which are the keywords. These keywords (K) will be two named entities from c_i; they will be used for Web-crawling, leading to the detection of related Web links (L).

In lines 8-12, for each l_j element of L, link l_j will be scraped to extract an article (A_j) that requires cleaning by removing all noise, such as images, advertisements, etc. AIDA is then applied to resolve the disambiguation, followed by annotation of the article, using ANNIE. The next step will be to apply co-referencing and anaphora resolution to the article (A_j).

In lines 13-14, JAPE grammar is applied to perform pattern-extraction from (A_j). In lines 16-17, the named entities are unified in Facts and Relations. These are subsequently stored in the knowledge base (KB). Lastly, in line 19, the logical inference framework is run to check the consistency of the claims (C).

3.5 FACT Approach Validation

The efficacy of the current FACT approach is evaluated based on approved and recognised metrics, and quantitative research methodology techniques. Research validation is of paramount importance for researchers in general. In this regard, Lucko and Rojas [123] argue that it is crucial to validate the research methods and results, in order for a study to be accepted as a scholarly endeavour. Several techniques and metrics exist that are equally effective for validating research results or the research methodology.

A research methodology may be classed as valid according to either the theory, research method, or research approach adopted. For example, use of an experimental method will support a quantitative research approach. This fact will in turn mean that the research methodology is valid for use in a study where a quantitative approach must be adopted. Meanwhile, the research result can be validated through different quantitative techniques, such as the data or sample selection, cross-validation, and so on. In addition, performance metrics may be calculated to validate a research result, for example, precision and recall.

3.5.1 Validation of Relation Extraction

In the field of information extraction, most evaluations are performed with

Precision is the ratio of the total number of correct relations extracted by the pipeline, to the total number of relations extracted; while recall is the ratio of the total number of correctly extracted relations, to the total number of relations in the source text [124].

Precision is calculated as follows (in equation 3.1) :

$$Precision = \frac{Total\ number\ of\ correct\ relation\ retrieved}{Total\ Number\ of\ relation\ retrieved} \tag{3.1}$$

and recall is calculated by (in equation 3.2) :

$$Recall = \frac{Total\ number\ of\ correct\ relation\ retrieved}{Total\ Number\ of\ relation\ the\ document} \tag{3.2}$$

These matrices calculations techniques are described in more detail in the experimental evaluation section in chapter 4.

3.5.2 Validation of Fact Consistency Checking results

To validate the fact-consistency checking, a ground truth dataset was created. The FACT approach was then undertaken, using the above-mentioned dataset. Comparing this with the result of another state-of-art fact-checking system, in relation to the same dataset, would enable the efficacy of the constructed approach to be tested, thereby allowing its validity to be concluded. It should be noted that in this sub-section, precision and recall were used to assess the validity of the approach. In Chapter 5, the validation is described in more detail.

3.6 Summary

In this chapter, FACT was presented as a novel integrated approach, which works by checking the consistency of a set of claims, using a knowledge base as a reference. Initially, the structure of FACT was explained, highlighting how expansion of the knowledge base took place, based on text extraction and processing to achieve incremental construction of the knowledge base. To build the knowledge base, information extraction was required, which is the focus of the next chapter.

CHAPTER 4

INFORMATION EXTRACTION TECHNIQUES FOR KNOWLEDGE-BASED CONSTRUCTION

4.1 Introduction

Information extraction relies on the automatic extraction of information from natural language text documents [125]. Various types of text document may be included, such as free, semi-structured, and structured text. As one of the most crucial processes in the field of text-mining, information extraction consists of two fundamental steps: relation extraction, and named entity recognition (NER). Information extraction processes generally begin with the collection of a number of different texts. Analysis of these texts then produces digested information. Next, relevant text fragments are isolated, so that information can be extracted. As different pieces of information are merged, a coherent framework is built [126, 127]. Thus, it is clear that information extraction could serve as one of the key tools for building trusted knowledge, and could reliably check the consistency of claims in the present study, thereby answering one of the research questions highlighted in this chapter 1 [24, 25]. Here, information extraction techniques would be used to construct the knowledge base for use in Chapter 5.

This current chapter consists of the following main sections: 1) a description of this information extraction research, providing extensive information on the available empirical research and primary dataset; use of JAPE grammar rules to extract family kinship relations; results of experiments; visualisation of the resulting family tree in GEDCOM files; evaluation of the selected kinship relations; analysis of the experimental results, and 2) a chapter summary, encompassing all sections and the key information set out within them.

4.2 Research Design

4.2.1 Introduction

A research design is defined as a description of the investigation associated with a study. This includes how it is organised and conducted, the data-gathering technique applied, the research tools deployed, and the means by which the data were analysed. In general, a research design is a set of logical steps that are undertaken in a research study [128, 129].

In this thesis, information extraction is utilised to enable incremental builds of a knowledge base, containing information on the kinship ties between members of the UK's royal family. The information extraction in this study relied on NLP, implemented in the following steps. Firstly, A JAPE grammar script was used to extract information on family kinship relations from the appropriate text. Secondly, GEDCOM files were deployed to create graphical representations of these ties. Finally, the results were presented, highlighting some of the key family relations among the UK royals.

For text extraction, the Web was accessed in a process that began with the user entering a series of claims, which served as the baseline information for the entire search process, namely, the ground truth. From each claim, two named entities were subsequently extracted and adopted as search keywords. Based on these search keywords, the system began crawling and scraping the Web to find relevant articles. In the current case study, the main topic of interest consisted of family kinship relations, especially parent or spouse relations. Ultimately, these were added to the knowledge base, paving the way towards generating a genealogical tree. It is worth clarifying here that all the entered claims were visited in turn and randomly, until no more claims remained. On this foundation, a knowledge base was created and expanded as each extracted entity or relation was added. The larger the knowledge base, the more effective it would be for checking the consistency of claims.

4.2.2 Primary Dataset

The primary dataset consisted of the ground truth set of N claims. This comprised 99 true claims about the UK royal family's kinship relations, according to the corresponding family tree knowledge base [1], and 99 false claims, created by randomly matching items on a list of the UK royal family's personal names with a list of various kinship relations. The ground truth set of a total of 198 claims may be found in the Appendix C. The claims in the ground truth set were specifically generated to verify the approach. They reflect the kind of statements that a user would wish to check in the real world, because some are found online. The following are examples of claims from the ground truth set:

$$Prince\ Edward\ is\ the\ child\ of\ ElizabethII.\ \text{"True Claim"}$$

[1]Royal Family tree and line of succession:https://www.britroyals.com/royaltree.asp

$$David\ Armstrong-Jones\ is\ the\ nephew\ of\ ElizabethII.$$ "False Claim"

The ground truth set is a structured hierarchy, starting from the parent and leading to the great-grandchildren[2]. Figure 4.1 below shows diverse family relations, illustrated visually. No further details of these kinship relations are given, because such a discussion is more relevant to the next chapter, wherein the logical rule is generated.

After this extensive presentation of the dataset, known as the ground truth set (including its size), and the type of claims contained within it (that is, true and false claims), the way in which it could be used to extract family kinship relations with the help of specially constructed JAPE grammar can now be described.

[2]The ground truth set may be found in Link

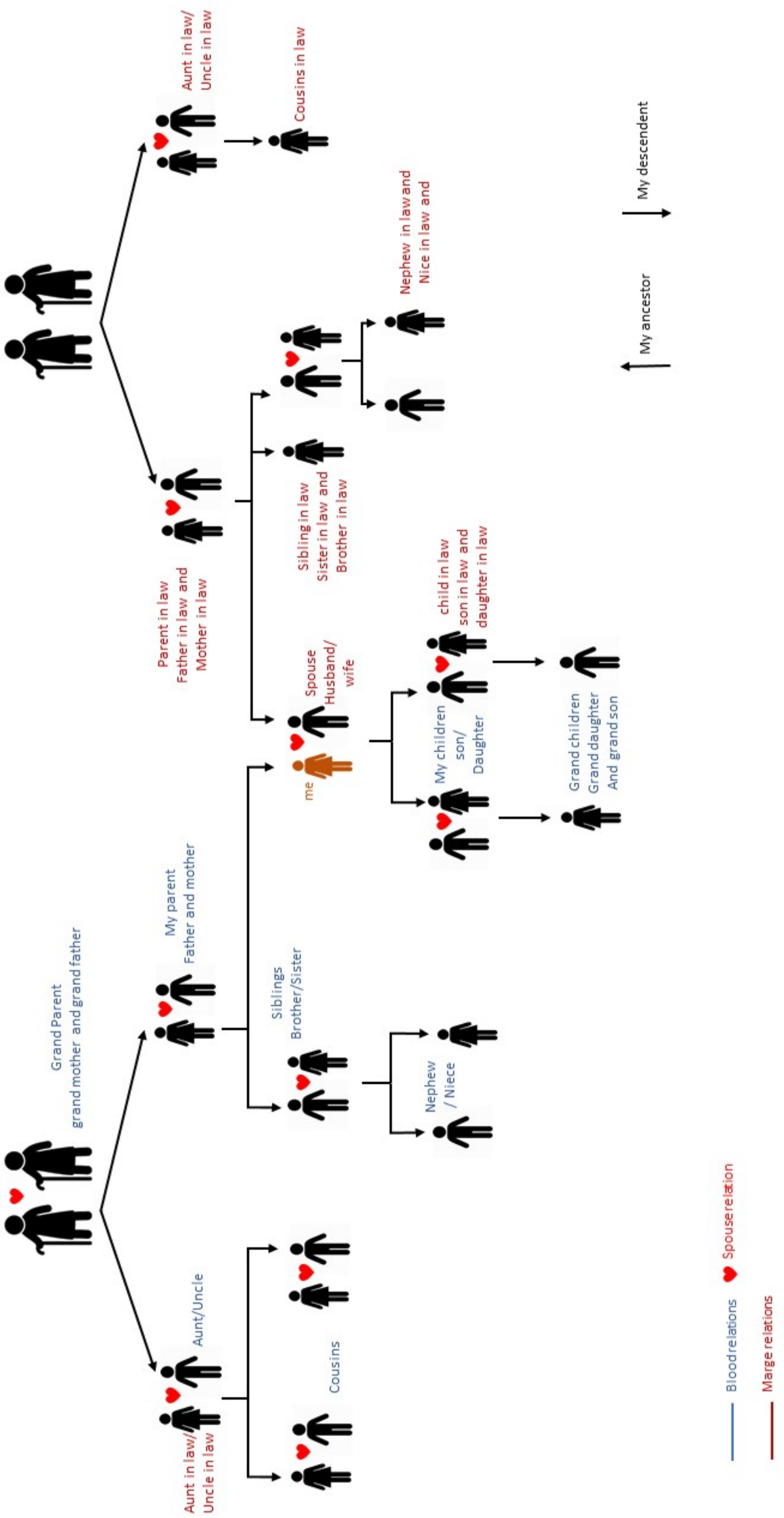


Figure 4.1: Different types of family relations

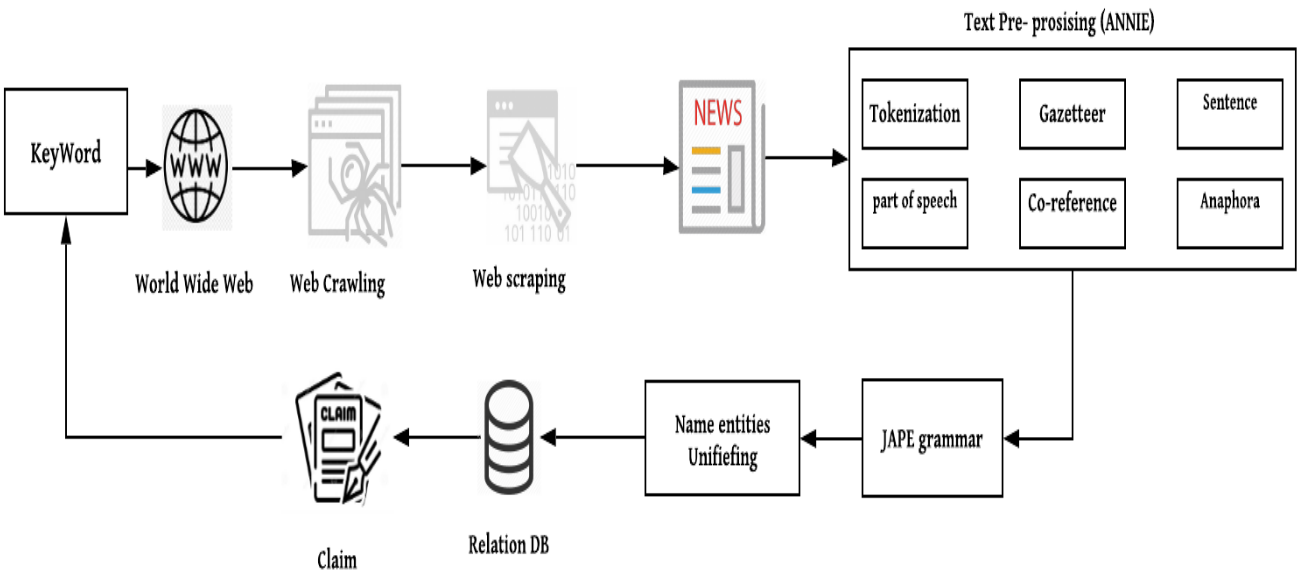


Figure 4.2: Steps followed for building the knowledge base from relations and facts

4.2.3 Extracting Family Relations Using JAPE Grammar

The JAPE grammar rules were utilised in this study to extract patterns from the articles [118]. These patterns were then stored as facts and relations in the knowledge base. Figure 4.2 shows the proposed pipeline for building the knowledge base. The pipeline begins by generating the text corpus. Next, ANNIE is deployed to pre-process the text. The JAPE grammar rules are subsequently used to extract the relations and facts. In this case study, the gazetteer was modified in ANNIE by adding some royal family titles to the lists. More details of each are provided in Chapter 3.

JAPE grammar language is a common pattern specification language (CPSL), based on the specification of patterns and action rules. The programmer can then match these patterns and rules, for example, by extracting semantics[130]. In this case study, JAPE grammar language was applied to extract and process the distinct relevant patterns that could be encountered in the inspected texts (claims and articles). Thus, JAPE grammar language extracted two types of pattern, representing parent or spouse relations. The JAPE grammar rules applied to extract parent and spouse relations are simplified and displayed in Table 4.1. To prove the concept in this particular study, 10 different patterns were generated to extract parent and spouse relations.

Table 4.1: JAPE grammar patterns for parent and spouse relations

Relation	Patterns
1. Parent relations	• $Person, \{Tokens\}, Token == (“child" \mid “son" \mid “daughter"), Token == “of", Person$ • $Person, \{Tokens\}, Token == (“child" \mid “son" \mid “daughter"), Token == “is", Person$ • $Person, \{Tokens\}, Token == “is", Person, Token == (“child" \mid “son" \mid “daughter")$
2. Spouse relations	• $Person, \{Tokens\}, Token == (“married" \mid “wife" \mid “husband"), \{Tokens\}, Person$

For the FACT approach to be applicable to other domains, the related set of patterns should be generated.

There are three different patterns that trigger the extraction of a parent relation (1):

1. If a person entity is followed by the word $child$, son. or $daughter$, followed by the word of and then another person entity; the system infers that the second person mentioned refers to a parent entity. For example, Prince Charles is the son of Queen Elizabeth II.
2. If a person entity is followed by the word $child$, son, or $daughter$, followed by the word is and then another person entity; the system infers that the first person mentioned refers to a parent entity. For example, Queen Elizabeth II's son is Prince Charles.
3. If a person entity is followed by the word is and then a person entity, followed by the word $child$, son, or $daughter$; the system infers that the second person mentioned refers to a parent entity. For example, Prince Charles is Queen Elizabeth II's son.

Similarly, another pattern is used to annotate the spouse relation (2):

1. If a person entity is followed by the word $married$, $wife$, or $husband$, followed by another person entity; then the system infers that each person entity in these relations is the spouse of the other. For example, Prince Philip is the husband of Queen Elizabeth II.

The extracted information , i.e., the kinship relations, are added to the knowledge base, when:

- These patterns are encountered across multiple texts.

- These patterns have different sources.

In Chapter 5, it is demonstrated how to model the family-tree relationships, using probabilistic reasoning.

4.2.4 Solving Named Entity Ambiguity and Storing Structured Information

On some occasions, person entities become confusing. For example, in this study, it was necessary to determine whether two entities (Charles and Prince Charles), referred to the same person. This can occur when extracted entities have different names in the same documents or multiple sources. When an ambiguity of this nature manifests amongst entities in a document; co-referencing will resolve the problem. In this study, ANNIE in GATE was the co-reference resolution tool deployed. However, when ambiguity occurs among entities mentioned in several sources, features extraction (which is based on using a knowledge base) is employed to extract features and identify the degree of similarity between named entities. Here, the AmbiverseNLU system was used for this mission. The similarity toll was applied to unify the questioned entities, wherever there was a high degree of similarity. To be more specific, the entities that were found to be the same person were unified by giving them a standard name to replace the confusing names in the sources [25]. In this case study, AIDA was also applied, this being an online tool that performs co-referencing and ambiguity resolution [117]. For instance, it could resolve royal family titles such as 'Prince of Wales'.

To summarise, the processes implemented to resolve named entity ambiguity in this study were: i) the resolution of co-referencing chains via GATE, presented as local co-referencing chains (see above); ii) the feature extraction and identification derived from the knowledge base by applying the AmbiverseNLU system, which dealt with Wikipedia links, name types, and AIDA for the royal family titles; iii) the use of similarity to check the co-references of a given entity and a feature extracted from the knowledge base, and iv) all entities displaying a high degree of similarity were replaced with the standard name for all entities of similar height.

4.2.5 Continuous Learning Process

The extraction of relations, achieved using ANNIE and JAPE grammar, was the sole technique applied to construct and expand the knowledge base in this research. Expanding the knowledge base is basically a continuous learning process, as new relations and properties are added to the knowledge base. This takes place each time a new claim is processed, and its related entities are used to search the Web to gather the related text [131]. Once the stage of expanding the knowledge base was completed for a claim, the next claim was processed in a similar manner. Hence, the newly added knowledge would always support the next claim.

Figure 4.3 shows how the knowledge base was initialised and then expanded using the continuous learning process. By extrapolation, the increase in the number of extracted relations

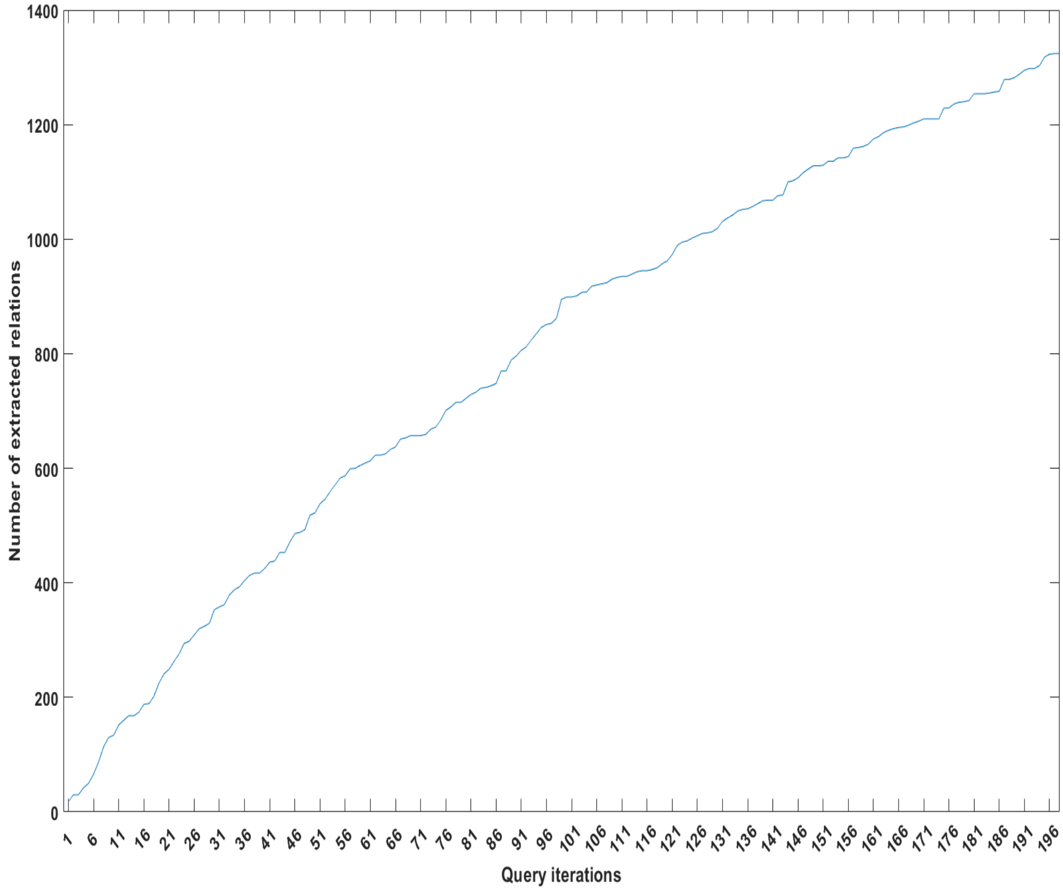


Figure 4.3: How the knowledge base increases with continuous learning: vertical axis (y-axis) shows number of facts learned by FACT; horizontal axis (x-axis) shows number of claims

slows down from around 101 query iterations. However, the general trend is still up, that is, an increase in the number of relations extracted. Therefore the size of the knowledge base will still increase. More details of this may be found in section 3.3.4.

4.2.6 Experimental Results and Analysis

In this section, the result of the case study is presented, wherein an attempt was made to establish family kinship relations within the UK's royal family, these being the main contribution to building the knowledge base. It is worth mentioning that in this research, only two types of relations were utilised: parent and spouse. No other relationship types were explicitly extracted. However, they could be discovered implicitly. For example, parent relations between X and Y, and parent relations between X and Z were extracted. Consequently, it was deduced that Y and Z were siblings. This implicit discovery of relations is called the 'Inference Process' and it is described in detail in Chapter 5.

In order to build the text corpus, the Web was crawled and scraped. To obtain text information with a degree of authenticity, only Web links that referred to trusted sources were targeted. A trusted source would represent any source added to a trusted list. In this study, the choice was made to only trust well-established, traditional news media, such as newspapers, not social media. However, this was adopted as a general approach, wherein the list of trusted sources could be easily changed. Trust was therefore not a technical concept in this study, and any user could apply this approach according to their own concept of a trusted source.

The system begins with processing claims from the dataset. As an example, the first claim, 'Prince William is the father of Prince George' may be considered. Entity extraction takes place and produces: 'Prince William' and 'Prince George'. The two entities are used as Web search queries. Web-crawling extracts related links. For each link, the articles are then scrubbed, and the resulting text is added to the text corpus. This procedure is repeated for each claim in the dataset, identified earlier as a continuous learning process. Consequently, the knowledge base is expanded, leading to more accurate fact-checking [25]. As a result, it was concluded here that the accuracy of the fact-checking would be correlated to the size of the knowledge base. Table 4.2 illustrates the number of articles scraped during the first 12 iterations of this continuous learning process, revealing that the number of extracted relations increased in each iteration. The full Table is illustrated in Appendix B.

For this case study, the ground truth dataset was used, containing 198 claims. All these claims were processed in random order. The results are illustrated in B, which demonstrates that the maximum number of processed articles for a claim was 172. Meanwhile, the minimum number was 44, and the maximum number of relations discovered by a single claim was 34. The cumulative number of spouse and parent relations reached 1046 and 248, respectively.

For the correlation between the combined entities and newly founded relations, some of the combinations were found to reveal a large number of relations. To be more specific, the combination of 'Lady and Sarah Chatto' and 'Princess Anne' appended 34 new relations to the knowledge base. However, other combinations of entities resulted in discovering very few relations. For example, 'Prince Harry', combined with 'Prince William' and 'Kate Middleton' and then incorporated with 'Prince William' did not add any relation to the knowledge base.

It was concluded that the causes of this fluctuation were:

1. As this combination of entities was previously searched, based on a prior claim, i.e. 'Kate Middleton' combined with 'Prince William', the relations of relevance were already extant in the knowledge base. Consequently, no new relations were added.

2. Due to the claims having been randomly processed, a particular combination of keywords would introduce new relations into the knowledge base, as it was the first time that they were being processed. However, subsequent occurrences of this combination would add no value to the knowledge base. This elucidates why a claim listed in the upper section of the

Table 4.2: Number of relations extracted with respect to number of articles processed in each iteration (first 12 iterations illustrated), as well as cumulative spouse relations (CSR), cumulative parent relations (CPR), parent relations (PR), spouse relations (SP), and total number of relations (TR)

Search keywords	# articles	# CSR	# SR	# CPR	# PR	# TR
“Prince Edward“ “Peter Phillips“	117	16	16	1	1	17
“Princess Anne“ “Zara Phillips“	117	31	15	6	5	37
“Prince Philip“ “Arthur Chatto“	105	35	4	11	5	46
“Mary Elphinstone“ “Elizabeth II“	113	41	6	12	1	53
“Prince Philip“ “Princess Margaret“	128	59	18	12	0	71
“Prince William“ “Elizabeth II“	130	79	20	13	1	92
“Lady Sarah Chatto“ “Princess Anne“	118	103	24	23	10	126
“Princess Beatrice“ “Prince Charles“	125	111	8	25	2	136
“Princess Beatrice“ “Elizabeth II“	141	127	16	25	0	152
“Kate Middleton“ “Prince William“	84	129	2	25	0	154
“Sarah Fergie Ferguson“ “Prince William“	102	135	6	27	2	162
“Princess Charlotte“ “Prince Harry“	118	138	3	27	0	165

claims list would make no contribution to expanding the knowledge base, while a later claim, bearing the same entities, would append some relations to the knowledge base.

3. Some of the extracted entities had low popularity in the domain; in the present case, this concerned any member of the royal family who was not well known. Therefore, they were rarely mentioned in the sourced Web articles, and so limited text could be relied upon to discover relations.

To gather more reliable relations, a constraint was imposed, namely, that a relation would only be appended to the knowledge base if it was discovered from a minimum of three different sources.

In Figure 4.4, a bar chart presents the correlation of combined entities and the number of relations discovered.The X-axis shows the combination of 198 keywords, while the Y-axis shows the number of relations extracted using these keywords. The Y-axis spans 0-34, which was the maximum number of relations extracted from a claim. To clarify this graph further, it is divided into four sub-graphs, each representing a combination of 50 keywords, and the associated relations that were generated.

When processing a combination of entities, the sub-graphs illustrate the following:

1. "Lady Sarah Chatto - Princess Anne" revealed 34 new relations.

2. "Elizabeth II - Anthony Armstrong" and the other combination of " Diana Spencer - Elizabeth II" appended 26 new relations.

3. "Elizabeth Bowes Lyon - Elizabeth II" , "Prince George - Lady Sarah Chatto" added 25 relations.

4. "Anthony Armstrong-Jones - Princess Margaret", "Peter Phillips - Elizabeth II", added 24 relations.

5. "Prince Harry - Isla Elizabeth Phillips" added 23 relations.

6. "Prince William - Elizabeth II" and "Mary Ephinstone - Cecilia Bowes" added 21 relations.

Conversely, specific keyword combinations introduced very few new relations into the knowledge base. These were:

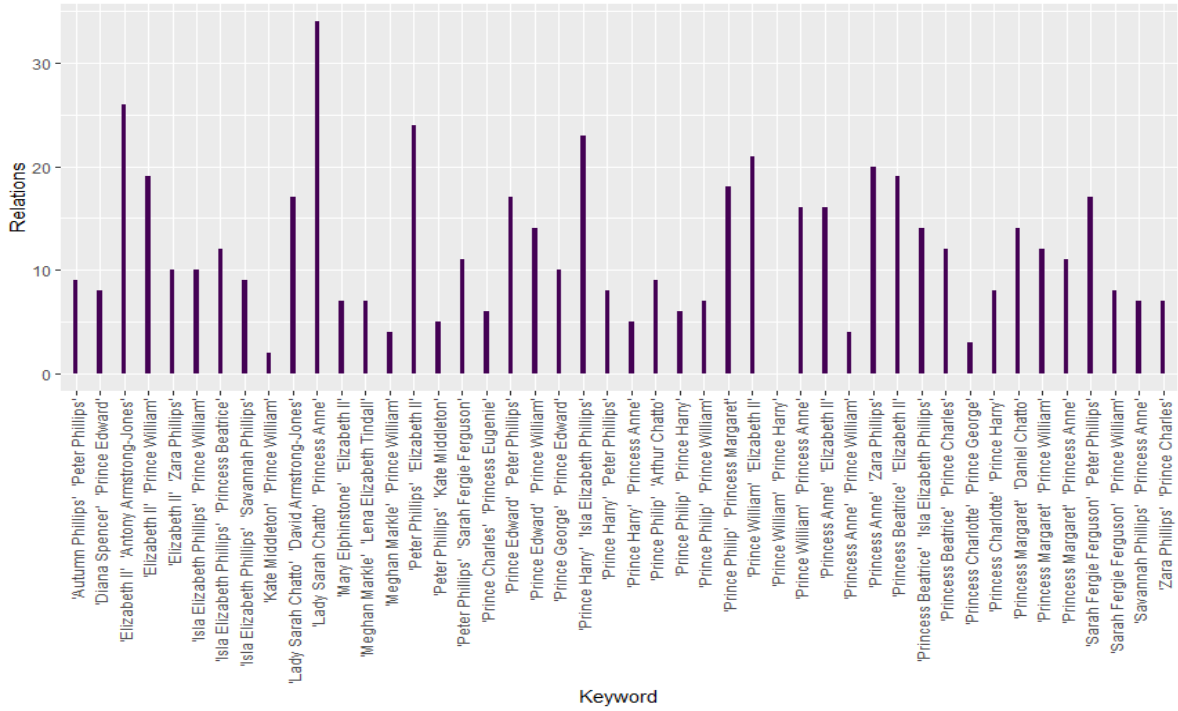
1. "Prince Harry - Prince William" added 2 relations.

2. "Prince William - Prince" and "Arthur Chatto - Prince Harry" appended 1 relation.

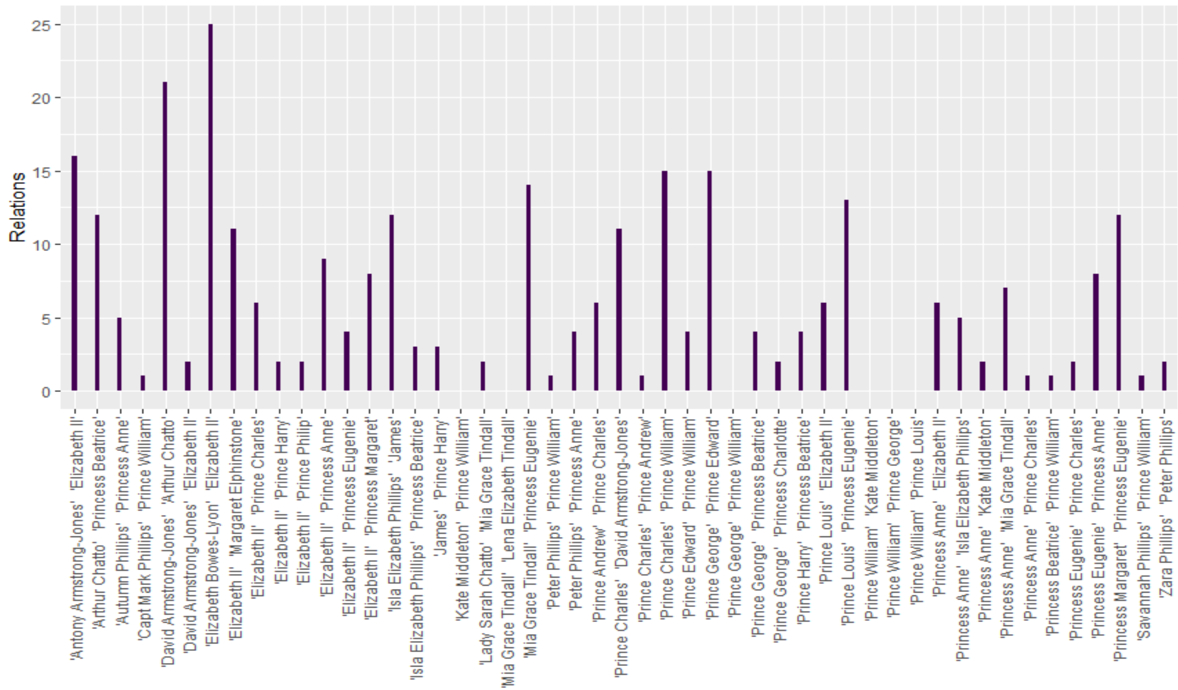
3. "Prince William - Prince Harry", "Kate Middleton - Prince William", "Prince George - Prince William" and "Prince Andrew - Princess Anne" brought no new relations to the KB.

In conclusion, the first claims to be run were more likely to generate large numbers of relations, while those run later brought limited new relations, given that a relation could only be added once to the knowledge base, and could not be added again [24, 25].

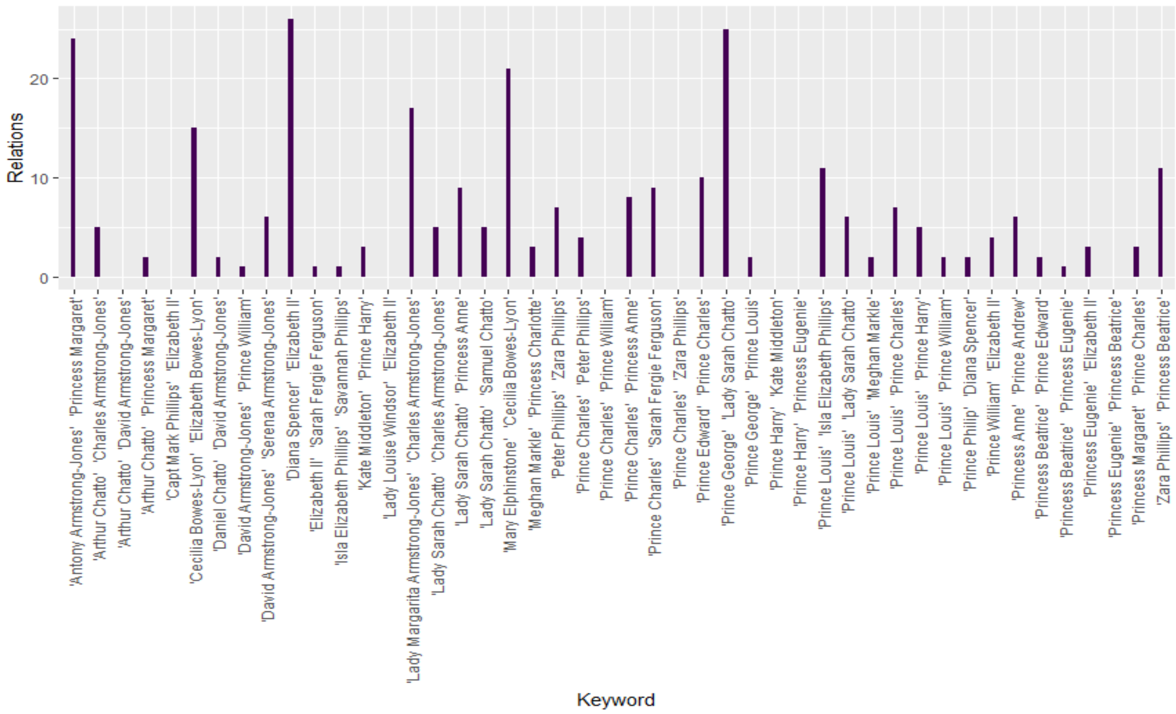
Another method of assessing this case study was to evaluate the correlation between the number of retrieved articles, and the number of relations discovered. As mentioned earlier, the maximum number of retrieved articles based on the claims in the dataset was 175. These 175 articles were then divided into 7 clusters, wherein each cluster contained 25 articles. Therefore, the first cluster ranged from 1-25, the second ranged from 25-50, and so on. Table 4.3 presents the number of the cluster, ranging from 1-7; the number of articles in each cluster, and the number of spouse and parent relations discovered in relation to a cluster. The same data are illustrated in Figure 4.5 as a bar chart.



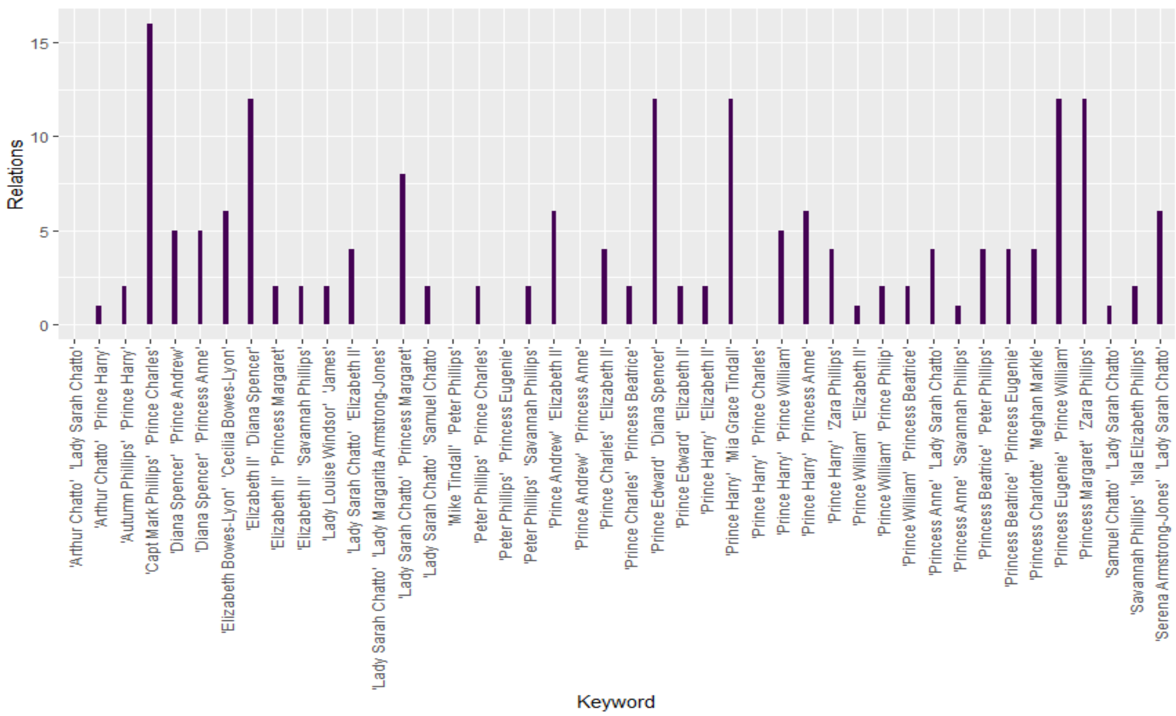
(a) Keywords from 1 to 50.



(b) Keywords from 51-100.



(c) Keywords from 101-150.



(d) Keywords from 151-198.

Figure 4.4: Number of relations extracted per keyword: X-axis shows keyword; Y-axis shows number of relations

Table 4.3: Number of articles processed in each iteration, and corresponding number of relations extracted

# Cluster	# Articles Extracted	# Spouse Relations	# Parent Relation
1	[0-25]	0	0
2	[25-50]	2	0
3	[50-75]	29	5
4	[75-100]	135	61
5	[100-125]	473	119
6	[125 -150]	341	63
7	[150-175]	66	0

For Cluster 1, where the number of articles ranged from 0-25, no relation of any type was found. Meanwhile, Cluster 2 comprised a number of articles, ranging from 25-50. When the number of articles reached 44, two spouse relations were deduced, but no parent relations. In the Cluster ranging from 50-75 articles, 29 spouse and 5 parent relations were generated. In Cluster 4, which ranged from 75-100 articles, 135 spouse and 61 parent relations were identified. The most prolific cluster, in terms of the number of generated relations, was Cluster 5, which ranged from 100-125 articles. Here, 473 spouse relations and 119 parent relations were identified. This was the highest number of relations extracted in a cluster. Nevertheless, in Cluster 6, 341 spouse and 63 parent relations were obtained. Finally, in Cluster 7, which ranged from 150-175 articles, 66 spouse relations were identified, with no parent relation.

Another evaluation technique in this case study was to disclose the interconnection between each keyword, as well as the number of relations detected. This is demonstrated in 4.4 and illustrated as a pie chart in Figure 4.6, which was built from Table 4.4.

The keywords were segmented according to the number of relations that they contributed to the knowledge base. Each segment represented a range of relations that were extracted from a specific keyword. Based on the results, four segments are defined. The selection of the keywords was informed by statistical segmentation, according to the number of relations generated, as follows: 0-5, 5-15, 15-25, and 25-35 (based on the distribution of the relations). This revealed that the keywords generated a number of relations in the range of 5-15, representing nearly half (49%) of the total contribution to building the knowledge base (presented in blue in the pie chart). The keywords subsequently generated a number of relations in the interval 15-25, which contributed 26% to building the knowledge base (indicated in orange). Next, the keyword contribution to building the knowledge base generated a number of relations in the interval 0-5, therefore, an 18% contribution to the knowledge base (indicated in green in the pie chart). Finally, the keywords that generated the number of relations in the interval 25-35 contributed 7% to building the knowledge base (indicated in red in the pie chart). The final segment shows that the

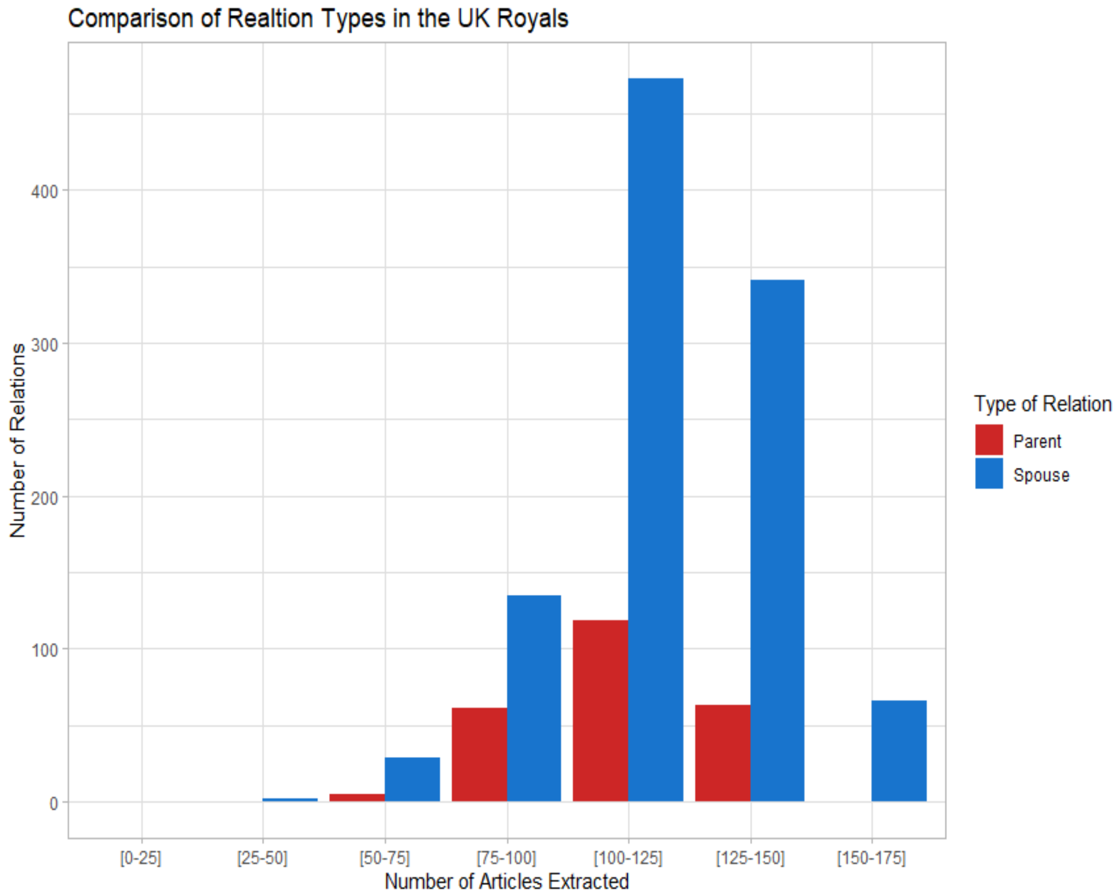


Figure 4.5: Number of spouse and parent relations generated, given range of number of articles

larger individual contributors inversely made the lowest overall contribution. On the other hand, the keywords in the 5-15 segment made the largest contribution to the knowledge base, due to the higher number of relations generated in this segment.

Based on the previous analysis we conclude that:

1. Few keywords were very prolific in terms of generating new relations. Moreover, their overall contribution remained insignificant, due to their limited number

2. Conversely, other keywords compiled a medium-level contribution, but as they were common, their combined contribution to expanding the knowledge base was greater.

3. The number of relations derived was a function of the time when the article was processed. As it was prohibited to duplicate a relation, the first time an article was processed was the most fruitful in terms of generating new relations.

4. Knowledge base expansion through a process of continuous learning is dynamic, and this dynamism was the key to accurate fact-checking in this study.

Table 4.4: Cumulative number of relations correlated with corresponding number of keywords

Keywords Cluster	Total # Relations
"Princess Charlotte - Prince Harry", "Prince Harry - Princess Anne", "Peter Phillips - Kate Middleton", ...	240
"Princess Margaret - Daniel Chatto", "Prince Edward - Prince William", "Mia Grace Tindall - Princess Eugenie", ...	633
"Elizabeth Bowes-Lyon - Elizabeth II", "Prince George - Lady Sarah Chatto", "Peter Phillips" "Elizabeth II", "Antony Armstrong-Jones - Princess Margaret",...	335
"Lady Sarah Chatto - Princess Anne", "Elizabeth II - Antony Armstrong Jones", "Diana Spencer - Elizabeth II"	86

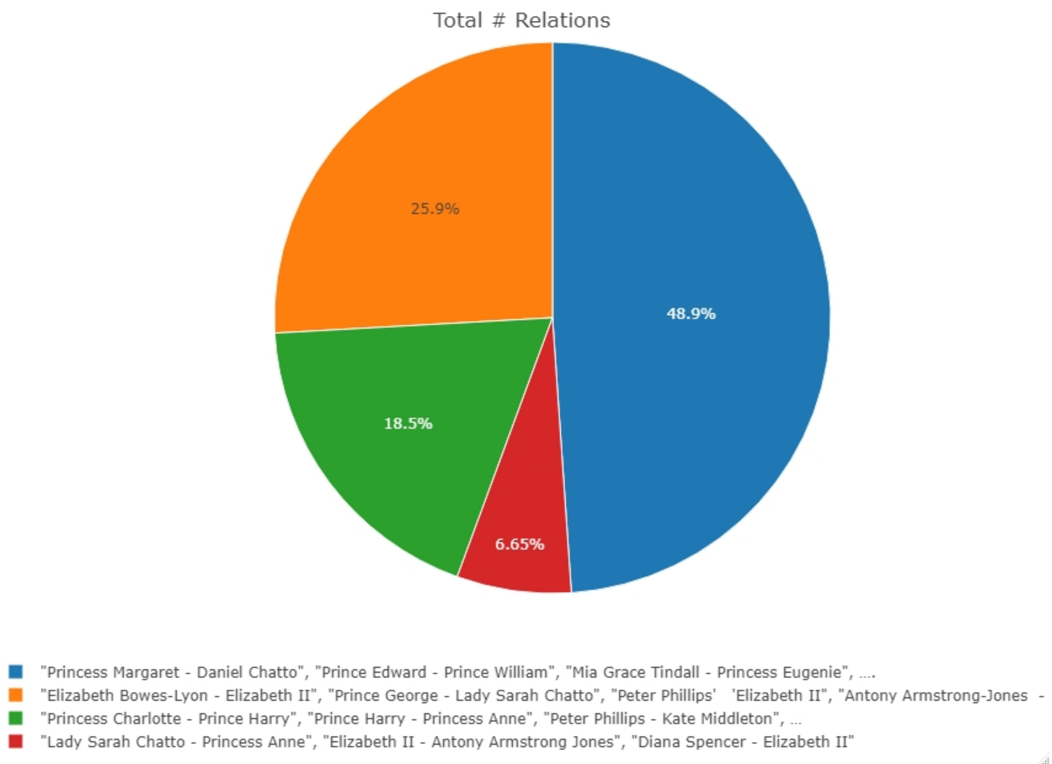


Figure 4.6: Total number of relations, given keyword segment associated with relations

4.2.7 Visualising the Family Tree with GEDCOM File

In this case study, visualisation of the UK royal family's genealogical family tree was one of the main outcomes. A GEDCOM file is in a format that is indicated by the extension 'ged', which

```
0 HEAD
0 @P1@ INDI
1 NAME Prince Andrew
1 FAMS @F1@
0 @P2@ INDI
1 NAME Princess Beatrice
1 FAMC @F1@
0 @P3@ INDI
1 NAME Princess Eugenie
1 FAMC @F1@
0 @F1@ FAM
1 HUSB @P1@
1 CHILD @P3@
1 CHILD @P2@
0 TDLR
```

Figure 4.7: GEDCOM text file sample

contains information about families and individuals, for example, Prince Andrew's name, date of birth, family ID, and position in the family (spouse/child). Screenshot of the GEDCOM text file example is illustrated in Figure 4.7. The GEDCOM file consisted of plain text and contained information about individuals' family relations, with metadata linking these records together. The GEDCOM file began with the 'HEAD' and ended with 'TDLR'. Meanwhile, the individual records (INDI) defined individuals, for example, Prince Andrew is identified as P1 in the Figure, and the family record (FAM) links the husband (HUSB), wife (WIFE), and child (CHIL) through their ID numbers. This file was used to build the family tree. More specifically, all the generated spouse and parent relations were stored in the GEDCOM file, with the aim of visualising the UK royal family tree, as illustrated in Figure 4.8 [25].

In Figure 4.8, Queen Elizabeth II is presented as the root, and her descendants are depicted as nodes connected to the root of the tree. The tree indicates that Queen Elizabeth II has five (5) children connected to her: three (3) male and two (2) female. However, Queen Elizabeth II actually only has four (4) children: three (3) male, and one (1) female. In addition, Lady Frances Armstrong-Jones is presented as the daughter of Princess Anne in the tree, which is incorrect.

Therefore, the generated tree is not accurate, and hence, there must be an issue with the consistency of the technique applied. This issue stems from the co-referencing, whereupon the system was unable to detect that 'Princess Anne' and 'The Princess Royal' were the same individual.

Despite this inconsistency in the production of the tree, it was at least proven that FACT [25] could accept a text corpus as input and then generate a tree, as shown in Figure 4.8. The task of duplication-resolution is usually carried out at the level of the unifying entities in the pipeline. However, the issue could not be fully resolved at that stage, because the information that was available in the knowledge base was insufficiently comprehensive. This means that more effort must be made to resolve the problem [25] .

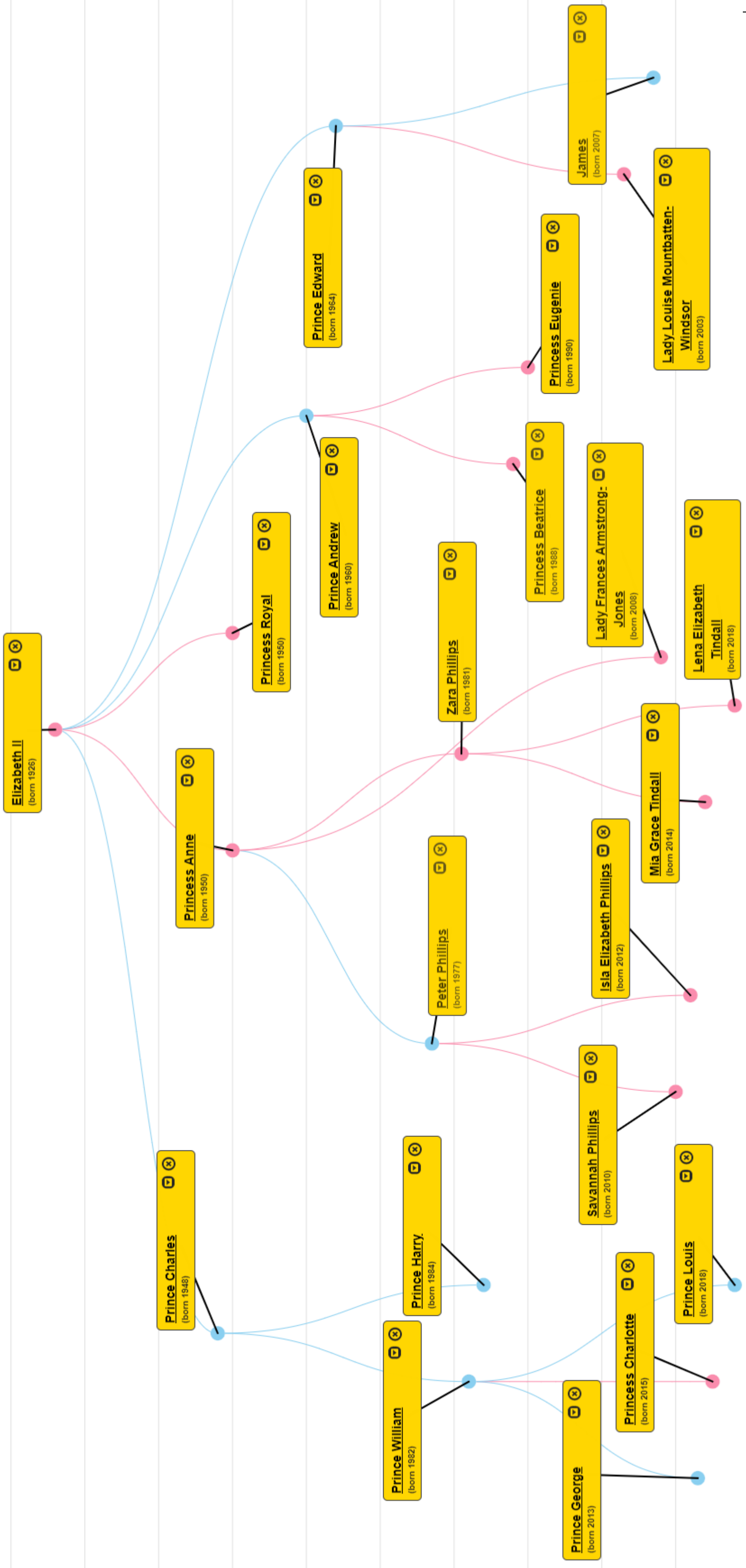


Figure 4.8: Family tree with Queen Elizabeth as root, together with her descendants

The logical reasoning inference framework, discussed in the following chapter (chapter 5), enables the approach to check the consistency of a claim with the knowledge base. Therefore, the question of whether or not Lady Frances Armstrong-Jones is the daughter of Princess Anne could easily be resolved via deduction, provided that the information residing in the knowledge base is sufficient to allow for the necessary inferences. However, this does not entirely solve the question of consistency because, in practice, such information might not be available in the knowledge base at the time of making the necessary inferences. It is important to note here that the knowledge base might contain some incorrect information, and this would need to be reduced.

4.2.8 Evaluation of the Relations Extraction

In the field of information extraction, most of the evaluations take place by employing quantitative analysis techniques, and there are many different options for making this assessment. In the current experiment, precision and recall were used to generate a beneficial valuation, as shown in 3.5.1. Both these measures were defined in terms of the number of retrieved and relevant relations. Thus, precision was defined as the fraction determined by the total number of correctly extracted relations, divided by the total number of relations identified by the system – as shown in equation 3.1. In contrast, recall was defined as the fraction determined by the total number of correctly identified relations over the total number of relations present in the text, which is shown in equation 3.2 [124].

Nevertheless, it is important to be aware that this evaluation was performed using randomly selected BBC articles about the UK's royal family. These were then manually annotated by an expert on the UK royals. The suggested pipeline in this research detected 16 spouse and 17 parent relations. The articles were subsequently analysed with the JAPE grammar script to detect how many of them fell into the following sets: true positives (TPs), true negatives (TNs), false positives (FPs), and false negatives (FNs).

The results obtained were as follows:
TPs = 15 (spouse) , 13 (parents)
FPs = 2 (spouse), 3 (parents)

It is important to note that because this experiment did not include any false relations, the number of false positives and false negatives was zero in each case. Therefore, the precision and recall applied to evaluate the method adopted in this study yielded the following results,
Precision = TP/(TP+FP) = 28/33 = 84.84%
Recall = TP/TP + FN = 28/33 = 84.84%

Consequently, this technique, which utilised the JAPE grammar constructed for this study and the extraction of kinship relations with respect to the UK royals, demonstrated reliable and authentic precision and recall. Thus, the suggested pipeline was found to be effective for building a knowledge base that could be employed later to check the consistency of facts.

4.3 Summary

In this chapter the construction of the primary dataset was explained in detail. The utilisation of ANNIE, in combination with JAPE grammar, to deduce relations that would build and expand the knowledge base for this case study was demonstrated in this chapter. The named entity unifying technique was also explained. Based on the results of the case study, it may be concluded that:

1. There is inconsistency in the correlation between the combined entities and their respective newly appended relations. To be more specific, some combinations revealed numerous relations, while others derived no relations. This is justified as the combination of entities having previously been searched for; consequently, no new relations were added. Thus, the first time a combination of entities was processed was the most fruitful. Another reason might have been that some of the entities were well known, while others awee not. The popular entities derived more relations, simply because more articles mentioned them.

2. There is a correlation between the number of articles processed in each iteration, and the corresponding number of extracted relations. The greater the number of relevant articles to analyse, the more relations there were to be defined. However, beyond a certain threshold, this correlation was inverted. To be more specific, the number of derived relations did not increase, even when more articles were investigated.

3. This represented variation in the number of extracted relations with regard to their keywords. The most significant contributors were those keywords that added medium-sized relations and also occurred frequently. Their effect in expanding the knowledge base was more important, compared to those keywords that deduced innumerable relations, but only occurred rarely.

4. The proposed pipeline achieved good performance in terms of precision and recall by achieving 84.84% and 84.84%, respectively.

CHAPTER 5

FACT CONSISTENCY CHECKING USING PROBABILISTIC REASONING

5.1 Introduction

Fake news is a challenge that the contemporary world is facing. Massive data are created and shared via the Internet every day. Therefore, identifying these false data and preventing their spread has become a matter of growing concern for the scientific community, with numerous studies being conducted in this field. Technologies, particularly AI, are consequently employed to address the problem. In particular, probabilistic logics such as PSL and MLN are utilised to check the credibility of data. If fake data are discovered, they are removed from knowledge transfer, knowledge acquisition, and knowledge-sharing systems [132].

The vast amount of true and false information sources makes it difficult, even for domain experts, to distinguish between the two. Several methods have been employed to address this issue; some approaches have focused on natural language analysis, in terms of stylistic signals or the referencing of explicit statements that appear in a corpus [18–21, 26]. One of these methods involves determining the consistency of a claim in relation to a given trusted knowledge base. To be more specific, it consists of marking certain information as valid by checking its source [133]. Therefore, any piece of data that is consistent with this trusted knowledge may be considered as truthful, while inconsistent information is regarded as untrue [25].

In this thesis, the FACT approach is proposed, which facilitates the checking of a claim's consistency against a previously built and trusted knowledge base [24, 25]. This represents the uniqueness of the current approach, which functions via fuzzy logic.

The present chapter comprises the following sections: 1) a description of PSL and MLN (the Tuffy framework); 2) the notion of the model's prior beliefs, addressed with an exposition of how such 'priors' can be used to build initial belief structures that will support and lead to the finding/inferring of new relations; 3) PSL and MLN inference, outlined in some detail; 4) a case study on checking the consistency of claims about the UK royal family; 5) a case study on fact-checking political relations using PSL, and a chapter summary.

5.2 Probabilistic Soft Logic (PSL) and a Markov Logic Network (MLN) as Models for Statistical Relational Learning (SRL)

Statistical relational learning (SRL) is an emerging domain within AI. It primarily involves a combination of statistical and logical methods. Statistical relational models mainly fall into two main categories: generative models and discriminative models. Lee and Wang [68] discuss PSL and MLN as the two most popular discriminative models of SRL. The applications of these models include social network modelling, entity resolution, and NLP systems.

Al-shukaili [134] studied the integration of MLN with linked data, aiming to enhance keyword search. Linked data search engines are classified according to the resource indexation method implemented. Two main approaches may be adopted: mapping to the document or mapping to the resource URL. Irrespective of which of these approaches is applied to support the search, the result is often a combination of many different types of resource.

For NLP, both PSL and MLN utilise the representation of natural language semantics, applying weighted inference rules to analyse the distribution of semantic information. In addition, PSL and MLN combine the application of logic and statistical methods. To be more specific, they both use a form of fuzzy logic based on Boolean logic, with a weighting scheme. The weight values in PSL vary in the interval [0, 1], whereas the weight values in MLN are purely binary (0 or 1) [68, 69]. This is one of the reasons why PSL has proved to be more effective than MLN in representing natural language semantics, as it relies on high-level representations to construct the conditions for sentence-similarity [67]. Furthermore, PSL, as a template language, is used to represent hinge-loss Markov random fields; it can deal with soft truth values, which is considered to be one of its main features. This feature allows PSL to indicate the degree of agreement between a PSL rule and a claim made about the subject area [135].

5.2.1 Language and Syntactic Structure of Probabilistic Soft Logic (PSL)

Probabilistic soft logic is a notable statistical relational template language, which defines its model structures based on weighted first-order logic [65]. A first-order knowledge base mainly consists of a set of first-order logic formulas or sentences. These formulas are developed using variables, functions, constants, and predicates. Interpretation explains the objects, relations, and functions of a particular domain, which are then represented through specific symbols [112]. Indeed, PSL is mainly a general-purpose probabilistic programming language that involves defining templates for constraints and potentials. In particular, each template implements an abstract dependency that can take the form of either a hard constraint or potential function. Both may be specified with all necessary parameters, including the weight of the potentials. The syntax allows for the use of identifiers, logical rules, predicates, targets, and observations [65].

Probabilistic soft logic rules are written as an implication. In logic, this implication can be changed to a disjunctive form. Moreover, a logical rule can be weighted/soft or non-weighted/hard, in either case consisting of a disjunctive clause of the literals, which are vital to represent the logical dependencies in a model. It is important to note that a literal can either be an atom or negated atom. A negated atom is depicted by a '!' or '~' symbol, with the value of 1-atom, where an atom is the positive value of the atom [65]. For example, if Friends ('Person A', 'Person B') has a value of 0.6; then the literal, !Friends ('Person A', 'Person B') will have a value of 0.4. Usually, a weighted logical rule begins with a non-negative weight and tends to end with the same value, but with an exponent of 2 ($\wedge^2$). The fuzzy logic operators are conjunction (& &, or &) and disjunction (|| or |). The symbol ($\Rightarrow$ or $\Leftarrow$) is used to show the implication in the logical rule. The following are examples of such logical rules:

```
1: Advisor(Prof, S) ∧ Department(Prof, Sub) ⇒ Department(S, Sub)
         Friends(X, Y) ∧ Friends(Y, Z) ⇒ Friends(X, Z).
```

A predicate may be considered as the relation that is named by creating a distinct identifier and associating it with a positive integer, called an 'arity'. This arity depicts the number of terms that it can accept as arguments. Each predicate in a PSL program is required to have a distinct identifier as its name. In addition, predicates must be classified as either open or closed, depending on whether they are unobserved or observed [65]. For instance, a predicate is said to be 'closed' when the information portrayed/required by that predicate is fully known, because it is already present in the knowledge base. Conversely, when that information needs to be inferred, the predicate is said to be 'open'. The following are examples of predicates:
The specification '`Friends/2`' introduces a binary predicate. It indicates that the Friend condition, relation, or predicate can exist in relation to two arguments: the two people (constants) who are friends.
`Advises (Professor, Student)` is part of the definition of the open predicate, 'Advises'.
`CourseTaken/2` introduces a binary predicate. This predicate takes two arguments, each of which represents a constant: a 'Student' and a 'Subject' of study
`CourseTaken (Person, Subject)` (closed) is a closed predicate.
Predicate `CourseTaken/3` is a ternary predicate. The example relates three entities to each other: an instructor, a student, and a subject `class (additional attribute)`.
`CourseTaken (Professor, Subject, Student)` (closed) is a closed predicate.

Observations consist of the data on which PSL programs are grounded. They are usually detailed in a list that is specifically constructed for that purpose [65]. Presented below is an example of such a list:

$$\texttt{Department ('Alex', 'Data Science')} = 1$$
$$\texttt{Department ('David', 'Statistics')} = 1$$
$$\texttt{Department ('Bob', 'Computer Science')} = 1$$
$$\texttt{Advises ('Alex', 'David')} = 1$$

As described in [136], PSL framework-grounding is the process through which each ground rule is instantiated, that is, provided with data and a rule template. This can be performed in two ways: top down or bottom up. Top-down grounding is simpler and easier to implement. It Initially involves application of the rules and then employing nested loops to undertake replacements across all the variables. In contrast, bottom-up grounding, as applied in PSL, is performed in two phases, the first of which involves issuing database queries to find all the constants that are assigned to the variables for a specific rule. Subsequently, the returned variable assignments are instantiated into a ground rule.

5.2.2 Language and Syntactic Structure of a Markov Logic Network (MLN)

Markov logic networks represent a language that applies weighted first-order logic. An MLN is viewed as a template for building a Markov network. Tuffy is an MLN inference engine that offers the benefits of scalability, together with an order of magnitude speed benefit, in comparison to previous similar systems [137]. A Tuffy program takes a text file containing predicates and rules as its input. Tuffy can use two types of inference: maximum a posteriori probability (MAP) inference (where posterior probabilities are found) and marginal inference (involving the computation of marginal probabilities) [137]. Markov logic network rules can either be hard or soft – either or both of which can represent the first-order logic. The only difference between soft and hard rules is that a soft rule is associated with a real number (positive or negative). In contrast, hard rules are indicated by the use of a full-stop, which depicts the weight: $+\infty$. The rules are made up of literals, connected through + conjunction or disjunction. The following are examples of MLN rules:

$$\texttt{0.5 Smokes(a1)} \Rightarrow \texttt{Cancer(a1)}$$
$$\texttt{0.4 Friends(a1, a2)} \wedge \texttt{Smokes(a1)} \Rightarrow \texttt{Smokes(a2)}$$

A predicate schema — one of the main components of an MLN specification – is a list of declarations of all the predicates to be used. This is where each predicate declaration specifies the name of the predicate, along with its parameters, thereby indicating the type of each predicate. As mentioned earlier, predicates are classified as open or closed; an open predicate includes tuples (of parameters) with unknown truth values, while a closed predicate is fully evidenced and any tuples that are absent from the evidence files are considered false. This situation is generally depicted with an asterisk. The following is an example of a predicate declaration:

`Smokes(person),`

```
Cancer(person),
* Friends (Anna, Bob)
```

Queries can be issued by the user; these consist of a set of atoms that are specified in query files or via a command line. The following is an example of a query in MLN:

```
-q Cancer(x)
Cancer (x)
```

In addition, evidence can be provided by the user to perform the MLN inference via Tuffy. An evidence file contains a list of atoms (wherein an atom preceded by the '!' symbol indicates negation). The following is an example of an evidence file:

```
Friends(Edward, Frank)
Friends(Gary, Helen)
!Friends(Gary, Frank)
```

The grounding process in the Tuffy framework makes use of bottom-up as PSL and use SQL queries. In the next section, we will compare PSL and Tuffy framework.

The grounding process in the Tuffy framework makes use of bottom-up grounding as in PSL, and applies Structured Query Language (SQL) queries. In the next section, PSL and the Tuffy framework are compared.

5.2.3 Comparison of the Probabilistic Soft Logic (PSL) and Tuffy Frameworks

This sub-section compares and contrasts the framework supporting PSL with the framework that supports Tuffy, and examines what these frameworks have allowed these systems to achieve in their application and performance. In this regard, it is important to recall that both PSL and Tuffy are based on statistical and logic frameworks. Moreover, both develop and build models based on uncertainty. For example, as in the present case, PSL enables the researcher to determine the level of a claim's consistency, according to the knowledge stored in a knowledge base. Tuffy, on the other hand, simply helps determine whether this same claim is true (1) or false (0). In this situation, PSL uses interval truth values [0, 1], while Tuffy merely uses binary values, basic to each of these systems and elements of their core frameworks, respectively. In this regard, [138] maintain that both approaches apply weighted logic rules while adopting a model to define joint distribution over missing information. It should be noted that Tuffy, despite employing a truth value, produces a binary result.Conversely, PSL could be seen as a discriminative statistical relational framework, which adopts HL-MRF (based on real-value probabilistic weights) as a graphical model. Meanwhile, Tuffy implements a discriminative statistical relational framework, applying MLN (based on binary truth values) as a graphical model. Each of these graphical models is a specific type of MRF [139].

Table 5.1: Comparison between PSL and Tuffy frameworks

Probabilistic Soft Logic (PSL)	Tuffy
PSL is a discriminative statistical relational framework, adopting HL-MRF as a graphical model	Tuffy is a discriminative statistical relational framework that adopts the MLN graphical model.
PSL uses explicitly defined inference targets, which are specified by the user. Therefore, additional work is required of the modeler.	Tuffy uses implicitly defined inference targets.
The grounding process is undertaken via bottom-up methods.	the grounding process is also implemented bottom up.

Another difference between the relational models built by the two systems is that in PSL, the grounding process is undertaken using a bottom-up approach, as with Tuffy, while the grounding process is conducted using SQL queries in both [140]. Another major point of divergence between these two frameworks relates to their inference targets. Probabilistic soft logic uses explicitly defined inference targets, which are specified by the developer/researcher. Therefore, additional work is required of the modeler. However, in Tuffy, the developer must define variable types for each predicate. This then allows the system to create implicit targets [140], which is an area that will be further developed in the next sub-section. Table 5.1 presents an alternative means of comparing the PSL and Tuffy frameworks, wherein details of the differences and similarities are highlighted.

5.3 The Importance of Priors in Logical Inference

In AI, and particularly in the field of fact-consistency checking, false claim detection, logic, etc., the word 'prior' is often used as a noun, rather than as an adjective. It denotes a specification (for example, a PSL rule or ground atom), which is set up before any normal processing is carried out. Priors are about prior domain beliefs in a model.

In this regard, [65] affirm that in PSL, a prior may be considered as a small weighted rule that generally applies a negative literal. The above researchers also support that if priors do not serve as potential functions, they should be defined over atoms and with weight distributed over the interval [0, 1]. Indeed, in PSL, priors are specified for a given predicate, and the absolute values of their weights are usually kept to a minimum. This is because they are initially in place, specifically so that they can be 'overpowered' by evidence [65]. Thus, atoms are most likely to have weak evidence, pointing towards zero. The use of an atom with a non-zero value will depend on the availability of proof of the atom's validity/existence. This is sufficient to 'overrule' its initial zero value. Nevertheless, an atom may be initialised with a low but non-zero value, which can be overruled when sufficient evidence is produced. For this to occur, simple priors may be adopted for overruling to be applied when evidence is available [65].

Thus, atoms are most likely to have an initial value that is equal to zero. The use of an atom with a non-zero value will depend on the availability of proof of the atom's validity/existence, which is sufficient to 'overrule' its initial zero value. Nevertheless, an atom may be initialised with a low but non-zero value, which can be overruled when sufficient evidence is produced. For this to happen, simple priors may be used for the overruling to be applied when evidence is available [65]. Two different types of priors can be used in PSL: negative and positive. However, in this study, the focus will be on negative priors alone. The reason why negative priors are preferred in this instance is that Web article searches for positive affirmations to disprove a negative prior are much easier to carry out than searches for refutations. For example, if it is decided to begin by specifying all the relations (which are already known to probably be true) as false in a PSL script, then the system's task will be to prove them to be true. Refuting a negative by proving a positive is much easier, in this situation, than proving that the refutation of a positive is impossible. Thus, the truth values of the ground atom values for all the predicates included in the PSL script are initially set to zero — indicating that these ground atoms are first assumed to be false. From experience, it has been found that negative priors are easier to use than positive priors [65, 141]. It is this fact that dictated the choice of negative priors in the present study.

An example of using negative and positive priors, specified for the spouse, S, relation (with regard to the UK royal family) is given below.
Taking two members of the UK royal family, Princess Kate Middleton and Prince William, f the following may be written: (1) `¬S(Princess Kate Middleton, Prince William)` or,
(2) `S(Princess Kate Middleton, Prince William)` For (1), which means that Princess Kate Middleton and Prince William are not spouses, the task will be f to find articles supporting the contrary, namely that they are spouses, whereby many such articles are likely to be found. Conversely, for (2), which means that Princess Kate Middleton and Prince William are spouses, the task will be to find an article that negates this claim, whereby it is likely that no such articles will be found. However, this lack of evidence to refute a claim does not really constitute strong evidence for affirmation; it is difficult to say in this case that the prior has been confirmed. Moreover, in the main processing stage, when checking for false claims, it is easier to assume that every claim is false, that is, to set up a negative prior for each claim and then look for evidence to the contrary.

5.4 Probabilistic Soft Logic (PSL) and Markov Logic Network (MLN) Inference

Probabilistic soft logic and MLN are both probabilistic logic frameworks that can be used to predict or generate new knowledge, where this was not previously present in an associated knowledge base. This knowledge base might also be known as observations in PSL and evidence in Tuffy for MLN. In general, a knowledge base or database containing relevant data (information) will enable the user to search queries that must be satisfied. However, it may happen that the information requested by a user is not directly present in the knowledge base. Thus, it is not easy and might even be impossible to answer the query. Consequently, it may be advantageous to use current information/knowledge items to build new information. Such an approach could necessitate the adoption of statistical learning methodologies, such as those implemented in PSL and MLN to infer new knowledge. Indeed, PSL and MLN both implement probabilistic logic frameworks that can handle this kind of task. Furthermore, in both cases, the task is performed through what is known as a knowledge graph. Building a knowledge graph first requires cleaning up the knowledge base, because this usually contains replicated/duplicated data. Secondly, it requires the creation of relations between these data. Once a useful knowledge graph has been built, the inference process can take place (in either PSL or MLN). Hence, new knowledge will be generated from previous knowledge stored in the knowledge base. In this manner, both PSL and MLN facilitate the inference of new knowledge, as described in the following sections.

5.4.1 Probabilistic Soft Logic (PSL) Maximum a Posteriori Probability (MAP) Inference

A probabilistic soft logic [69] script may be seen as implementing a logical framework, underpinned by a hinge-loss Markov random field (HL-MRF) inference model [65]. Controlled by a graphical representation, an HL-MRF is similar to a discrete Markov random field (MRF) [66]. However, PSL does not use discrete values but rather continuous variables, taking values in the interval [0, 1]. The related probability distribution is defined over the dependent variable (Y), given the independent variable (X) and using the following equation:

$$P(Y|X) = \frac{1}{Z(\lambda)} exp[f_\lambda(Y,X)] \tag{5.1}$$

Here $Z(\lambda)$ is a normalisation item and $f_\lambda(Y,X)$ is a constrained Hinge-Loss feature function.

$$f_\lambda(Y,X) = \sum_{r \in R} \lambda_j \phi_j(y,x) \tag{5.2}$$

with $\lambda \in (\lambda_1, ..., \lambda_m)$ being the respective weights of the potential $r \in R$. In turn, this relates to the importance of the respective potential (rule) in the model, while R is the set of all the potentials in the model. Moreover, $\phi_j(Y,X)$ are the potentials represented by hinge-loss functions, which make the model manageable, i.e. they increase the researcher's control over the model. PSL has been selected for use in this study as a specific relational domain, wherein the modelling techniques applied include a probabilistic method, supported by HL-MRF. In fact, PSL is a programming language that can be implemented relatively easily, and its results can be represented in HL-MRF in a straightforward manner. The logic of the developer's model may be specified via a first-order logic syntax, which can also be used to describe the features of the model that define an HL-MRF. The following is an example of a PSL logical rule:

$$w : P(X,Y) \wedge Q(X,A) \Rightarrow R(Y,A)$$

w represents the rule's weight. This determines the importance of the rule in relation to the other rules in the 'ruleset'. Primarily, however, the model is defined by its predicates (P, Q, R) and variables (X, Y, A). By way of example, it could be stated that where the P is '$friend$', then $friend(X,Y)$ simply means that the variables 'x' and 'y' are friends. A rule based on this relation would be:

$$1.0 : friend(X,Y) \wedge support(X,A) \Rightarrow support(Y,A)$$

Given that A represents a team, it is evident that this rule states that a person will logically/generally support the same team as his/her friend. Furthermore, it should be noted that in a logical rule, each predicate is an atom. Whenever an atom's variables are instantiated to constants, the atom becomes an instance or a ground atom, for example, support ('Tom', 'Manchester United'), with the ground atom truth value being defined in the interval [0, 1] (such as the PSL interval value domain). The mapping between the ground atom and truth value corresponds to the interpretation, I, implemented in PSL. Thus, it should be ascertained how these truth values will be processed. Here, Lukasiewicz logic is applied, in order to relax the logic for relationships between ground atoms in terms of the following operators: logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). For the case presented below, a and b are truth values and belong to the interval 0 to 1:

$$a \wedge b = max\{0, a+b-1\} \tag{5.3}$$

$$a \vee b = min\{a+b, 1\} \tag{5.4}$$

$$\neg a = 1 - a \tag{5.5}$$

The equations (5.3), (5.4) and (5.5) present the application of Lukasiewicz logic to the relaxation process, as employed in the potential function. Indeed, while relaxing the logical rules (see above and in the logical identity: $p \Rightarrow q \equiv \neg\, p \vee q$), a rule-grounded instance specified as: r ($r\_body \rightarrow r\_head$) is satisfied, meaning $I(r) = 1$, if and only if ($I(r\_body) \leq I(r_{head})$). Here, it is useful to recall that PSL creates a probability distribution over the interval [0, 1], which expresses the distance to total satisfaction of the rule in relation to the ground atoms. This distance to the rule's total satisfaction is represented by I and determined as follows:

$$I_r = max\{0, I(r_{body}) - I(r_{head})\} \tag{5.6}$$

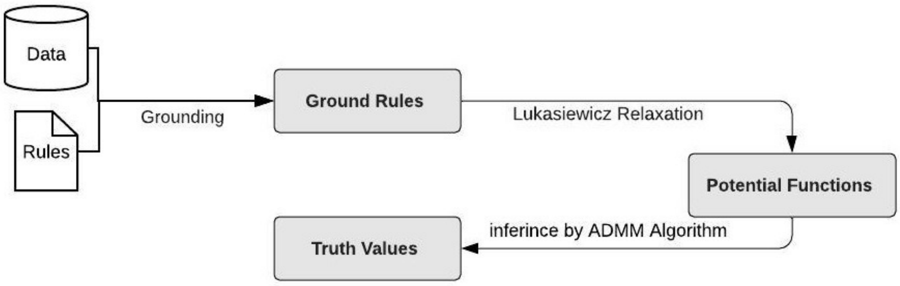


Figure 5.1: How PSL assigns truth values to random variables

The rules applied in the PSL model conform to an equation relating to a hinge-loss potential, which can be expressed as follows (5.6): the loss function (I) corresponds to the distance to the rule's total satisfaction, as expressed in (5.1). Specifically, the objective of optimising the loss function is to minimise the total weighted sum of the distance to the total satisfaction of every single rule. (5.1) is a log-concave equation based on Y; therefore, as a logical model, PSL can provide a solution to (5.1) by finding MAP for the HL-MRFs. Indeed, this may be achieved by applying a convex optimisation algorithm with continuous truth values. The issue of finding the MAP may also be regarded as the problem of finding the value of the dependent variable ($Y = f(x)$) for a function (f), given the values of the independent variable/observation (x). The solution to this convex optimisation problem may be found by applying the alternating direction method of multipliers (ADMM) technique, as adopted in [65].

Figure 5.1 presents the different steps involved in PSL truth-value assignment, that is, in assigning a truth value to a variable. The specifications given to the PSL model include a set of rules, augmented by several observations. The latter may also be considered as input data. In the holistic process, PSL begins by using the observations to ground the rules, thereby producing 'ground rules'. This first basic step is followed by relaxing the ground rules, whereby Lukasiewicz logic allows/supports PSL in assigning truth values in the interval [0, 1]. A convex potential function is generated as a result of relaxing each individual rule. Subsequently, this triggers a step wherein an overall potential function may be calculated by adding the average weighted sum of the rules applied. Finally, the maximisation process that follows is conducted using the ADMM algorithm [142].

5.4.2 Probabilistic Soft Logic (PSL) Lazy Maximum a Posteriori Probability (MAP) Inference

Lazy MAP is a mapping technique that existed before PSL was developed as a logic framework. In this study, lazy MAP was used in such a way that a MAP state was found without needing to give comprehensive details of all the targets (including the various combinations of claims, etc.). Instead, only the necessary target needed to be presented (on the lazy MAP) for the framework to be able to solve the inference problem [65]. Thus, lazy MAP was adopted to check the consistency of facts in this study, thereby determining the truth value (to be assigned to the claim in question). The claims to be checked were presented to the system by the researcher, but not all possible combinations, as in the more common type of MAP inference. The lazy MAP process takes its name from the low-level mapping, from which it is derived. In fact, the notion of lazy maps stems from the use of low-dimensional subspace (to which these lazy maps are confined). A set of potentials are found that are sufficiently small for all the other potentials to be zero in a MAP state. Therefore, lazy MAP inference can be performed efficiently with limited computing resources (time and memory). This can be seen from the following mathematical formula, which shows the distribution in relation to defining a MAP state [65]. If the subset ϕ of the index set 1, ...,m of the potential ϕ is considered, it may be understood that a feasible assignment to 'y' minimises y

$$\sum_{j\in\phi} w_j \phi_j(y,x) \tag{5.7}$$

and $\phi_j(y,x) = 0,\ \forall_j \notin \phi$,
Therefore, the assignment must be a MAP state, since zero is a global minimal for any potential.

5.4.3 Markov Logic Network (MLN) Maximum a Posteriori Probability (MAP) Inference

The most prominent challenges in the domain of machine learning are complexity and uncertainty. Indeed, at times, machine learning can be so complex that it becomes very difficult for the developer to describe or explain the model, or the results obtained, to a layperson. In addition, maximal uncertainty can surround the results of AI and machine learning in certain cases, such that it becomes necessary to support these results with specific performance measures, for example, confidence intervals and the level of error to be expected in the results [112]. This complexity and uncertainty often pertain to MLN, etc. In this study, a specific framework comprising probability and first-order logic was developed. Probability relates to the issues surrounding uncertainty, while first-order logic caters to the challenge of complexity. Several approaches to using probability and first-order logic have been proposed, including: relational dependency networks, stochastic logic programs, probabilistic relational models, relational dependency networks, and Bayesian logic programs [66, 112]. However, these models have tended to be quite complex and this has restricted their application.

The Markov logic network has emerged as a framework that can combine probabilistic graphical models with first-order logic, so as to undertake probabilistic learning and inference based on logic constraints [143, 144]. Thus, complexities are handled via logic, whilst uncertainty is handled via probabilities. Markov logic networks use MRFs or Markov networks as their graphical model. A Markov network is a graph with edges that tend to model desired local influences, and nodes that model random variables. In other words, the graph characterises a joint distribution for a set of variables X = {X1, X2, X3..., Xn}, having an indirect graph associated with certain functions. In particular, each node in the graph represents a variable, while the functions are represented by cliques in the graph. Specifically, a function is a non-negative function with real values, each of which depicts the state of a clique in the graph. The following expression can be used to represent the joint distribution in a Markov network:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(X_k) \tag{5.8}$$

Here,
Z is the partition function, i.e., $Z = \sum_{x \in X} \prod_k \phi_k \; (x_{\{k\}})$.
$x_{\{k\}}$ is the state of the Kth clique in the graph.

Conventionally, Markov networks are characterised as linear logic models where each clique in the graph can be replaced with the weighted sum of the features. This leads to the following formula:

$$P(X = x) = \frac{1}{Z} exp\left(\sum_j w_j f_j(x)\right) \tag{5.9}$$

According to Singla and Domingos [66, 145], a first-order knowledge base is a collection of formulas or sentences in first-order logic. It is important to note that four types of elements are used to construct these formulas: predicates, functions, variables, and constants. Constants refer to objects in the domain of interest, while functions characterise mappings from tuples of objects to objects. Meanwhile, relations between the attributes of objects, or the objects in a domain, are denoted by predicate symbols, and a first-order knowledge base constitutes a set of hard constraints, whereby violation of any formula can lead to zero probability. However, in PSL, the underlying Markov logic is specified by the statistical relational language based on Markov networks, whereas first-order logic is specified to handle the constraints. Here, a violation of a formula in the knowledge base will result in reduced probability, rather than, necessarily, the yielding of zero probability [137]. Therefore, each formula has an associated weight, representing the strength of the constraint – the higher the weight, the bigger the difference and impact [66, 145].

An MLN is regarded as a first-order knowledge base, wherein weights are associated with each formula and are used as the template to develop Markov networks. An MLN is capable of providing compact specifications of knowledge for large, complex Markov networks, while reducing brittleness, tolerating imperfection, and dealing with contradictory knowledge by applying first-order logic [112, 146]. An MLN specification is predominantly a set of pairs (F_i, w_i), where w_i denotes a real number, and F_i denotes a formula of first-order logic. The corresponding Markov network $M_{L,C}$ is defined with (F_i, w_i) and the set of constants, C= $\{ c_1, c_2, c_3, C_c \}$ [112, 146].

Thus, the probability distribution defined by the ground Markov network $M_{L,C}$ is given as follows:

$$P(X=x) = \frac{1}{Z} exp\left(\sum_{i=1}^{F} w_i n_i(x)\right) \tag{5.10}$$

Here,
F is the number of formulas in first-order logic.
w_i is a real number, showing the formula weight. n_i is the total number of times that the formula is satisfied by the state of the world (x).
Z is regarded as the normalisation constant, which is needed to make all probabilities sum to 1.

It is important to note that MLN, because it uses the discrete MRF as its graphical model, presents its results simply as 0 or 1. To understand this situation, a specific example should be considered. Thus, a model is assumed to have two variables: vteach and vprof. The latter (vprof) indicates whether or not an individual is a professor, while the former (vetch) denotes whether or not an individual teaches. Since the representation of an MLN model is usually in the form of 'p' binary random variables; the dependencies between the model variables are depicted as a set of weighted constraints, which are defined over them [143]. For example, the statement, 'if an individual is a professor, then he teaches or vice versa' is a constraint. A bitwise-AND predicate, vprof ∧ vteach can be used to represent this constraint [143]. In the given case, the MLN framework considers a weight ('w'), which increases the probability of vteach and vprof, simultaneously (i.e. either vprof = 1 and vteach = 1 or vprof = 0 and vteach = 0). Hence, the result of the MLN processing will be 1 or 0.

5.5 Experimental Case Study: Fact-checking of UK Royal Family Relations Using Probabilistic Soft Logic (PSL) and a Markov Logic Network (MLN)

The case study reported in this section is based on two semi-related experiments, run on two closely related datasets. The first dataset that was built contained all the relevant UK royal family kinship relations (containing 198 claims or statements about these relations). Meanwhile the second was a subset of the first, containing only 30 claims. Four models were run on these datasets: FACT (PSL), using PSL as its probabilistic logic framework; FACT (MLN), using MLN as its probabilistic logic framework; the CredEye system, and random assignment. The inclusion of MLN served as a limiting factor for the size of the dataset. Indeed, this was the reason why a subset of the full UK royal family dataset was used in the second experiments. Nevertheless, it is important to mention that the objective pursued in this experimental case study was to identify which of the four models was more efficient for fact-checking. In order to achieve this, there needed to be a common environment in which these models could be tested. Thus, the UK royal family relation model was set up, as presented below.

5.5.1 Set-up of the Family Relation Model

In this sub-section, the PSL and MLN models are considered, as they relate to observations, which were the relations stored in the knowledge base. It is important to recall that the information extraction process required to build the knowledge base was dealt with in this chapter 4. Once these relations were made available in an appropriate format, it was easier to investigate these family-tree relationships. For instance, the interest in this current study lay in the various properties of the constants that designated people in the relations. Presented below are examples of the predicates (property specifications) used in the model's logic rules. The various predicates used to build the proposed kinship model are shown below: $Male(X)$ signifies that X has the property 'male', and $Female(X)$ signifies that X has the property 'female'. Furthermore, $Parent(X,Y)$ signifies that X is the parent of the child (Y), and $Spouse(X,Z)$ signifies that there is a marital relationship between X and Z, one being the husband and the other the wife. Once there was a valid model of this kind, representing the kinship relations of the UK royal family, it could be ascertained whether the information extraction tasks were functioning appropriately, and whether the instances extracted had been effectively processed by the models (PSL and MLN) constructed in this research.

The predicates indicated below are conditional rules, which can be processed by PSL and MLN to infer unobserved relations. The predicates representing these relations (which were not extracted from the corpus and therefore, were never 'observed') are presented in Table 5.2, with gendered and un_gendered family relations being clearly distinguished. Below are a number of rules relating to the observed and unobserved predicates. These rules allow relations to be inferred directly from observations and facilitate the encoding of a comprehensive range of family-tree relations, as defined in the previous chapter (chapter 4 Figure 4.1). In the family tree, relations were employed as the bases of inference and prediction. For the sake of brevity, only selected family-tree relations are presented below, while the rest are shown in the Appendix D:

$$Parent\_Of(X,B) \wedge Parent\_Of(X,A) \wedge A \neq B \Rightarrow Sibling\_Of(A,B)$$
$$Parent\_Of(X,B) \wedge Parent\_Of(Y,A) \wedge Sibling\_Of(X,Y) \Rightarrow Cousin\_Of(A,B)$$
$$Parent\_Of(X,B) \wedge Sibling\_Of(X,Y) \wedge Female(Y) \Rightarrow Aunt\_Of(Y,B)$$
$$Parent\_Of(X,B) \wedge Sibling\_Of(X,Y) \wedge Male(Y) \Rightarrow Uncle\_Of(Y,B)$$
$$Parent\_Of(X,B) \wedge Sibling\_Of(X,Y) \wedge Male(B) \Rightarrow Nephew\_Of(B,Y)$$

The family relations listed above may be described as follows:
i) The sibling relation, where the rule signifies that if X is the parent of B and A, with A and B being different individuals, then B and A are sibling(s).
ii) The cousin relation, where the rule signifies that if A and B are cousins, and X is the parent of B and Y is the parent of A, then X and Y are siblings.
iii) The aunt relation, where the rule signifies that if X is the parent of B, and X is the sibling of Y and Y is female, then Y is the aunt of B.

Table 5.2: Inferred predicts

Un_gender Predict	Gender Predict
Ancestor_Of(X, Y)	—
Descendent_Of(X, Y)	—
Sibling_Of(X, Y)	Sister_Of(X,Y) Brother_Of(X,Y)
Parent_Of(X, Y)	Mother_Of(X,Y) Father_Of(X,Y)
Child_Of(X, Y)	Daughter_Of(X,Y) Son_Of(X,Y)
Spouse_Of(X, Y)	Wife_Of(X,Y) Husband_Of(X,Y)
Uncle_Of(X, Y)	—
Uncle_In_Law_Of(X, Y)	—
Aunt_Of(X, Y)	—
Aunt_In_Law_Of(X, Y)	—
Niece_Of(X, Y)	—
Niece_In_Law_Of(X, Y)	—
Nephew_Of(X, Y)	—
Nephew_In_Law_Of(X, Y)	—
Cousin_Of(X, Y)	—
Cousin_In_Law_Of(X, Y)	—
Child_In_Law_Of(X, Y)	Son_In_Law_Of(X,Y) Daughter_In_Law_Of(X, Y)
Parent_In_Law_Of(X, Y)	Mother-In_Law_Of(X,Y) Father-In_Law_Of(X, Y)
Sibling_In_Law_Of(X, Y)	Sister_In_Law_Of(X,Y) Brother_In_Law_Of(X, Y)
Grand_Child_Of(X, Y)	Grand_Daughter_Of(X,Y) Grand_Son_Of(X,Y)
Grand_Parent_Of(X, Y)	Grand_Mother_Of(X,Y) Grand_Father_Of(X,Y)

iv) The uncle relation, where the rule signifies that if X is the parent of B, and Y is the sibling of Y and Y is male, then Y is the nephew of B.
v) The nephew relation, where the rule signifies that if X is the parent of B, and X is the sibling of Y,and B is male, then B is the nephew of Y.

As previously mentioned, priors represent the system's initial 'beliefs'. These are also the rules applied to open predicates. There are two types of predicates: open and closed, as explained in section 5.3. In this research, negative priors were applied. This was intended to facilitate

the process of truth-value detection and assignment by the PSL model. It should be noted that priors were only applied in the PSL model; they have unique importance for this model because it operates by distributing the truth value across the interval [0, 1]. Conversely, MLN is a binary-based system that does not make use of priors. In this experiment, each absolute logic rule was assigned a weight of 1.0. The negative priors (set of rules) were included in the model as negative, with a low weight (0.1). Therefore, these would be easily out-weighed by the other rules and by observations. These priors consisted of the initial beliefs built into the model [65]. Examples of some of the negative prior rules are presented below:
$Sibling\_Of(A,B) = 0$
$Cousin\_Of(A,B) = 0$
$Aunt\_Of(Y,B) = 0$
$Uncle\_Of(Y,B) = 0$
$Nephew\_Of(B,Y) = 0$
These are some of the rules and negative priors used to build the knowledge base for UK royal family relations.

5.5.2 Fact-checking of UK Royal Relations Using Probabilistic Soft Logic (PSL)

One of the main objectives pursued in this case study was to establish and measure the effectiveness of the selected research methodology. This was an exciting task when one considers all the apprehension and uncertainty surrounding it. Furthermore, the established end goal would make this effort more enjoyable. The ground truth set accumulated in the study, encompassing N claims relating to the UK royal family, was one of the main assets deployed in this case study, built according to the UK royal family tree [1].

As mentioned in Chapter 4, the ground truth dataset contained 99 true claims and 99 false claims, regarding the UK royal family's kinship relations. It was first built by randomly selecting a name from the UK royal family's list of names, and then randomly matching this with an element from the list of kinship relations. The result of this first combination was again matched with a second name, randomly selected from the list of names of members of the UK royal family (with the original name excluded). The random relation between these two family members then became output as a claim. This process was performed iteratively until a sufficient number of claims had been generated. An example of a false claim generated as outlined above was: 'Prince George is the sibling of Prince William.' With regard to building the 99 true claims, the UK royal family tree was used directly. Hence, the latter claims were not randomly built but were rather factual. For instance, 'Prince William is the father of Prince George' is an example of a true claim. In this manner, the ground truth set of 198 claims was built to test FACT (PSL), FACT (MLN),

[1] Royal Family tree and line of succession:
https://www.britroyals.com/royaltree.asp

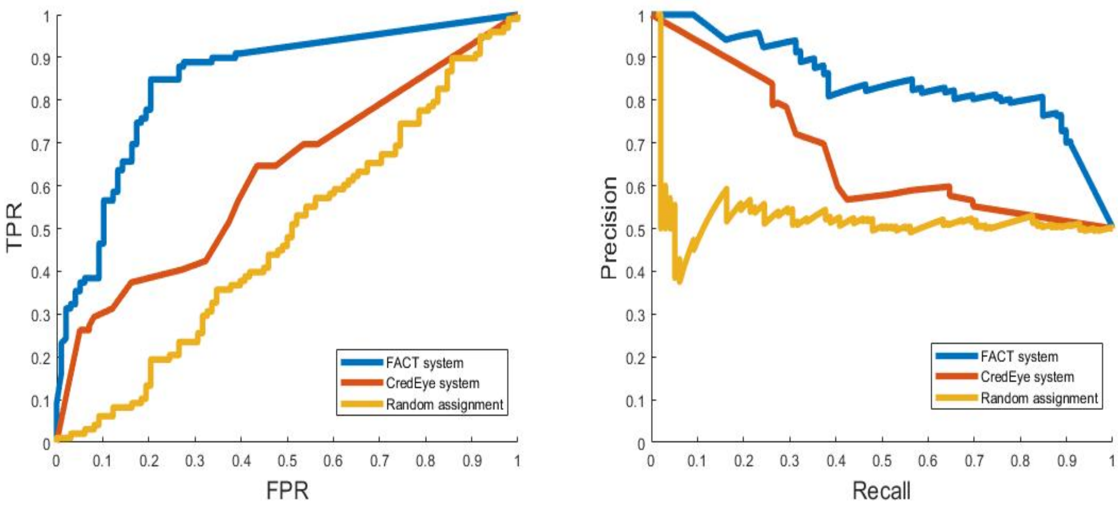
and the CredEye system. In this study, FACT (PSL) was the FACT approach adopted for use with PSL as its logical reasoning, and FACT (MLN) was the FACT approach using Tuffy as an MLN inference engine. An example of a claim based on the ground truth set is shown in Chapter 4.

The ground truth set used in this study had an internal structure that conferred an embedded structural hierarchy. Thus, it was easy to examine the file in terms of familial hierarchies (for example, from a parent to great-grandchildren). The ground truth set can be found in Ground Truth Set Link). The ground truth set, as described above, was added to the logical reasoning models (PSL and MLN). This was sufficient to start running and testing the model to determine the truth values yielded by the various combinations (as described above). The results were evaluated in terms of the number of true claims, i.e. claims that were consistent with the information stored in the knowledge base, built as described in the previous chapter (Chapter 4), and with the model's rules. Particular attention was paid to the number of claims that ended up with truth values very close to zero (0), meaning that they were either false claims or there was insufficient information in the knowledge base to support them.

In order to compare the fact-checking model created in this study with the CredEye system [2], an experiment was run that applied both models (FACT (PSL) and CredEye) to the ground truth set. This compared the truth values generated by FACT (PSL) with the probabilistic values returned by CredEye. The CredEye system has an interactive user interface, which facilitated the automatic generation of the probabilistic credence value of a claim, employing the usual stages of searching the Web for the articles required to test the claim's consistency, and then testing for consistency. These stages included crawling and scraping the Web to extract useful articles relating to the claims. The CredEye method also included checking the language style and stance of the articles, along with the trustworthiness of their underlying sources. The aim of this was to judge the truthfulness of the given claims, while providing some evidence to validate that judgment. The result was a value that indicated the probability, according to the system, of the input claim being true. Thus, this probability value would differ slightly in kind from the result provided by PSL.

Consequently, by looking at both the probability value returned by CredEye and the truth value generated by PSL, it was possible to compare the performance of the two models in relation to the same claim. Conversely, given that negative priors were used in this case study, it must be remembered that the values allocated to the claims were all set to zero (0) from the start. In addition, after running FACT (PSL), the values obtained were expected to stay very close to zero (0) for false claims and to increase to one (1), or very near one, for true claims. In contrast, the CredEye system would return values that were more widely distributed over the interval [0, 1] in response to the various claims. However, ultimately, the result was judged in relation to the claims that each system, in effect, classified as either true or false. Therefore, a true or false

[2] Credibility lens for analysing and explaining misinformation.:
https://gate.d5.mpi-inf.mpg.de/credeye/



(a) ROC curves. (b) Precision and Recall curves.

Figure 5.2: Performance figures: blue line represents results yielded by the FACT (PSL) system; red line represents results of the CredEye system, and yellow line represents results of random assignmen

value was assigned to each result from each system, based on whether the result was nearer to zero (0) or one (1). Finally, the 'random assignment' method was applied to the same claims as an experimental control, and random truth values in the interval [0, 1] were assigned to each of the claims. This was repeated 1000 times. The average truth value was calculated for each claim, and this was returned as approximately 0.50 for each.

The overview of the experiment outlined above required both models to be run several times on the ground truth set (99 true and 99 false statements). Once a sufficient number of results had been accumulated, receiver operator characteristic (ROC) curves could be calculated. These were used to compare the models by measuring their respective fact-checking abilities. Figure 5.2a presents the ROC curves for both the FACT (PSL) and CredEye systems. It is important to recall that a ROC curve is a plotting of the true positive rate (TPR), given the false positive rate (FPR). The former is plotted on the y-axis and the latter on the x-axis but for different truth values, with respect to the ground truth set and corresponding to various tunings of the thresholds. With regard to FACT (PSL), the TPR corresponded to the ratio of the number of claims that were correctly classified as true (true positives), divided by the total number of claims that were actually true. Conversely, the FPR corresponded to the ratio of the number of false claims classified as true (false positives), divided by the total number of claims that were actually false. Finally, the performance of the model was evaluated using the area under the curve (AUC) of the ROC curve (AUC-ROC). The result obtained for the FACT (PSL) system was 0.79 (Figure 5.2a FACT (PSL) line), while the CredEye system achieved only 0.62 (Figure 5.2a CredEye line).

These models, in addition to comparing them with each other, were assessed using an evaluation method based on the null hypothesis, whereby producing a random truth value for each element of the ground truth set would lead to plotting the first diagonal. In theory, this would separate the whole area into two halves, with 50% corresponding to the AUC, as shown in Figure 5.2a. The line generated using the random assignment method, representing the means of providing a null hypothesis evaluation in this study, did in fact delineate an area below it of 49% of the total area. This was very close to the theoretical expectation of 50%, as described above. For the purposes of comparing the predictive model to this random predictive model, the null hypothesis test was repeated 1000 times on the ground truth set. The average and best performance results generated from all the models were recorded. The p-values obtained with respect to the random model were < 0.001, indicating the statistical significance of the results. In turn, this showed that the model performed better than the random model in terms of predicting correct truth values.

In addition, using the 'by-chance' rate as a control, the model created in this study performed better than the CredEye system, as the latter scored only 0.12 above the 'by-chance' rate, while the current model scored 0.29 above the 'by-chance rate' (thus, more than double). This means that the current model was twice as efficient as CredEye, compared to the by-chance rate. Therefore, it may be stated that FACT outperformed CredEye. It could also be posited that this was because FACT was able to check claims related to facts that were not explicitly mentioned in the text. In contrast, the CredEye system was unable to construct facts based on inference, and was therefore unable to check claims against inferred facts. These results answer one of the subsidiary questions raised by the research question, concerning the relevance of checking facts that are not explicitly stated in a text corpus, but must be inferred, to the correct verification/rejection of claims. In summary, it may be stated that FACT outperformed CredEye, because of the former's ability to check claims in relation to inferred facts that are not explicitly mentioned in the text.

Other metrics used to compare the two methods were precision and recall [147, 148]. In this respect, the following definitions were applied in this study. Precision was defined as the number of true claims identified (true positives), divided by the total number of true and false claims identified (true positives and false positives). In other words, it represented the percentage of claims that were correctly identified as true among all the claims that were classified as true. Conversely, recall represented the number of true claims identified (true positives), divided by the total number of claims that were correctly identified (true positives and total number), this being the TPR, as defined above. For both models, precision and recall were calculated, and these values were used to plot their representative curves (precision given recall) (see Figure 5.2b for the curve representing FACT, and Figure 5.2b for the curve representing CredEye). These plots enabled the models to be ranked according to their predictions. Once more, the current model was observed to have outperformed CredEye. Indeed, the curve representing FACT in Figure 5.2b may be observed as well above the curve representing CredEye in Figure 5.2b. This implies that the area beneath the former was greater than the area beneath the latter. In addition, in statistical terms, it is important to compare both of these models with the random assignment model. This comparison revolves around a theoretical straight line at precision = 0.5, where $0,5$ of the area beneath this theoretical line roughly coincides with the value yielded by the measurement (precision) calculation, based on chance truth assignment. This is illustrated in the random assignment plot (see Figure 5.2b). According to this comparison, the FACT (PSL) system also outperformed CredEye.

Therefore, to summarise, it is clear that in this case study, the model created was found to outperform both the random interval value [0, 1] assignment and the CredEye system. This case study also helped answer one of the research questions, namely, on the significance of FACT (PSL) being able to use information that is not explicitly stated in the text corpus, but implicitly present within it, as determined by fact extraction and probabilistic reasoning. The above concludes, for now, the discussion of the experiments using FACT (PSL) in this thesis.

After conducting the experiment using FACT (PSL), a similar experiment was conducted with FACT (MLN) and the results were compared.

5.5.3 Comparative Study of the Probabilistic Soft Logic (PSL) Fact-checking Model versus the Markov Logic Network (MLN) Fact-checking Model

This sub-section describes a number of experiments that were similar to those described above. However, the current fact-consistency checking system was implemented with different probabilistic reasoning (MLN) and only a subset of the ground truth set. The MLN grounding process requires a much larger memory resource than the PSL grounding process. Thus, due to the time and memory limitations of the available resources, the only practicable option was to reduce the size of the test dataset (i.e. the ground truth set) provided to the framework [140]. Only 30 claims were included in the ground truth set for this experiment: 15 true claims and 15 false claims related to the UK royal family and based on the UK royal family tree [3]. The 30 claims were selected in such a way that all the different relations involved in the study were represented, including siblings, sisters, fathers, etc. Thus, the subset of 30 claims was built in readiness to be inputted into FACT (MLN), FACT (PSL), and the CredEye system.

The 30-claim subset was added to both the PSL and MLN models as a target, enabling the models to be run to determine their performance in identifying true and false claims. Also run were CredEye and random assignment on the same set of 30 claims. In FACT (MLN), the various constructed relations only returned truth values — values belonging to the set of binary numbers 0, 1, that is, either true (1) or false (0). The final result for each claim was likewise either true or false (as opposed to the corresponding results from FACT (PSL), which were either a near 1 or near 0 probability of truth). Therefore, it was possible to base the evaluation of this method on the number of true positives, etc., as before. However, a positive was designated simply by

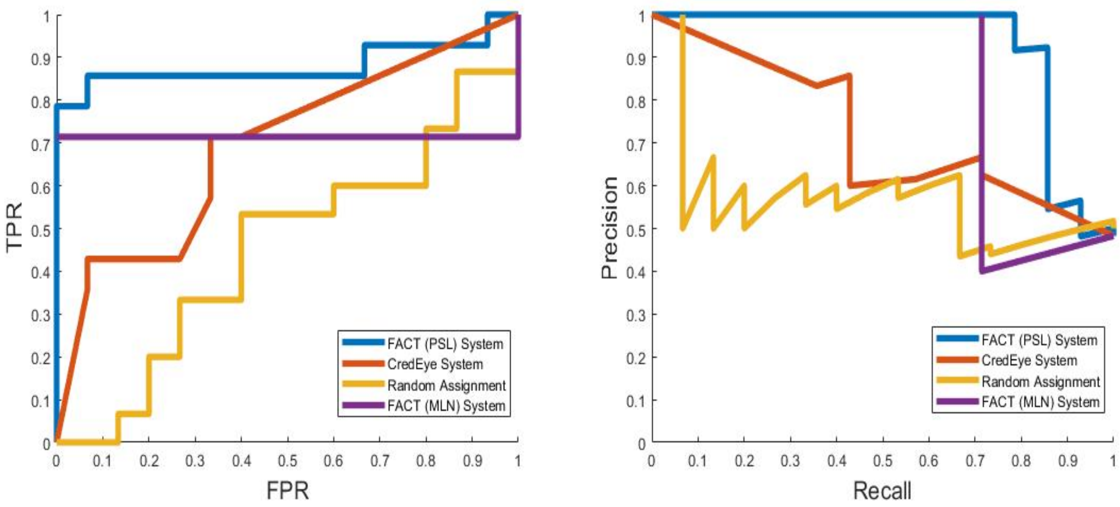
[3]Royal family tree and line of succession:
https://www.britroyals.com/royaltree.asp

the value 'true' (indicating, that the claim was true in relation to the knowledge base (chapter 4) and the rules of the model). In contrast, for the CredEye system, as previously, the results were expected to reflect the probability of each claim's credibility (section 5.5.2). Moreover, as before, such a high probability was taken to mean that the system considered the claim to be true, while a low probability meant a claim being considered as false. Finally, in terms of random assignment, random truth values were attributed, as explained in the PSL sub-section above (section 5.5.2). In this manner, the four models (FACT(PSL), FACT(MLN), CredEye, and random assignment) were ranked according to their ability to classify the claims as either true or false.

The outcome of all these tests, using the subset, was the generation of ROC curves, which were then used to compare the fact-checking performance of all four models. Figure 5.3a presents the ROC curves for FACT (PSL), FACT (MLN), random assignment, and the CredEye system. These ROC curves were constructed using TPR, given the FPR. The final performance metrics obtained, based on the precedent calculation method defined above in the PSL sub-section (section 5.5.2), and the results obtained from the AUC-ROC, were as follows: FACT (PSL) = 0.88, FACT (MLN) = 0.71, the CredEye system = 0.62, and random assignment = 0.50 (Figure 5.3a FACT line), and CredEye (Figure 5.3a CredEye line). Given these values, it may be concluded that FACT (PSL) outperformed FACT (MLN) and in turn, outperformed CredEye, which likewise outperformed random assignment.

These four models could also be usefully assessed with respect to the null hypothesis. Producing random truth values based on the ground truth subset would, theoretically, result in plotting the 'first diagonal' line, splitting the area of the graph in half. This is illustrated in Figure 5.3a. As expected, the line plotted from the results of the random assignment model, with regard to the ground truth subset (30 claims), was very close to the line described above, giving an area under the line of 49%. This was nearly equal to the theoretical expectation of 50%. For the purposes of comparing the predictive models in this study (FACT (MLN) and FACT (PSL)) to this random assignment predictive model, the null hypothesis test was repeated 1000 times on each of the models. The recorded values were the average of the best performance results generated during this experiment. Meanwhile, the p-values obtained were < 0.001, which is less than 0.05, indicating the relatively strong statistical significance of the results. Moreover, this result shows that the current models performed better than the random assignment model in predicting truth values.

As per the other baselines, the current models performed better than the CredEye system in relation to the by-chance rate. For example, CredEye scored only 0.19 above the by-chance rate, while FACT (PSL) scored 0.38 above, and FACT (MLN) scored 0.21 above. It may be noted that the FACT (PSL) score was more than double the CredEye system score (above the by-chance rate), while the FACT (MLN) score was only just above. It could certainly be concluded that the FACT models performed better than the CredEye system, using the by-chance rate as a baseline. Additionally, it may be confirmed that the current models achieved this performance by being



(a) ROC curves.

(b) Precision and Recall curves.

Figure 5.3: Performance figures: blue line represents results yielded by FACT (PSL); red line represents results of the CredEye system; purple line represents results of FACT (MLN), and yellow line represents results of random assignment

able to infer knowledge that was not explicitly stated in the text corpus. However, the CredEye system was unable to do so, as mentioned earlier (subsection 5.5.2). These results represent an answer to one part of the research question, focused on the extraction of knowledge that is not explicitly stated in a text, but implicitly present within it, using probabilistic reasoning based on fact extraction. Both the FACT models achieved better results than the CredEye system and random assignment, at least partly due to their ability to check claims against facts that were not explicitly mentioned in the texts.

As above, precision and recall were applied as appropriate metrics to compare the four models in this study. In this respect, precision and recall were calculated, and the values obtained to plot the precision given recall curves were used (see Figures 5.3b for the FACT curves, and Figure 5.3b for the CredEye curve). These plots also ranked the models according to the predictions made, where the two FACT models outperformed the CredEye system and random assignment. Here, the curve representing FACT (PSL) was above the curve representing FACT (MLN), and this was above the curve representing the CredEye system. Moreover, the difference between these last two curves was much greater than the difference between the first two. The curves for these three models appeared above the curve generated for random assignment (see Figure 5.3b). This implies that the AUC for FACT (PSL) was larger than the AUC for FACT (MLN), and the latter was larger than the AUC for the CredEye system, which was in turn larger than the AUC for random assignment. It should be noted that the CredEye system performed only marginally better than random assignment. Its resultant curve was very close to the theoretical curve expected of random assignment, that is, a straight diagonal line cutting the area of the graph exactly in half, with 40.5 of this area beneath it (see Figure 5.3b). The precision given recall plot for FACT (PSL) demonstrated a much greater accuracy in classifying claims within the ground truth subset, as compared to FACT (MLN) and the CredEye system.

Finally, these results show that in this third experimental case study, the FACT (PSL) and FACT (MLN) models outperformed both the CredEye system and random assignment. In addition, this result is in line with the preceding results, thereby enabling an answer to the part of the research question that relates to the extraction of information that is not explicitly stated, but is rather implicitly present in a set of documents. Thus, by using fact extraction and probabilistic reasoning, as in PSL and MLN, improved results can be expected.

5.6 Experimental Case Study: Fact-checking Political Relations Using Probabilistic Soft Logic (PSL)

Results are presented here that also make use of PSL, but with different types of relations. These relations were obtained in another study where subject-verb-object (SVO) triplets were used, rather than relations extracted with JAPE grammar [149]. The author's only contribution to the above-mentioned work involved the use of PLS and not the extraction of relations, but this is reported here as another example of PSL for NLP relations. In this study, political relations were inferred and fact-checked among actors in a political network, generated from 130,213 English-language news articles about the 2012 US Elections. This involved fact-checking supporting or opposing views of political actors, with regard to other actors and issues. Data collection was performed via the extraction of news articles using a modular media content analysis system [150]. This system contained US and international media. A topic classifier was also trained to classify articles on the Elections.

5.6.1 Building the Knowledge Base with Facts

In this case study, JAPE grammar was not used to build the knowledge base, unlike in the previous case study on family relations. Instead, different NLP pipelines were adopted by means of extracted triplets, created by [149]. In [149], SVO triplets were extracted from the Elections news collection via a fully automated pipeline, which detected named entities, co-referencing, and anaphora resolution before triplet extraction. In the triplets, subjects and objects were the named entities or noun phrases (issues), and the verb expressed a positive or negative attitude between the subjects and objects in the political discourse. The number of triplets was reduced after filtering for high confidence triplets. The latter were used to create positively and negatively weighted relations between actors. Positive and negative verb lists were deployed to count triplets as a vote in favour of a positive or negative attitude, and to calculate a weight for the relation between actors. Verb lists denoting political support/opposition were then manually created by going through actions in the triplets extracted from the Elections corpus. When quantifying the weight of a relation between actors a and b, a confidence interval [151] around the estimate of the value was also considered. Based on computed confidence intervals, relations were extracted that were sufficiently supported by the corpus. Positive and negative weights were likewise calculated and used to assemble a network consisting of nodes that represented actors/issues, and edges that represented weights, ranging from [-1 +1]. From this network, structural balance [152] rules were applied to infer political relations among actors and between actors and issues, using PSL.

Structural balance can, at most, provide the plausibility of a claim, as it is not an exact relation like a family relation. An inferred political relation will have a weight that corresponds to the level of support or opposition between actors in the relation conveying its plausibility.

5.6.2 Set-up of the Political Relations Model

In order to prove that political relations could be inferred among actors from the network, a few links were removed from the network, and the remaining relations were used to predict the removed links. The aim here was to ascertain whether the removal of 5%, 10%, or even 20% of the links from the network would still enable them to be inferred from the remaining observed relations. Since truth values had been obtained for the removed links, the system's performance was also evaluated. The network selected for this study contained 169 nodes and 238 links with weights in the interval [-1, 1]. To render this appropriate for the PSL framework, the weights were normalised to [0, 1] intervals. First, the number of links to be removed from the network was carefully established. This involved links that connected nodes with a degree greater than or equal to 2, so that no singletons were introduced into the network when the links were removed. In total, 126 links were quantified as removable.

Next, 5% (12 links), 10% (24 links), and 20% (48 links) of the links were removed from the entire network, having been randomly selected from the 126 removable links that were identified and predicted by PSL. The logical rules created to predict the links were based on structural balance theory [152], with a binary predicate *Rel* (relations between actors). Several logical rules are listed below.

$$
\begin{gathered}
Rel(A,B) \wedge Rel(B,C) \Rightarrow Rel(A,C) \\
Rel(A,B) \wedge \neg Rel(B,C) \Rightarrow \neg Rel(A,C) \\
\neg Rel(A,B) \wedge Rel(B,C) \Rightarrow \neg Rel(A,C) \\
\neg Rel(A,B) \wedge \neg Rel(B,C) \Rightarrow Rel(A,C)
\end{gathered}
$$

The first four rules were adapted to structurally balance intransitive triads in the network. These rules stated: 'a friend of my friend is my friend', 'a friend of my enemy is my enemy', 'an enemy of my friend is my enemy', and 'the enemy of my enemy is my friend.'

Since the truth values for the predicted links were also known, in each case, the mean absolute error (MAE) over all the predicted links was measured in 100 iterations. The overall

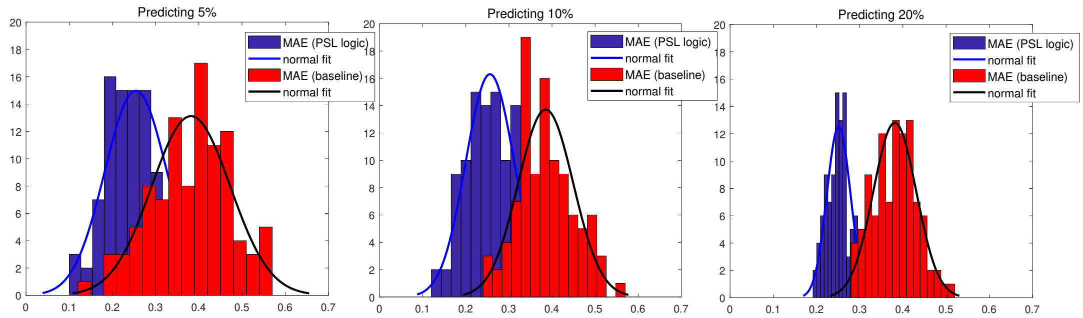


Figure 5.4: MAE distribution, with normal fitted curve over 100 iterations for predictions with PSL and random (baseline), when predicting 5% (left), 10% (middle) and 20% (right) of links from the network

MAE predictions were given by,

$$MAE = \frac{|\, y^i - x^i \,|}{n} \tag{5.11}$$

where y^i refers to the prediction of the i^{th} link, x^i refers to the truth value of the i^{th} link, and n is the total number of links predicted.

Mean absolute error was computed over 100 iterations when removing 5%, 10%, and 20% of the links from the network and predicting them with PSL. To compare this with a baseline and prove its superiority to random prediction, in each experiment, a value was randomly selected as the prediction from the entire link weight distribution of the network, and MAE was computed as before. Figure 1 shows the MAE distribution with a normal fitted curve over 100 iterations for predictions with PSL, and random predictions (baseline) when predicting 5%, 10%, and 20% of the links from the network. The most common MAEs lay in the range of 0.19-0.27 (5%), 0.22-0.28 (10%), 0.25-0.28 (20%) for PSL predictions, and 0.33-0.41 (5%), 0.33-0.39 (10%) and 0.34-0.39 (20%) for the baseline. Therefore, the test revealed that PSL performed better than random prediction of relations.

5.6.3 Fact-checking Political Relations Using Probabilistic Soft Logic (PSL)

Once proven that political relations could be inferred given a set of political relations between actors, it was possible to check facts about political relations via these means.

For example, given a claim/fact such as,

Claim: ‘Hillary Clinton opposes abortion’

the system would add this fact to the PSL target list and run the inference process to fact-check the truth. The weight of this relation could be assigned to 0, since ‘oppose’ was a negative verb in the Elections context, and the most negative weights were mapped to 0 values in PSL. The target would be comprised of Hillary Clinton, abortion, and the negative weight associated with the relation.

Target: (Hillary Clinton, abortion)
Claim Weight: 0.0
Inferred Weight: 0.85
Verdict: = 0.0 / False

The inferred weight for the given target was 0.85, indicating that there was reasonably high support for abortion from Hillary Clinton. Compared to the weight of the claim (0.0), the system returned the verdict 'False'. It would likewise be possible to reason out this decision, stating that Hillary Clinton supports Obama and Obama supports abortion; therefore, Clinton supports abortion. This violates the first logical rule given to PSL, which is that if A supports B and B supports C, then A supports C.

5.7 Summary

In this chapter, implementation of the basic model in the study was demonstrated on two probabilistic reasoning frameworks. Both these frameworks measured the consistency of a given claim relating to the UK royal family tree, comparing the claim to knowledge within a knowledge base (built as described in the previous chapter (chapter 4). The truth values representing the consistency of the claim allowed these claims to be classified into either a true or false claim category. In addition, the framework was applied to political relations using PSL alone.

To achieve this objective of checking the consistency of facts through probabilistic reasoning, the language and syntactical structure of PSL was deployed. Probabilistic soft logic is a notable statistical relational framework, which facilitates the construction of models based on weighted first-order logic. Probabilistic soft logic scripts contain formulas that apply variables, functions, constants, and predicates. Probabilistic soft logic can also be thought of as a programming language that includes template definitions to handle constraints and potentials.

Having implemented the model in this study using PSL, an attempt was made to employ Tuffy for the same purpose. Tuffy's main inference engine is MLN, which offers high levels of scalability, albeit tempered by the fact that it can be resource hungry, taking predicates and rules as input. The two models were subsequently compared, looking at the similarities and differences between PSL and Tuffy, the main difference being that PSL uses distributed truth values over the interval [0, 1] as results, while MLN uses binary truth values 0, 1.

In addition, the importance of priors was highlighted in the logical inference method employed in this study. Moreover, lazy MAP inference in PSL was applied to facilitate the task of target definition, via such means as minimum combinations, which served as targeted claims. In turn, this ensured that less memory space was used. Such prudence was necessary for better time management, as well as to determine the fact-consistency of truth values across a reasonably large set of claims. Additionally, complexity and uncertainty were elaborated upon in terms of machine-learning research. Here, the choice of algorithms to address these issues was crucial, for example, probabilistic logic systems such as PSL and MLN. In addition, the use of performance measures that could competently support the results obtained was important. Moreover, the results were analysed using p-values (< 0.001).

Four different models were deployed to check claims relating to the UK royal family's kinship relations: FACT (PSL), the FACT approach as implemented in the PSL framework; FACT (MLN), the FACT approach as implemented in the Tuffy framework; CredEye, and a random assignment model used as a control. A dataset of 198 claims was generated to exercise the FACT (PSL) system in its entirety, and compare it with CredEye and random assignment. A subset of 30 claims was then used to compare all four of the above-mentioned models. This manoeuvre was necessary because FACT (MLN) could not be run on a larger set of claims within a reasonable processing time or memory constraints. Finally, the precision measure was plotted against recall for all runs. The results for the 198-claim dataset showed that FACT (PSL), with an AUC-ROC of 0.79, outperformed the CredEye system, which achieved an AUC-ROC of 0.62. In turn, the latter outperformed random assignment, which achieved an AUC-ROC of 0.50. The second experiment, using the 30 claim dataset, The second experiment, using the 30-claim dataset, compared and contrasted PSL and MLN. From the results, a set of plots was generated, which encompassed all four models The same curves were plotted as previously, with results to indicate that FACT (PSL), with an AUC-ROC of 0.88, was superior to FACT (MLN), with an AUC-ROC of 0.71. In turn, the latter outperformed the CredEye system with an AUC-ROC of 0.62, and this was better than random assignment, which produced an AUC-ROC of 0.50. The p-values obtained in all three experimental case studies were (< 0.001), showing the strong statistical significance of the results.

Given these outcomes, which demonstrated that probabilistic models could be employed to competently establish the consistency of claims in relation to a knowledge base, it is now important for the reader to understand how the PSL model, in particular, distributes a truth value representing the level of consistency of a given claim. Therefore, it is necessary to examine the PSL model in detail to provide an explanation of this distribution of truth values, and also to present the decision-making process leading to these truth value assignments. These topics are covered in the next chapter.

CHAPTER 6

FACT DECISIONS EXPLANATIONS

6.1 Introduction

Several research studies have been conducted with the goal of using AI and machine-learning algorithms to detect false claims[100, 153, 154]. The ultimate goal of these studies was to innovate automated computational intelligence approaches that could support automated fact-checking systems for real-world application [18–21, 26].

Despite several previous attempts that have relied on the nature of the language, linguistics, and stylistic signals, results for this research area are limited [26, 27]. Automated fact-checking systems offer a finite ability to check the consistency of claims with an established knowledge base. The limitations of these studies, as described in the literature, include their susceptibility to bias. For this reason, there is an urgent need to improve the reliability, fairness, usability, and justifiability of such systems. In particular, the AI community must overcome the explainability barrier to enhance the efficiency of these systems.

Furthermore, researchers who study the social issues associated with technology addiction have argued that the world's increasing dependence on technology has its own pitfalls, relating to ethical concerns and improper use. However, most of these issues could be avoided if users were well informed about the reasoning behind computerised decisions[104]. Adadi and Berrada [100] emphasise that a lack of trust in AI and machine-learning systems stems from the absence of any explanation as to why a particular decision is concluded. Therefore, the necessity to understand the decisions made by intelligent software systems has become apparent[100, 104].

Consequently, explainable AI may be employed. Explainable AI is widely recognised as a critical requirement for the deployment and integration of AI models [155]. The rationale for a system's decision should be expressed in a way that is understandable to humans, known as interpretation or explanation. A fact-checking process, such as the one described in the present author's previous work [24, 25], could serve as the foundation for a system that was capable of explaining its own operations. This would help users gain insights into the AI models involved, as well as understanding their weaknesses. Moreover, studies such as [156] have proposed fact-checking techniques that deploy knowledge graphs to support explanations of AI decisions.

In [157], the authors developed an explainable factsheet framework to systematically compare and contrast different explanatory approaches, thereby identifying inconsistencies between the properties of these approaches and their theoretical qualities. Furthermore, in [158], a model-independent method was presented. This method aimed to capture the influence of inputs on outputs to explain why a particular decision was concluded.

However, Miller [104] disputes that an explanation is purely the researcher's own logic, in relation to the application, adding that the continuous incremental building of knowledge should be preserved to demonstrate real progress. Furthermore, an explained decision can be right or wrong. For example, it may be right, according to a number of logical steps, but wrong in a relevant real-world situation. Alternatively, the rationale reflected in the explanation may contain one or more syllogisms that are wrong in their premises (and therefore, their conclusions) or just in their conclusions. As a result, the true decision must adhere to a set of logical rules that are both intelligent and robust[159, 160].

Adadi and Berrada [100] state that several methods and techniques have been developed to induce explanation. Researchers in this study area have commonly used the term '**Interpretability**', which correlates with accuracy and explainability. The more complex the system, the more difficult it is to interpret. Therefore, a targeted system needs to be thoroughly understood, and its logical steps accurately illustrated before any explanation can be generated[102, 161].

This chapter demonstrates how the FACT approach was designed to be interpreted using PSL rules, thereby inferring results and a knowledge base to generate explanations. The main contribution of the proposed FACT approach is its ability to perform comprehensive and consistent checking, supported by knowledge base inference. It is not essential for these inferences to be based on current corpus knowledge; instead, they can be derived from the knowledge implied by probabilistic reasoning [24, 25]. Furthermore, the innovative approach presented in this thesis employs explainable AI by making machine-learning and AI models interpretative, since their decisions are well explained and justified. Consequently, it could be stated that the proposed FACT approach was interpreted by design.

Accordingly, this chapter is organised as follows: 1) a description of the proposed FACT decision explanation algorithm is introduced ; 2) the proposed algorithm is illustrated with examples from a royal family case study; 3) the explanations automatically generated by the innovated FACT are evaluated to authenticate their significance; 4) FACT explanations and the explanations generated by a standard system (CredEye) are evaluated and compared to benchmark FACT's performance in terms of being true, understandable, and satisfying, and 5) the chapter is summarised.

6.2 The FACT Decision Explanation Algorithm

An **'explanation'** is based on a set of ground atoms, such as premises, combined with PSL inference to support a conclusion with a specific degree of confidence. The explanation is presented as a sum of variously weighted factors, which can be logical rules.

Because of the graphical nature of the model, the aim here was not to give a full textual explanation of what PSL has achieved. Instead, the goal was to provide a graphical form that would support the PSL decision. Consequently, the set of ground atoms was considered, upon which the decision of a sufficient explanation was based. To be more specific, one 'winning' set of relevant beliefs was sought, in which more than one set of beliefs could produce a similar extant conclusion. The next section gives a detailed description of the explanation induction algorithm, which creates an explanation from the operations and results of PSL processes.

In this thesis, an explanation is regarded as the specification of a set of beliefs that are put together to provide a rational explanation of the phenomenon being induced. In order to add an explanation feature to an existing system, the best approach is to use the available resources, such as a log file. The log file of the FACT system contains all the inference rules that were automatically activated by PSL during the inference process. By looking at the log file, it is possible to find the nearest precedent, that is, the closest antecedent. This is undertaken one step at a time, until the initial set of facts are reached. Indeed, by closely monitoring the chain of beliefs to obtain the root of all the activated rules, a set of beliefs may be found that are strong enough to support the conclusion. A description of the decisions made by FACT (PSL) is given below through an (Algorithm 2), detailing the various steps of the proposed explanation process.

The (algorithm 2) elucidates the steps undertaken in inducing the explanation of a decision made by FACT using PSL. It shows how by setting a claim and a confidence threshold (defined by the end-user), the model can backtrack one step at a time until it reaches the last antecedent belonging to the initial belief, thereby completing the explanation process.

The algorithm could give us three different explanations:

- If an explanation of the claim is encountered, the claim is decided to be true. For example:
  Claim: Prince William is the father of Prince George.
  Explanation: Because Prince William is the parent of Prince George. Prince William is male.

- If no explanation for the claim is found, due to the absence of this relation in the knowledge base, the claim could be true or false. For example:
  Claim: Prince Louis is the father of Prince George.
  Explanation: No evidence supports this claim.

Algorithm 2: The FACT Decision Explanation Algorithm

Input : Claim C. a threshold for confidence (truth value) TS. The Ground rules $GroundRule$ used by PSL for inference.

Output : Textual explanation TE and (or) a tree for diagram explanation T.

Extract the relation r and the two entities e from the claim C to Ground atom $GroundAtom$.
if $GroundAtom$ *truth value* $> TS$ **then**
  Extract $GroundRule_i$ from the list of $GroundRule$ that have this $GroundAtom$ in its consequent.
  Repeat this to explain the antecedents if any Ground atoms in the antecedent not from the initial beliefs.
  Translate the antecedent as the text explanation TE and create the tree diagram explanation T.
  return TE *and* T
else
  for *each Predicate p in predication list* **do**
    if *p have the two entities e* **then**
      if $GroundAtom$ *truth value* $> TS$ **then**
        Extract $GroundRule_i$ from the list of $GroundRule$ that have this $GroundAtom$ in its consequent.
        Repeat this to explain the antecedents if any Ground atoms in the antecedent not from the initial beliefs.
        Translate the antecedent as the text explanation TE and create the tree diagram explanation T.
        return TE *and* T
      end
    end
  end
  return *No evidence supporting the claim*
end

- If no explanation for this relation is achieved, but there is an alternative relation supporting this claim, the claim will be false. For example:
  Claim: Prince William is the brother of Prince George.
  Explanation: No evidence supports this claim, but it could be proven that Prince William is the father of Prince George. Prince William is the parent of Prince George. Prince William is male.

To explain the algorithm further, the following section shows examples that apply to the royal family case study.

Table 6.1: Various steps corresponding to implementation of the algorithm, and related results from the text and tree diagram

Steps	Explanations of each step
Input	Claim C, Princess Eugenie is the sister of Princess Beatrice. Truth value TS =0.2.
Grand Atom	[$SISTER\_OF$(Princess Eugenie, Princess Beatrice)]
Check Grand Atom truth value	$SISTER\_OF$(Princess Eugenie, Princess Beatrice) truth value = 0.5463 which is superior than TS = 0.2.
Ground Rule$_i$	$SIBLINGS\_OF$(Princess Eugenie, Princess Beatrice) $\wedge$ $FEMALE$(Princess Eugenie) $\Rightarrow$ $SISTER\_OF$(Princess Eugenie, Princess Beatrice)
Antecedent of the rule	$SIBLINGS\_OF$(Princess Eugenie, Princess Beatrice) and $FEMALE$(Princess Eugenie).
Repeat the process	Because 'sibling' is not one of the initial beliefs, the process is repeated. An explanation is required for grand atom [$SIBLINGS\_OF$(Princess Eugenie, Princess Beatrice)]
Check the grand atom truth value	$SIBLINGS\_OF$(Princess Eugenie, Princess Beatrice) truth value = 0.6009, which is superior to TS = 0.2.
Ground Rule$_i$	$PARENT\_OF$(Prince Andrew, Princess Eugenie) $\wedge$ $PARENT\_OF$(Prince Andrew, Princess Beatrice) $\Rightarrow$ $SIBLINGS\_OF$(Princess Eugenie, Princess Beatrice)
Output	Text explanation: Prince Andrew is the parent of Princess Eugenie and the parent of Princess Beatrice. And Princess Eugenie is female. Tree diagram explanation is shown in Figure 6.1.

6.2.1 FACT Decision Explanation Algorithm by Examples

In Table 6.1, the steps of the algorithm are illustrated by a given example. In this research, the text-based method of explanation, which is the most common explanatory approach, encompasses its own specificity. The second technique applied, namely, a tree-based explanation that complements the text-based approach, is novel. Accordingly, the text and tree diagram (Figure 6.1) explanations, as generated by FACT, are unique in that they utilise PSL log inference. This backtracks the final decision with the aim of locating its antecedents. The process is then repeated to explain these antecedents. Indeed, from the presented example, a sentence could be generated in natural language and a small tree, allowing an explanation of the **Sister** relation that was not defined to be monitored by the proposed pipeline. For example, Princess Beatrice and Princess Eugenie are sisters.

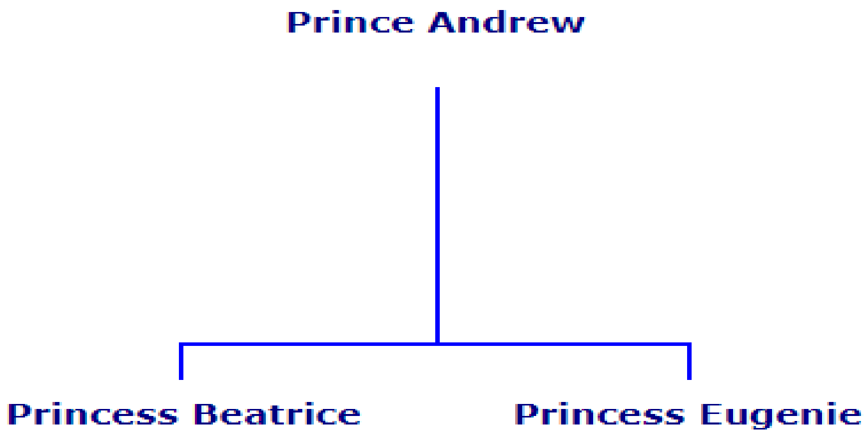


Figure 6.1: Tree diagram explanation.

6.3 Evaluating FACT Explanations

6.3.1 Introduction

Different methods of measuring the value of explanations have been utilised by researchers in the field of machine learning, in order to validate their work. In this regard, Doshi-Velez and Kim [162] raised the concern that the interpretability of these methods can be questionable. To refute this concern, different types of interpretability evaluations are commonly agreed upon. Application-grounded evaluation, which is the most commonly adopted approach, is based on a human expert who evaluates the explanations that are automatically generated by these systems. An updated version of this method is then introduced, which needs no human intervention. The method is widely used when the tested model has already been validated. The third type is human-grounded evaluation. This is conducted by a lay person, not an expert, and is appropriate to use when the objective is to test general notions or knowledge [163].

The final approach is one that is used to evaluate the explanations of claims in this study. Here, lay (non-expert) evaluators were relied on. The only selection criterion for these evaluators was that they be fluent in English. It is important to mention that all evaluations performed on an explanation were subjective. The reason for this is that such an evaluation is necessarily centred on the opinion of the human evaluator assessing the explanation, rather than a standard set of criteria. In fact, human evaluators who are not expert in the field use their own judgement and level of understanding to decide whether the explanations are true, understandable, and satisfying. In fact, there is no commonly agreed set of criteria for evaluating an explanation using this technique. However, a different set of evaluation experiments were conducted, utilising the judgement of these evaluators to mitigate the issue outlined above.

In this study, for the first two experiments, the explanation evaluation technique was based on the application of two criteria. However, the third experiment implemented three criteria.

The first criterion related to whether an explanation was actually true, i.e. whether it accurately reflected the way in which decisions were made. The second criterion focused on whether the explanations were understood, and the third criterion measured whether the explanations contained enough information to satisfy the respondent.

6.3.2 Design of the First and Second Experiments

The objective of this section was to ascertain the quality of the generated explanations. To be more specific, it needed to be established that the induced explanations really justified the decisions made. In the current implementation, this related to determining whether the explanations were true or false. In addition, it needed to be proven whether these explanations were understandable to the system end-user.

In these experiments, 61 different human evaluators checked whether the automated generated explanations were true and understandable. The participants comprised a mix of male and female evaluators, ranging in age from 20-40 years. Moreover, they lived in different countries, although they were all fluent English speakers. An anonymous Google Form was sent to each participant (screenshots of this may be found in the Appendix E). The claim and related explanations were listed in the Form, to which the evaluator responded either positively or negatively. Finally, after all the evaluators had submitted their completed Forms, the results were aggregated into an Excel spreadsheet.

Firstly, two criteria were applied: true and understandable. The meaning of each of these was illustrated in the Form as

- True: defines whether an explanation can positively justify the claim induced using the UK Royal Family Tree.
- Understandable: defines whether an explanation allows an individual to have a basic cognitive level (knowledge) when exposed to this explanation. .

On this basis, each evaluator had to define whether each explanation was true and/or understandable. Below is an example of an explanation and its associated claim.
Claim: Prince William is the husband of Princess Kate Middleton.
Explanation: Prince William is the spouse of Princess Kate Middleton and Prince William is a male.

In the above example, the respondent might define the explanation as

- True: Because Prince William is the spouse and Prince William is male (since all the families in the royal family currently constitute: male, female, and children).
- Understandable: Because the explanation informs us of William's marital relationship and that he is male.

The evaluators were divided into two groups. The first group contained 30 evaluators, while the second group contained 31. In addition, 20 different explanations were provided, associated with 20 different claims. These were divided into two sets, each made up of 10 explanations. The motive for this was the voluntary nature of filling out the evaluation Form. Consequently, it was deemed preferable in this study to give each respondent just 10 explanations to evaluate, rather than 20. This would ensure the evaluators' engagement in completing their evaluations in their entirety with authentic answers.

Each group of evaluators was then provided with a set of explanations to assess. Evaluators in the same group all received a similar set of explanations. To elucidate, the evaluators were divided into Groups A and B, whereas there were also two sets of explanations: A and B. Set A of the explanations was given to the evaluators in Group A, identified as the first experiment. Set B was then assigned to the evaluators in Group B, this being labelled as the second experiment. The aim of implementing this technique was to mitigate any bias in the results. Reliance was placed on the low probability of a different group of respondents giving the same answer to false data. However, they would be likely to give the same response to an authentic piece of information. In this manner, the results would tend to be authentic.

6.3.3 Results and Discussion of the First and Second Experiments

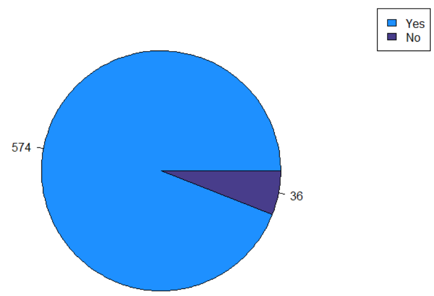
Results :

Based on the description of the evaluation process implemented, the results that were generated were a series of 'Yes' and 'No' answers for 'true' and 'understandable' evaluation points. These were converted into 1 for the 'Yes' option and 0 for 'No'.

Considering the **True** criterion, in the first experiment, the results were 277 'Yes' and 23 'No', making a total of 300 responses. Meanwhile, in the second experiment, the responses were 297 'Yes' and 13 'No', making a total of 310 responses. Overall, the responses were 574 'Yes' and 36 'No' out of a total of 610 responses. Thus, it may be confirmed that **94.10%** of the survey respondents agreed that FACT explanations truly related to the decision made for a claim.

Table 6.2: FACT explanations evaluated as 'True' in the First and Second Experiments

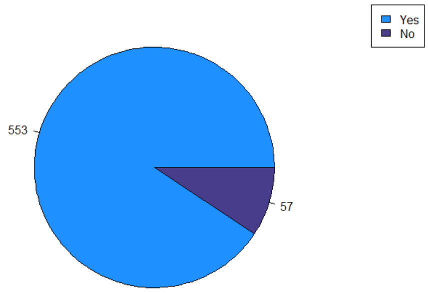
Experiment	#Yes	#No	%Yes
First Experiment	277	23	92.33
Second Experiment	297	13	95.81
Both	574	36	94.10



Considering the **'understandable'** criterion, in the first experiment, the results were 284 'Yes' and 16 'No', this being a total of 300 total responses. Meanwhile, in the second experiment, the responses were 269 'Yes' and 41 'No', making a total of 310 responses. Overall, the responses were 553 'Yes' and 57 'No', out of a total of 610 responses. Thus, it may be confirmed that **90.66%** of the participants in the survey agreed that FACT explanations give a basic understanding of the rationale behind the decision made.

Table 6.3: FACT explanations evaluated as 'Understandable' in the First and Second Experiments

Experiment	#Yes	#No	%Yes
First Experiment	284	16	94.67
Second Experiment	269	41	86.77
Both	553	57	90.66



Discussion of the First Experiment : Regarding the first evaluation point, which marks the explanation as '**True**', all the responses confirmed (Yes) that the explanation of Claim 10 was true, while 96.67% of the human evaluators supported that the explanations of Claims 2, 4, 6, and 9 were true. In addition, 93.33% decided that Claims 3 and 8 were true. However, for Claims 5 and 1, the percentages dropped from 86.67% to 76.67%, respectively (see Appendix G, Table G.1). These results are displayed in the bar chart in Figure 6.2.

Conversely, for '**understandable**', the generated outcome shows that the explanation for Claim 10 scored highest: 93.33%. Meanwhile, 96.67% evaluated the explanations for Claims 3, 4, 5, and 8 as understandable. In the same manner, 93.33% of the human evaluators judged the explanations for Claims 1, 2, 6, and 9 as understandable. Finally, the percentage of evaluators who considered the explanation for claim 7 to be understandable dropped to 86.67% (see Appendix G, Table G.1) for details. Figure 6.2 describes the same results as a bar chart.

Discussion of the Second Experiment : For the second experiment and regarding the **'true'** criteria, all the human evaluators confirmed that the explanations for Claims 1, 2, 4, 5, 7, 8, and 9 were true, while 87.10% evaluated the explanations for Claims 3 and 10 as true. Finally, 83.87% of the evaluators ascertained that the explanation for Claim 6 was true (see Appendix G, Table G.1 and the bar chart in Figure 6.2 for more details on this point).

For the **'understandable'** condition, all of the responses confirmed that the explanations for Claims 1 and 2 were understandable. Moreover, specifically, 96.77%, 93.55%, and 90.32% of the human evaluators agreed that the explanations for Claims 5, 8, and 10 were understandable. Regarding the explanations for Claims 4 and 7, the percentage of evaluators who found these understandable was 80.65%, while 77.42% found the explanation for Claim 3 understandable. Finally, 74.19% of the evaluators indicated that the explanations for Claims 6 and 9 were understandable. Further details on this point are presented in the Appendix G, Table G.1) and the bar chart in Figure 6.2.

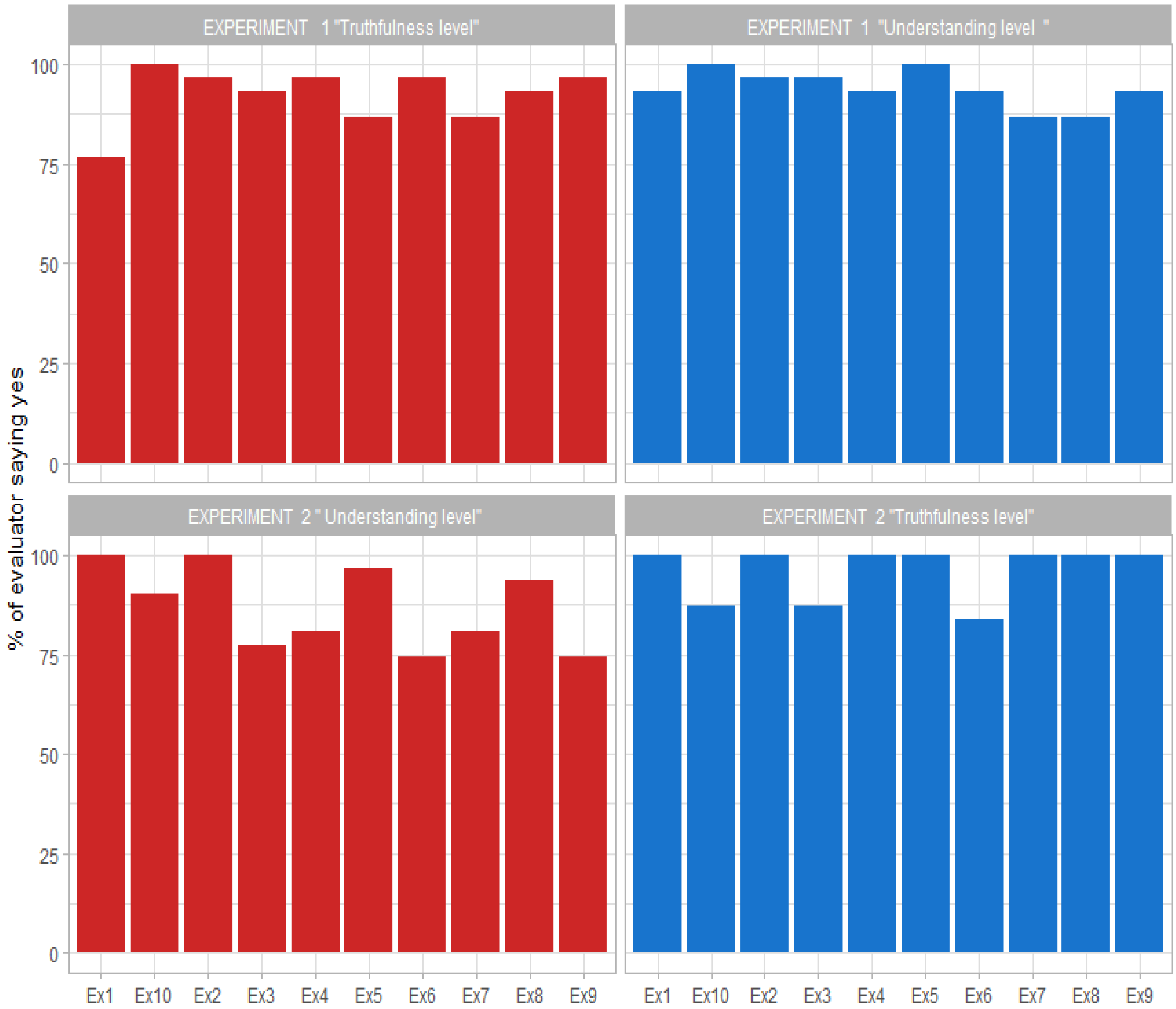


Figure 6.2: Explanations for 10 claims in the first and second experiments, applying the criteria of 'true' and 'understandable'. The x-axis displays the claims with their corresponding explanations, illustrated in different colours, while the y-axis shows the percentage of evaluators indicating 'Yes' (see Appendix F for more details on this point

Conclusion to the First and the Second Experiments : Although the evaluators were not expert in the subject area, the results obtained showed that they exercised reasonable judgement of the claim explanations for the UK royal family tree. However, better performance would be expected in terms of the accuracy of the evaluation if domain experts were employed.

For the two experiments, all the evaluators agreed on 8 out of 20 explanations being positively true, meaning that 40% of the explanations were undoubtedly true. Moreover, all the evaluators agreed that 3 out of 20 explanations were guaranteed to be understandable.

Overall, for the first criterion, namely, **'true'**, the results confirmed that (**94.10%**) of the evaluators considered the generated explanations to be true. For the second criterion, namely,

'understandable', the experiment revealed that (**90.66%**) of the evaluators agreed that the explanations were understandable.

These results were considered promising, and the method proposed in this study automatically generated a proper explanation without human interference. However, it was necessary to compare these results with a state-of-the-art approach in the field to ensure the authenticity of the innovated pipeline's performance. Consequently, in the next section, these results are compared with those of the CredEye model, using the same claims. The CredEye model was previously used to compare the level of consistency for different claims (see (chapter 5) for more detail on this point).

6.3.4 Design of the Third Experiment

The goal of this experiment was to evaluate the efficiency of FACT explanations, compared with other, widely used systems. Furthermore, it was necessary to explore how FACT explanations were relevant, meaningful, and informative for false data. To be more specific, it was necessary to ensure that the explanations of false claim decisions actually defined why such information or decisions were refuted.

For this experiment, four different claims were made: two true and two false. These were evaluated using the current innovative approach, namely FACT, as well as the CredEye system. A human evaluator was likewise employed to check the three criteria specified for this experiment: true, understandable, and satisfying. Hence, the participants were requested to conduct their evaluations based on these three criteria.

According to the setting of this experiment, there needed to be two explanations for each claim: one induced by the current innovative approach, FACT, and the other generated by CredEye. As elucidated in (Table 6.4), the claims were labelled as follows: ex1, ex2, ex3, and ex4. In each case, there were two explanations.

CredEye applies evidence directly from text, which may occasionally fail to provide a proper structure or rationale. To be more specific, CredEye builds its decisions on evidence that must be explicitly present in the inspected text. This makes the technique more or even totally dependent on the information gathered. Consequently, the evidence tends to be random, given that the items of collected data may be unrelated to each other, and more importantly, unrelated to the claim. A good example of this is Claim 1 in Table 6.4. In the case of a false claim, the model does not focus on trying to establish its veracity; instead, it uses a set of evidence that is usually unrelated to the claim, as illustrated in Claim 4 in the same Table. The immediate consequence is the risk that the system will justify its decision to confirm a claim that is clearly false.

Conversely, FACT does not need the facts involved in explaining a claim decision to be explicitly mentioned in the text. This means that FACT's explanation technique is less dependent on the text (articles collected from trusted sources) because the model is able to infer new facts. In addition, the technique applied in this study, which is more in-depth, produced an explanation that presented more structured evidence. It thereby represented, to a greater degree, the rationale underpinning each decision about a claim. This made it easy for the reader to follow and understand. Additionally, in the FACT model, everything is assumed to be false until proven otherwise.

The four claims and their associated explanations, generated using the two approaches, FACT and CredEye, may be found in Table 6.4. In this Table, the first column lists the four claims, while the second presents the corresponding FACT-generated explanations, and the third shows the explanations generated by CredEye.

The settings of this experiment are similar to those of the previous two experiments. The evaluators comprised a mix of male and female participants, comprising a total of 30 and aged between 20 and 40 years. An anonymous Google Form was shared with the participants, so that they could enter their own evaluation of the three criteria (see Appendix for screenshots E). Finally, the total responses were aggregated into a single spreadsheet.

However, this experiment employed a Likert scale, rather than simple Yes/No responses. It therefore included five levels of agreement or disagreement: Totally disagree, Disagree, Neutral, Agree, and Totally agree. This change of scale appeared to be capable of comparing the results obtained using the CredEye system, wherein the same Likert scale was used. Thus, this broad scale was essential for precise and concise comparison between the two approaches.

In this experiment, the same definitions of 'true' and 'understandable' were shared in the evaluation Form. However, a definition of 'satisfying' was appended. This was illustrated as the extent to which an individual would consider the given explanation to be informative.

Table 6.4: Comparing the FACT approach to the CredEye system in terms of the ability to generate claim explanations

Claim	FACT Explanation	CredEye Evidence
ex1- Princess Eugenie is the Cousin of Prince William	Prince Andrew is the Parent of Princess Eugenie. Prince Charles is the Parent of Prince William. Elizabeth II is the Parent of Prince Andrew. Elizabeth II is the Parent of Prince Charles.	Eugenie's cousin Prince Harry and Meghan Markle privately congratulated the couple, PEOPLE understands. Princess Eugenie Welcomes Baby Boy with Jack Brooksbank.
ex2- Princess Eugenie is the sister of Princess Beatrice.	Prince Andrew is the Parent of Princess Eugenie and Princess Beatrice. And Princess Eugenie is Female.	Princess Eugenie · Sister Princess Beatrice People also search for Princess Eugenie Sister Sophie, Countess of Wessex Catherine, Duchess of Cambridge Princess Margaret Princess Diana Data from: Wikipedia Feedback.
ex3-Prince Charles is the child of Prince William.	No evidence or explanation supporting this claim. But we have evidence for Prince Charles is the father of Prince William.	Only Prince William, 35, and Prince Harry, 33, are recognised as Charles' children, and heirs to his 100 million fortune and 1.3 billion Duchy of Cornwall . But, GLOBE claims to have uncovered a daughter and three sons who Charles has fathered over the years. Only Prince William and Prince Harry are recognised as Charles' children 'Prince Charles' four love children revealed'
ex4- Prince George is the uncle of Prince Edward.	No evidence or explanation support this claim.	The Duke of Kent is rumoured to have been addicted to drugs, especially morphine and cocaine, a rumour that reputedly originated with his friendship with Kiki Preston, whom he first met in the mid-1920s ...

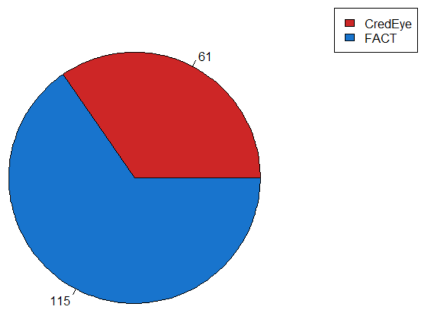
6.3.5 Results and Discussion of the Third Experiment

Result :

For the 'true' criterion only, the evaluators would simply answer 'Yes' or 'No'. The other two criteria, 'understandable' and 'satisfying' would be evaluated using the 5-point Likert scale. For the 'True' characteristic, the total responses for each approach amounted to 120, as each of the 30 evaluators assessed four claims. In the proposed FACT approach, for all claims, the results were 115 'Yes' and 5 'No'). Meanwhile, for the CredEye system, the results were 61 'Yes' and 59 'No'. This shows that FACT was positively evaluated by (**95.83%**) of the participants as giving true explanations, whilst only (**50.83%**) of the participants considered that CredEye gave true explanations. This proves that FACT was significantly (**88.52%**) more effective than CredEye for this criterion.

Table 6.6: FACT and CredEye explanations evaluated positively as 'True'

	#Yes	%Yes
FACT	115	95.83
CredEye	61	50.83
Difference	54	-
Enhancement	88.52	-

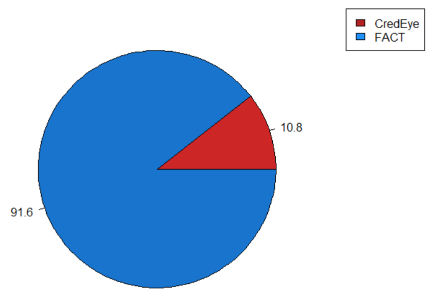


For the 'understandable' and 'satisfying' criteria, 'Totally agree' and 'Agree' were considered as positive evaluations, and 'Totally disagree' and 'Disagree' were negative evaluations. Due to its meaning, 'Neutral' was not counted.

Accordingly, for the **'understandable'** criterion, in relation to all the claims, FACT gained an average of (**91.6%**) positive evaluations, compared to (**10.8%**) for CredEye. These results show that FACT was an approach enhanced by (**748%**), compared to CredEye, in terms of inducing understandable explanations. Thus, FACT demonstrated a significant difference.

Table 6.7: FACT and CredEye explanations evaluated positively as 'Understandable'

Approach	%
FACT	91.60
CredEye	10.80
Difference	80.80
Enhancement	748

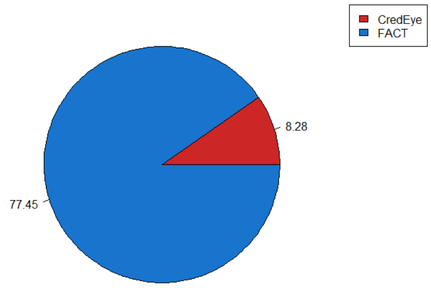


For the **Satisfying** criterion, for the 4 claims, FACT is agreed to be informative enough by

(**77.45%**) of the responses. On the other hand, CredEye is evaluated on the same scale and is positive by (**8.28%**). Based on that, FACT oversees CredEye in terms of being satisfied by (**836%**), which is a drastic change.

Table 6.8: FACT and CredEye explanations evaluated positively as 'Satisfying'

Approach	%
FACT	77.45
CredEye	8.28
Difference	69.18
enhancement	836



For the second goal, however, which was to evaluate FACT's performance on **false claims**, these being Claims 3 and 4 in the experiment; FACT was positively evaluated as giving a true, understandable, and satisfying explanation by **95%, 91.6%, and 61.65%** of the participants, respectively. Meanwhile, CredEye's performance was evaluated by **46.67%, 14.95%, and 11.60%** of the participants for the same metrics, respectively. FACT was still evaluated as performing considerably better than CredEye by **104%, 513%, and 431%** of the participants for the criteria: true, understandable and satisfying.

Table 6.9: FACT and CredEye explanations evaluated positively as 'True' for false claims only

Approach	#Yes	%Yes
FACT	57	95.00
CredEye	28	46.67
Difference	29	-
Enhancement	103.57	-

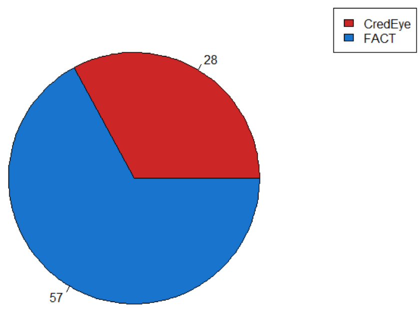


Table 6.10: FACT and CredEye explanations evaluated positively as ‘Understandable’ for false claims only

Approach	%
FACT	91.6
CredEye	14.95
Difference	76.65
enhancement	513

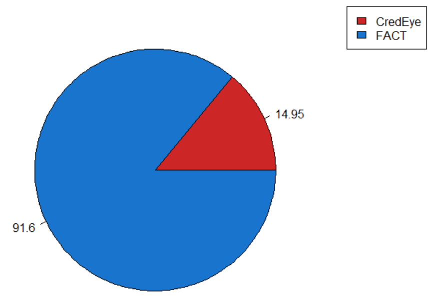
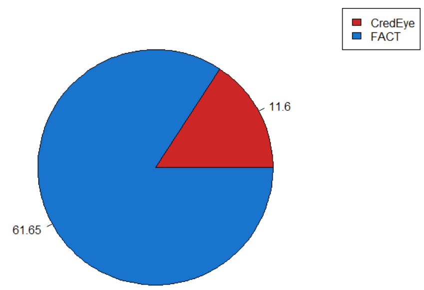


Table 6.11: FACT and CredEye explanations evaluated positively as ‘Satisfying’ for false claims only

Approach	%
FACT	61.65
CredEye	11.6
Difference	50.05
enhancement	431



To assess whether FACT’s performance began to drop when all of the claims were false, FACT was compared to itself, whereupon it evaluated a set of all true claims and another set of all false claims. The results showed that FACT performed almost equally well for true and false claims, in terms of the claims being true and understandable. However, its performance was decreased by (**34%**) for the ‘satisfying’ criterion.

Table 6.12: FACT explanations evaluated positively as ‘True’ for both true and false claims

Claim type	#Yes
FACT(True Claims Only)	58
FACT(False Claims Only)	57
Difference	1
Enhancement	1.75

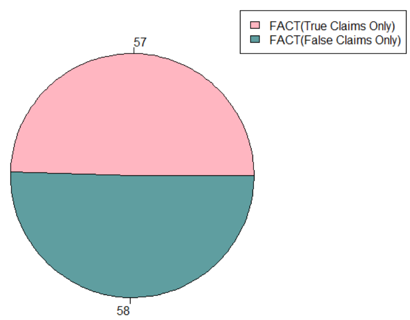


Table 6.13: FACT explanations evaluated positively as ‘Understandable’ for true and false claims

Claim type	%
True Claims Only	91.6
False Claims Only	91.6

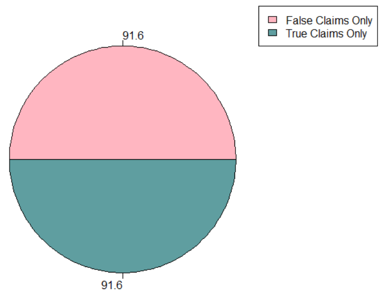
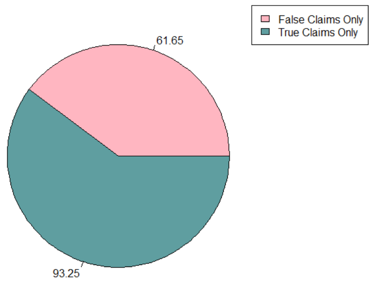


Table 6.14: FACT explanations evaluated positively as ‘Satisfying’ for true and false claims

Claim type	%
True Claims Only	93.25
False Claims Only	61.65
Difference	31.6
enhancement	34



Discussion of the First Criterion : For Claim 1, the results highlighted that the FACT approach was more efficient than the CredEye system. For example, FACT was defined as true by 96.6% of the respondents, while CredEye was defined as true by only 16.6%.

For Claims 2 and 3, the evaluations were almost identical, with only a marginal difference. Most of the evaluations detected that the explanations were true for both FACT and CredEye. For Claim 2, 96.67% of the respondents indicated FACT explanations as true, compared to the 93.33% positive evaluation provided for CredEye. Meanwhile, for Claim 3, FACT’s explanation was evaluated as true by 96.67% of the participants, compared to the 90% positive evaluation of CredEye’s explanations.

Similarly, 93.3% of the respondents ascertained that the explanation generated by FACT was true, while only 3.33% of the same respondents found the explanations given by CredEye to be true for Claim 4.

Based on these results, FACT proved to be a better solution than CredEye with regard to inducing a true explanation of claims. More details of this are illustrated in Appendix G, Table G.3).

Discussion of the Second Criterion : With regard to the **‘understandable’** condition, the results generated for the first claim showed that only 36.6-53.3% of the respondents agreed that the explanation given by FACT was understandable, whilst 3.3% of the participants indicated that the explanations provided were either ‘Neutral’, or else they ‘Totally disagreed’ with them. This may be compared to 0-13.3% and 50-33.3%, respectively, for the same metrics using CredEye.(Table G.4).

For the second claim, the results were similar, revealing that 33.3-60% of the respondents agreed that the explanation given by FACT was understandable, while 26.6-33.3% negated that. For CredEye, the averages for these metrics were 0-30% and 33.3-26.6%, respectively.

For the third claim, the averages were found to be 3.3-53.3% agreement and 0-3.3 disagreement that FACT explanations were understandable. This was in contrast to CredEye, which achieved an average of 0-26.6% positive and 23.3-33.33 negative responses. Thus, FACT explanations were evaluated as understandable.

For the fourth claim, the FACT averages were 0-46.6% positive evaluations and 0-10 refuted as being understandable. The corresponding averages for CredEye were 3.3-10% and 6.6-66.6%, respectively.

These results make it clear that FACT drastically exceeded CredEye in terms of its understandability, by a margin of **80.8%**. For more detail on this point, see (Appendix G, Table G.4).

Discussion of the Third Criterion : Considering the results for the **'satisfying'** criterion, the majority of the respondents 'Totally agreed' or 'Agreed' that the explanation generated by FACT was satisfying, with 46.6% defined for 'Totally agree' and 'Agree' in relation to Claim 1. For Claim 2, the agreement responses ranged from 40-53.3%, respectively. Meanwhile, for Claim 3, the agreement responses ranged from 33.3-50%, respectively, and for Claim 4, there was 20% total agreement and agreement by the evaluators.

However, the majority of responses to explanations generated by CredEye consisted of 'Disagree' or 'Totally Disagree'. To be more specific, these negative responses ranged from 40-53.33%, 30-43.3%, 30-40%, and 10-76.6% for Claims 1, 2, 3 and 4, respectively.

Based on these results, it is crystal clear that FACT outperformed CredEye in terms of the automatic generation of satisfying explanations to justify claims, by a margin of **69.18%**. More details on this point may be found in (Appendix G, Table G.5).

Conclusion : Based on the results, followed by detailed discussion, FACT was found to dramatically outperform CredEye by **88.52%**, **784%**, and **836%**, respectively, in terms of the criteria: true, understandable, and satisfying.

Furthermore, for false claims, these being Claims 3 and 4 in the study survey, FACT performed almost equally well for the first two criteria (true, understandable). However, its performance decreased by (**34%**) for the criterion, 'satisfying'. Nevertheless, FACT still exceeded CredEye across all metrics.

6.4 Summary

This chapter began by highlighting the gap in the extant literature, with regard to fact-checking, namely, a lack of interpretability. In other words, the rationale behind the decisions made needed to be presented. In this study, PSL was employed, combined with a knowledge base to check the consistency of claims (chapter 5), also utilising the results to generate explanations of decisions. Consequently, the contribution made by this study in this area may be defined as follows:

1. FACT is interpreted by design as it generates explanations related to judging a claim as either true or false.

2. FACT generates two forms of explanation, namely text and tree diagrams.

3. FACT text explanation is unique, as it relies on the correlation of PSL and an incremental knowledge base, rather than a linguistic approach. This makes FACT explanations reliable, justifiable, and non-biased.

4. The explanations produced by FACT were evaluated and it was confirmed that their meaning related to decisions made by **94%** of the evaluators, and were understandable for almost **91%** of the participants.

5. The performance of FACT explanations was superior to the performance of CredEye explanations, based on an objective evaluation by almost **89%, 748%, and 836%** of the participants, in terms of true, understandable, and satisfying conditions, respectively. The remarkable enhancement achieved by FACT in terms of its explanations being understandable and satisfying was due to its method of extracting those explanations, which was not based solely on linguistic inference.

6. FACT is unique among other research in the field, as it can provide related, comprehended, unbiased, and detailed explanations of both authentic and fake information. In contrast, where other approaches are capable of providing such explanations, it is solely for true claims. FACT was found to be successful in justifying why a claim was considered false. It performed equally well with true and false claims, in terms of the 'true' and 'understandable' metrics. However, its performance was decreased by **34%** for the 'satisfying' metric. In generic terms, FACT can give proper justification for positive and negative decisions.

Chapter 7

Conclusion

7.1 Summary and Conclusion

In this study, the goal was to investigate the efficiency of employing probabilistic reasoning in fact-checking systems. Thus, the FACT innovation was developed, this being a fully automated fact consistency-checking system, which does not require any human support. Instead, FACT utilises natural language-processing techniques to extract information from trusted sources, in order to dynamically build a knowledge base. Probabilistic reasoning is then applied to this knowledge base, so that FACT can return a decision over whether a questioned claim is true or false. One virtue of FACT is that it can induce information that is not explicitly mentioned in the source text. Furthermore, the performance of FACT is progressively enhanced because it is supported by a continuous learning technique, whereby it acquires new sources of information about any claims that are being investigated.

In addition, FACT can produce an automatic explanation of the resulting justification, as it is interpreted by design, due to its reliance on a dynamic incremental knowledge base, supported by probabilistic reasoning. It can determine why a claim or item of information is true or false, and represent its explanation as text and a tree diagram. To benchmark FACT, it was compared with CredEye in this study, since CredEye is a well-known and widely used automated fact-checking system. The above-mentioned comparison not only included decisions on whether claims were true or false, but also evaluations of the automatically generated explanations. However, to ensure that the results were reliable and fair in this study, a survey was conducted to gather human opinions about the decisions and explanations generated by both systems: FACT and CredEye. Based on these survey results, FACT appeared to perform significantly better than CredEye in terms of being true, understandable, and satisfying. To be more specific, FACT provided decisions and supported those decisions with explanations. FACT's explanations were also comprehensible to the human participants and contained sufficient detail to be informative. Moreover, FACT performed equally well for true and false claims, in terms of being 'true' and 'understandable'.

Throughout the research, the political relations and UK royal family and its internal relations was used as a cases studies. Thus, it illustrated how each phase of the FACT approach was implemented. To conclude this case study, FACT produced decisions on a number of claims relating to the UK royal family. It also induced explanations, presented as text or a tree diagram for each claim. Furthermore, to support the implementation of probabilistic reasoning, two more case studies were conducted. On this basis, it was found that PSL, used as a probabilistic logic model, performed better than MLN in a fact-checking system.

7.2 Contribution to knowledge

This study makes a number of significant contributions to knowledge, in the field of automated fact-checking systems. This was achieved by attempting to answer the primary research question: **'How can probabilistic reasoning be used to check for the internal consistency of a set of claims?'** , in addressing this question, an innovative approach was adopted, and FACT experiments were conducted using case studies and probabilistic reasoning (specifically, PSL) to test systems designed to check the consistency of claims. FACT was found to perform dramatically better than existing automated fact-checking systems such as CredEye, when employed in or associated with building a knowledge base that is engaged in continuous learning and the resolution of named entity problems.

Other contributions of this study are associated with the sub-questions derived from the main research question. Firstly, the sub-question, **'How can the construction and dynamic expansion of a knowledge base enhance checking the validity of claims?'** was addressed by the following means.

- Collecting articles related to a specific claim by crawling and scraping the Web creates a repository that is a starting point for fact-checking systems.
- A pipeline of information extraction techniques is applied to this repository, inducing named entities and the relationships between them. These can be saved in a knowledge base. Repeating the previous steps enriches this knowledge base on an ongoing basis.
- Finally, this knowledge base is used in conjunction with PSL to define the authenticity of a claim. Through experience, this approach proved to be more efficient than other automated systems like CredEye.

For the second sub-question, **'What is gained by employing probabilistic reasoning in such systems?'**, the answer provided the following contributions.

- First, probabilistic reasoning can be used to infer relations that are not explicitly defined in the text using logic and implementing an information extraction pipeline. In this study, only two relations were investigated, parents and spouses. However, the results implicitly revealed further relations, such as siblings, due to the inference process.
- Second, probabilistic reasoning can automatically generate explanations for decisions. This is the reason why FACT is interpreted by design. Based on the survey results, the human participants evaluated FACT's explanations, with these explanations being evaluated by 94% and 91% of the participants, respectively, as 'true' and 'understandable'. These results exceeded those achieved by CredEye, where **89%, 748%, and 836%**, respectively, agreed that the explanations were true, understandable, and satisfying.
- Third, probabilistic reasoning can automatically generate explanations of decisions as text and as tree diagrams.
- Fourth, FACT, given its reliance on probabilistic reasoning, was found to perform equally well for true and false claims in terms of the 'true' and 'understandable' measures. Consequently, it may be concluded that probabilistic reasoning can help provide proper justification for positive and negative decisions. For negative decisions, it demonstrates that there is no evidence to support a claim, giving an alternative relation to support the claim, if there is one available.
- Fifth, FACT explanations are reliable, justifiable, and non-biased, as they are generated through the probabilistic reasoning of relations in a knowledge base.

In responding to the third sub-question, **'What measures should be used to quantify the efficiency of the proposed solution?**', precision and recall were identified as appropriate measures for evaluating the proposed approach. Precision is defined as the percentage for the total number of correctly extracted relations in respect of the total number of relations identified by the system. Meanwhile, recall is defined as the percentage of the total number of correctly identified relations over the total number of relations present in the text. FACT achieved almost 85% for both recall and precision, these being considered as good ratios. Other contributions include:

- Probabilistic soft logic outperforming an MLN in checking the consistency of facts.
- An effective method of measuring the worthiness of the explanations generated by FACT, employing human evaluators. However, human evaluation should be designed in a way that will minimise bias. In this research, the survey set-up was innovative, as it was designed to ensure fairness. This was implemented by employing a number of human evaluators, all from different backgrounds. These evaluators were given the same set of claims, with automatically generated decisions and explanations of those decisions. If these human

evaluators agreed on the worthiness of a claim, it would indicate that their decisions were free of bias. Moreover, if they gave positive evaluations, it would prove that the FACT system was an effective tool for checking the consistency of facts.

The contributions set out above represent some highly contributions to the research area. However, a number of research limitations were also identified in this study. These, and any future work that might grow out of this study, are discussed below.

7.3 Future Work and Research Limitations

- While some important contributions were made by this research, more case studies in different fields of knowledge should be conducted to improve the generalisability of the results. However, for each new domain investigated, a different information extraction technique and JAPE grammar are required. This process would be a major learning curve in the path towards understanding the domain and creating an appropriate JAPE grammar in each case. The same problem arises with regard to the logical rules, which would need to be updated for each domain. Nevertheless, in future work, a system could be developed to automatically generate a specific JAPE grammar for the desired domain. In addition, FACT could be implemented using different logical rules for different domains by utilising the available online tools. Furthermore, two or more knowledge bases could be developed, relating to the same topic but covering a different aspect of it to perform more accurate fact-checking. For example, in the case of the UK royal family, kinship and geographical rules could be applied to create two knowledge bases. These knowledge bases could then be adopted to study other topics, such as immigration and its effects on the dispersal of families.

- No existing datasets were found for use in this research. Hence, a dataset was created specially for the study. However, this dataset was very limited in size. For future work, larger datasets could be defined, for example, by extracting them from social media, such as Facebook and Twitter.

- Nevertheless, large datasets require huge computational resources. Thus, it would be necessary to create a cloud-based version of FACT, utilising low-cost cloud services to support the research.

- For the evaluation of explanations, only a limited number of explanations were produced, with a relatively small number of evaluators recruited. This was due to the constraints imposed in response to the current COVID-19 pandemic and the students was unwillingness to participate in this kind of survey. Hence, for future work, it is intended to expand the set of explanations and include more evaluators. In addition, the MLN version of FACT would be evaluated, just as the PSL version was evaluated in this study.

- It is evident from this Conclusion and statement of the research limitations that this growing and in-demand field of research requires a combination of approaches to enhance the performance of automated fact-checking systems. The advantage of FACT is that it can make inferences from existing knowledge. However, in the proposed approach, greater generalisability of the results could be achieved by deploying larger datasets across a wider spectrum. This would point to the development of cloud-based FACT to maximise computational resources. Greater human participation to evaluate automated performance on this scale would consequently be necessary, operating under fewer constraints. Moreover, despite some knowledge bases being transferable to other domains, specific information extraction techniques and JAPE grammar necessary for each domain, whereupon these could be automatically generated by an innovative system. Finally, other versions of FACT, such as deploying an MLN, could be developed.

APPENDIX A

APPENDIX A: ABBREVIATION

FACT	Fact Automated Consistency Testing.
CredEye	A credibility lens for analyzing and explaining misinformation.
GATE	General Architecture for Text Engineering.
JAPE	a Java Annotation Patterns Engine.
ANNI	A Nearly-New IE system.
NELL	Never Ending Language Learner.
PSL	Probabilistic SOft Logic.
MAP	Markov Random Field.
HL-MRF	Hinge-Loss Markov Random Fields.
NLP	Natural Language Processing.
KB	Knowledge Base.
POS	Part Of Speech.
IE	Information Extraction.
KB	Knowledge Base.
ADMM	Alternating Direction Method of Multipliers.
LHS	Lift Hand side in the grammar rule.
RHS	Right Hand side in the grammar rule.
ROC	Receiver operating characteristic.
AUC	Area Under the ROC Curve
TP	True Positive.
FP	False Negative.
FPR	False Positive Rate.
TPR	True Positive Rate.

APPENDIX

B

APPENDIX B: EXPERIMENT RESULT

The table bellow shows the number of articles processed in each iteration and the number of relations extracted for both parent relation and spouse relation:

Search keyword	#articles	#CRS	#SR	#CNPR	#PR	#TR
"Prince Edward" "Peter Phillips"	117	16	16	1	1	17
"Princess Anne" "Zara Phillips"	117	31	15	6	5	37
"Prince Philip" "Arthur Chatto"	105	35	4	11	5	46
"Mary Elphinstone" "Elizabeth II"	113	41	6	12	1	53
"Prince Philip" "Princess Margaret"	128	59	18	12	0	71
"Prince William" "Elizabeth II"	130	79	20	13	1	92
"Lady Sarah Chatto" "Princess Anne"	118	103	24	23	10	126
"Princess Beatrice" "Prince Charles"	125	111	8	25	2	136
"Princess Beatrice" "Elizabeth II"	141	127	16	25	0	152
"Kate Middleton" "Prince William"	84	129	2	25	0	154
"Sarah Fergie Ferguson" "Prince William"	102	135	6	27	2	162
"Princess Charlotte" "Prince Harry"	118	138	3	27	0	165
"Prince Charles" "Princess Eugenie"	110	144	6	27	0	171
"Princess Charlotte" "Prince George"	126	146	2	28	1	174
"Princess Anne" "Prince William"	116	150	4	28	0	178
"Prince Harry" "Peter Phillips"	99	155	5	31	3	186
"Peter Phillips" "Elizabeth II"	134	169	14	41	10	210
"Prince Edward" "Prince William"	102	179	10	45	4	224
"Princess Margaret" "Princess Anne"	126	190	11	45	0	235
"Princess Charlotte" "Prince Harry"	113	195	5	45	0	240
"Princess Beatrice" "Isla Elizabeth Phillips"	84	201	6	53	8	254
"Prince Harry" "Princess Anne"	103	205	4	54	1	259
"Elizabeth II" "Antony Armstrong-Jones"	167	231	26	54	0	285
"Prince William" "Princess Anne"	121	239	8	55	1	294
"Meghan Markle" "Prince William"	86	243	4	55	0	298
"Prince Philip" "Prince William"	127	249	6	56	1	305
"Prince William" "Prince Harry"	104	249	0	56	0	305

Search keyword	#articles	#CRS	#SR	#CNPR	#PR	#TR
“Prince George“ “Prince Edward“	131	259	10	56	0	315
“Meghan Markle“ “Lena Elizabeth Tindall“	116	263	4	59	3	322
“Princess Anne“ “Elizabeth II“	136	278	15	60	1	338
“Prince Philip“ “Prince Harry“	120	284	6	60	0	344
“Zara Phillips“ “Prince Charles“	86	290	6	61	1	351
“Elizabeth II“ “Prince William“	126	306	16	64	3	370
“Princess Beatrice“ “Prince Charles“	125	308	2	64	0	372
“Isla Elizabeth Phillips“ “Prince William“	111	314	6	68	4	382
“Prince William“ “Princess Anne“	119	320	6	69	1	389
“Peter Phillips“ “Kate Middleton“	110	323	3	71	2	394
“Diana Spencer“ “Prince Edward“	128	331	8	71	0	402
“Sarah Fergie Ferguson“ “Peter Phillips“	82	347	16	72	1	419
“Peter Phillips“ “Sarah Fergie Ferguson“	78	357	10	73	1	430
“Isla Elizabeth Phillips“ “Princess Beatrice“	87	363	6	79	6	442
“Prince Harry“ “Isla Elizabeth Phillips“	124	375	12	90	11	465
“Elizabeth II“ “Zara Phillips“	137	385	10	90	0	475
“Savannah Phillips“ “Princess Anne“	119	390	5	92	2	482
“Princess Margaret“ “Prince William“	121	398	8	96	4	494
“Autumn Phillips“ “Peter Phillips“	141	406	8	97	1	503
“Isla Elizabeth Phillips“ “Savannah Phillips“	125	413	7	99	2	512
“Princess Beatrice“ “Elizabeth II“	137	415	2	100	1	515
“Lady Sarah Chatto“ “David Armstrong-Jones“	97	425	10	107	7	532
“Princess Margaret“ “Daniel Chatto“	122	437	12	109	2	546
“Princess Anne“ “Mia Grace Tindall“	67	441	4	112	3	553
“Princess Anne“ “Isla Elizabeth Phillips“	103	445	4	113	1	558
“Zara Phillips“ “Peter Phillips“	149	447	2	113	0	560
“Peter Phillips“ “Prince William“	100	448	1	113	0	561
“Prince George“ “Princess Charlotte“	116	450	2	113	0	563
“Princess Anne“ “Elizabeth II“	129	456	6	113	0	569
“Prince Edward“ “Prince William“	106	460	4	113	0	573
“Prince Louis“ “Elizabeth II“	136	466	6	113	0	579
“Prince Charles“ “David Armstrong-Jones“	122	476	10	114	1	590
“Prince George“ “Princess Beatrice“	111	480	4	114	0	594
“Kate Middleton“ “Prince William“	84	480	0	114	0	594
“David Armstrong-Jones“ “Elizabeth II“	141	482	2	114	0	596
“Capt Mark Phillips“ “Prince William“	102	483	1	114	0	597
“David Armstrong-Jones“ “Arthur Chatto“	105	489	6	115	1	604
“Prince Harry“ “Princess Beatrice“	105	493	4	115	0	608
“Elizabeth II“ “Margaret Elphinstone“	92	503	10	116	1	619
“Mia Grace Tindall“ “Lena Elizabeth Tindall“	84	503	0	116	0	619
“David Armstrong-Jones“ “Arthur Chatto“	109	513	10	120	4	633
“Mia Grace Tindall“ “Princess Eugenie“	76	517	4	130	10	647
“Prince George“ “Prince William“	113	517	0	130	0	647
“Elizabeth II“ “Princess Anne“	137	525	8	131	1	656
“Arthur Chatto“ “Princess Beatrice“	134	533	8	135	4	668
“Lady Sarah Chatto“ “Mia Grace Tindall“	44	535	2	135	0	670
“Princess Beatrice“ “Prince William“	104	535	0	136	1	671
“Elizabeth Bowes-Lyon“ “Elizabeth II“	146	556	21	140	4	696
“Isla Elizabeth Phillips“ “Princess Beatrice“	84	558	2	141	1	699
“Isla Elizabeth Phillips“ “James“	122	567	9	144	3	711
“Princess Margaret“ “Princess Eugenie“	111	578	11	145	1	723
“Elizabeth II“ “Princess Margaret“	126	586	8	145	0	731

Search keyword	#articles	#CRS	#SR	#CNPR	#PR	#TR
“Prince William“ “Prince George“	112	586	0	145	0	731
“Prince George“ “Prince Edward“	132	597	11	149	4	746
“Elizabeth II“ “Prince Harry“	141	599	2	149	0	748
“Autumn Phillips“ “Princess Anne“	103	604	5	149	0	753
“Elizabeth II“ “Princess Eugenie“	127	608	4	149	0	757
“Elizabeth II“ “Prince Philip“	133	610	2	149	0	759
“Princess Eugenie“ “Prince Charles“	117	611	1	150	1	761
“James“ “Prince Harry“	145	613	2	151	1	764
“Savannah Phillips“ “Prince William“	139	613	0	152	1	765
“Peter Phillips“ “Princess Anne“	122	617	4	152	0	769
“Elizabeth II“ “Prince Charles“	130	623	6	152	0	775
“Antony Armstrong-Jones“ “Elizabeth II“	170	639	16	152	0	791
“Prince Charles“ “Prince Andrew“	108	639	0	153	1	792
“Prince Andrew“ “Prince Charles“	110	645	6	153	0	798
“Princess Anne“ “Kate Middleton“	96	647	2	153	0	800
“Prince William“ “Prince Louis“	109	647	0	153	0	800
“Prince Louis“ “Princess Eugenie“	105	660	13	153	0	813
“Prince William“ “Kate Middleton“	94	660	0	153	0	813
“Prince Charles“ “Prince William“	132	666	6	162	9	828
“Princess Anne“ “Prince Charles“	121	667	1	162	0	829
“Princess Eugenie“ “Princess Anne“	118	674	7	163	1	837
“Isla Elizabeth Phillips“ “Savannah Phillips“	132	675	1	163	0	838
“Prince Harry“ “Kate Middleton“	96	675	0	163	0	838
“Prince Charles“ “Prince William“	128	675	0	163	0	838
“Prince Harry“ “Princess Eugenie“	112	675	0	163	0	838
“Lady Sarah Chatto“ “Samuel Chatto“	129	675	0	168	5	843
“Arthur Chatto“ “Charles Armstrong-Jones“	85	677	2	171	3	848
“Kate Middleton“ “Prince Harry“	89	679	2	172	1	851
“Prince Louis“ “Meghan Markle“	104	681	2	172	0	853
“Princess Anne“ “Prince Andrew“	122	687	6	172	0	859
“Lady Sarah Chatto“ “Charles Armstrong-Jones“	99	691	4	173	1	864
“Elizabeth II“ “Sarah Fergie Ferguson“	109	691	0	174	1	865
“Arthur Chatto“ “Princess Margaret“	114	691	0	176	2	867
“Prince Louis“ “Prince Charles“	113	693	2	176	0	869
“Daniel Chatto“ “David Armstrong-Jones“	103	695	2	176	0	871
“Mary Elphinstone“ “Cecilia Bowes-Lyon“	75	714	19	178	2	892
“Prince Philip“ “Diana Spencer“	122	716	2	178	0	894
“Prince Louis“ “Prince William“	102	716	0	180	2	896
“Prince Charles“ “Peter Phillips“	97	720	4	180	0	900
“Lady Sarah Chatto“ “Charles Armstrong-Jones“	98	720	0	180	0	900
“Prince Charles“ “Princess Anne“	125	728	8	180	0	908
“Peter Phillips“ “Zara Phillips“	140	734	6	181	1	915
“Prince Louis“ “Prince Harry“	108	736	2	182	1	918
“Lady Margarita Armstrong-Jones“ “Charles Armstrong-Jones“	114	742	6	184	2	926
“Princess Beatrice“ “Princess Eugenie“	105	743	1	184	0	927
“Zara Phillips“ “Princess Beatrice“	101	753	10	185	1	938
“David Armstrong-Jones“ “Prince William“	140	753	0	186	1	939
“Princess Margaret“ “Prince Charles“	126	755	2	187	1	942
“Cecilia Bowes-Lyon“ “Elizabeth Bowes-Lyon“	78	767	12	190	3	957
“Prince Charles“ “Sarah Fergie Ferguson“	123	775	8	191	1	966
“Diana Spencer“ “Elizabeth II“	127	795	20	197	6	992
“Prince Charles“ “Zara Phillips“	89	795	0	197	0	992

Search keyword	#articles	#CRS	#SR	#CNPR	#PR	#TR
“Arthur Chatto“ “David Armstrong-Jones“	104	795	0	197	0	992
“Princess Beatrice“ “Prince Edward“	109	797	2	197	0	994
“Lady Sarah Chatto“ “Princess Anne“	129	805	8	198	1	1003
“Princess Eugenie“ “Princess Beatrice“	94	805	v 0	198	0	1003
“Antony Armstrong-Jones“ “Princess Margaret“	172	829	24	198	0	1027
“Meghan Markle“ “Princess Charlotte“	108	831	2	199	1	1030
“Prince William“ “Elizabeth II“	142	835	4	199	0	1034
“Lady Margarita Armstrong-Jones“ “Charles Armstrong-Jones“	117	841	6	202	3	1043
“Lady Louise Windsor“ “Elizabeth II“	104	841	0	202	0	1043
“Prince Louis“ “Isla Elizabeth Phillips“	98	846	5	208	6	1054
“Prince George“ “Lady Sarah Chatto“	116	869	23	210	2	1079
“Prince Louis“ “Prince Charles“	105	873	4	211	1	1084
“Prince Edward“ “Prince Charles“	126	883	10	211	0	1094
“Princess Eugenie“ “Elizabeth II“	121	885	2	212	1	1097
“Prince Louis“ “Lady Sarah Chatto“	96	891	6	212	0	1103
“Capt Mark Phillips“ “Elizabeth II“	111	891	0	212	0	1103
“David Armstrong-Jones“ “Serena Armstrong-Jones“	119	897	6	212	0	1109
“Prince George“ “Prince Louis“	99	899	2	212	0	1111
“Prince Louis“ “Prince Harry“	109	901	2	212	0	1113
“Prince Charles“ “Elizabeth II“	128	905	4	212	0	1117
“Savannah Phillips“ “Isla Elizabeth Phillips“	121	907	2	212	0	1119
“Prince Harry“ “Prince William“	120	909	2	213	1	1122
“Prince Edward“ “Elizabeth II“	118	911	2	213	0	1124
“Prince Andrew“ “Elizabeth II“	150	917	6	213	0	1130
“Princess Anne“ “Lady Sarah Chatto“	112	921	4	213	0	1134
“Lady Sarah Chatto“ “Princess Margaret“	119	925	4	213	0	1138
“Capt Mark Phillips“ “Prince Charles“	124	937	12	217	4	1154
“Prince Harry“ “Princess Anne“	105	941	4	219	2	1160
“Samuel Chatto“ “Lady Sarah Chatto“	122	941	0	219	0	1160
“Princess Anne“ “Savannah Phillips“	116	941	0	220	1	1161
“Lady Sarah Chatto“ “Lady Margarita Armstrong-Jones“	101	941	0	220	0	1161
“Princess Beatrice“ “Princess Eugenie“	105	943	2	222	2	1165
“Prince Harry“ “Elizabeth II“	149	945	2	222	0	1167
“Princess Charlotte“ “Meghan Markle“	125	949	4	222	0	1171
“Elizabeth II“ “Diana Spencer“	133	958	9	225	3	1183
“Arthur Chatto“ “Lady Sarah Chatto“	128	958	0	225	0	1183
“Prince Harry“ “Prince Charles“	122	958	0	225	0	1183
“Prince Harry“ “Prince William“	114	960	2	225	0	1185
“Lady Louise Windsor“ “James“	99	962	2	225	0	1187
“Princess Beatrice“ “Princess Eugenie“	99	962	0	225	0	1187
“Prince William“ “Elizabeth II“	131	962	0	226	1	1188
“Arthur Chatto“ “Prince Harry“	123	962	0	227	1	1189
“Diana Spencer“ “Prince Andrew“	127	966	4	228	1	1194
“Prince Edward“ “Diana Spencer“	125	978	12	228	0	1206
“Elizabeth II“ “Princess Margaret“	132	980	2	228	0	1208
“Prince Harry“ “Mia Grace Tindall“	82	984	4	236	8	1220
“Elizabeth Bowes-Lyon“ “Cecilia Bowes-Lyon“	86	999	6	236	0	1235
“Princess Beatrice“ “Peter Phillips“	106	1003	4	236	0	1239
“Lady Sarah Chatto“ “Samuel Chatto“	130	1005	2	236	0	1241
“Princess Margaret“ “Zara Phillips“	103	1009	4	244	8	1253
“Samuel Chatto“ “Lady Sarah Chatto“	123	1009	0	245	1	1254
“Prince William“ “Prince Philip“	124	1011	2	245	0	1256

Search keyword	#articles	#CRS	#SR	#CNPR	#PR	#TR
“Lady Sarah Chatto“ “Princess Margaret“	120	1015	4	245	0	1260
“Peter Phillips“ “Savannah Phillips“	129	1017	2	245	0	1262
“Serena Armstrong-Jones“ “Lady Sarah Chatto“	71	1023	6	245	0	1268
“Prince Charles“ “Princess Beatrice“	121	1025	2	2457	0	1270
“Peter Phillips“ “Princess Eugenie“	109	1025	0	245	0	1270
“Elizabeth II“ “Savannah Phillips“	120	1026	1	246	1	1272
“Princess Eugenie“ “Prince William“	113	1038	12	246	0	1284
“Mike Tindall“ “Peter Phillips“	117	1038	0	246	0	1284
“Autumn Phillips“ “Prince Harry“	119	1040	2	246	0	1286
“Prince William“ “Princess Beatrice“	99	1042	2	246	0	1288
“Prince Andrew“ “Princess Anne“	129	1042	0	246	0	1288
“Prince Harry“ “Zara Phillips“	108	1044	2	248	2	1292
“Peter Phillips“ “Prince Charles“	102	1046	2	248	0	1294

APPENDIX

C

APPENDIX C: GROUND TRUTH

The table bellow shows the ground truth dataset:

True statements
Princess Anne is the ancestor of Zara Phillips.
Prince Charles is the descendent of Elizabeth II.
Elizabeth II is the wife of Prince Philip.
Autumn Phillips is the wife of Peter Phillips.
Antony Armstrong-Jones is the husband of Princess Margaret.
Prince William is the husband of Kate Middleton.
Elizabeth Bowes-Lyon is the mother of Elizabeth II.
Lady Sarah Chatto is the mother of Samuel Chatto.
Prince Charles is the father of Prince William.
Prince William is the father of Prince George.
Princess Anne is the daughter of Elizabeth II.
Peter Phillips is the son of Princess Anne.
Lady Sarah Chatto is the child of Princess Margaret.
Prince Harry is the child of Prince Charles.
Prince Andrew is the child of Elizabeth II.
Prince Edward is the child of Elizabeth II.
Lady Margarita Armstrong-Jones is the siblings of Charles Armstrong-Jones.
Savannah Phillips is the siblings of Isla Elizabeth Phillips.
Princess Beatrice is the siblings of Princess Eugenie.
Elizabeth II is the siblings of Princess Margaret.

Zara Phillips is the siblings of Peter Phillips.
Prince George is the siblings of Prince Louis.
Prince Edward is the siblings of Prince Charles.
Prince William is the siblings of Prince Harry.
Princess Eugenie is the sister of Princess Beatrice.
Elizabeth II is the sister of Princess Margaret.
Lady Louise Windsor is the sister of James.
Princess Anne is the sister of Prince Charles.
Mia Grace Tindall is the sister of Lena Elizabeth Tindall.
Princess Anne is the sister of Prince Andrew.
Lady Margarita Armstrong-Jones is the sister of Charles Armstrong-Jones.
Prince Charles is the brother of Princess Anne.
Prince Andrew is the brother of Princess Anne.
Prince Charles is the brother of Prince Andrew.
Prince Andrew is the brother of Prince Charles.
Peter Phillips is the brother of Zara Phillips.
Prince George is the brother of Princess Charlotte.
Princess Eugenie is the niece of Princess Anne.
Zara Phillips is the niece of Prince Charles.
Lady Sarah Chatto is the niece of Elizabeth II.
Princess Eugenie is the niece of Prince Charles.
Princess Charlotte is the niece of Prince Harry.
Princess Charlotte is the niece-In-Law of Meghan Markle.
Prince William is the nephew of Princess Anne.
Peter Phillips is the nephew of Prince Charles.
David Armstrong-Jones is the nephew of Elizabeth II.
Prince Louis is the nephew-In-Law of Meghan Markle.
Princess Margaret is the aunt of Princess Anne.
Princess Anne is the aunt of Prince William.
Princess Margaret is the aunt of Prince Charles.
Mary Elphinstone is the aunt of Elizabeth II.
Lady Sarah Chatto is the aunt of Charles Armstrong-Jones.
Meghan Markle is the aunt-In-Law of Princess Charlotte.
Princess Anne is the aunt-In-Law of Kate Middleton.
Sarah Fergie Ferguson is the aunt-In-Law of Prince William.
Prince Charles is the uncle of Princess Eugenie.
Prince Edward is the uncle of Prince William.

Prince Edward is the uncle of Peter Phillips.
Prince Charles is the uncle of Peter Phillips.
David Armstrong-Jones is the uncle of Arthur Chatto.
Capt Mark Phillips is the uncle-In-Law of Prince William.
Princess Eugenie is the cousin of Prince William.
Peter Phillips is the cousin of Princess Eugenie.
Prince William is the cousin of Princess Beatrice.
Prince Charles is the cousin of David Armstrong-Jones.
Princess Anne is the cousin of Lady Sarah Chatto.
Elizabeth II is the cousin of Margaret Elphinstone.
Peter Phillips is the cousin-In-Law of Kate Middleton.
Autumn Phillips is the child-In-Law of Princess Anne.
Capt Mark Phillips is the son-In-Law of Elizabeth II.
Diana Spencer is the daughter-In-Law of Elizabeth II.
Elizabeth II is the parent-In-Law of Sarah Fergie Ferguson.
Elizabeth II is the mother-In-Law of Diana Spencer.
Prince Philip is the father-In-Law of Diana Spencer.
Elizabeth II is the sibling-In-Law of Antony Armstrong-Jones.
Diana Spencer is the sister-In-Law of Princess Anne.
Serena Armstrong-Jones is the sister-In-Law of Lady Sarah Chatto.
Kate Middleton is the sister-In-Law of Prince Harry.
Meghan Markle is the sister-In-Law of Prince William.
Diana Spencer is the sister-In-Law of Prince Andrew.
Diana Spencer is the sister-In-Law of Prince Edward.
Antony Armstrong-Jones is the brother-In-Law of Elizabeth II.
Daniel Chatto is the brother-In-Law David Armstrong-Jones.
Capt Mark Phillips is the brother-In-Law Prince Charles.
Prince Charles is the brother-In-Law Sarah Fergie Ferguson.
Prince Edward is the brother-In-Law Diana Spencer.
Prince Philip is the brother-In-Law Princess Margaret.
Mike Tindall is the brother-In-Law Peter Phillips.
Prince William is the grand child of Elizabeth II.
Prince Harry is the grand child of Elizabeth II.
Lady Louise Windsor is the grand child of Elizabeth II.
Princess Eugenie is the grand child of Elizabeth II.
Prince William is the grand son of Prince Philip.
Princess Beatrice is the grand daughter of Elizabeth II.

Elizabeth II is the grand parent of Princess Eugenie.
Princess Anne is the grand parent of Mia Grace Tindall.
Princess Anne is the grand parent of Savannah Phillips.
Prince Philip is the grand father of Prince Harry.
Elizabeth II is the grandmother of Prince William.
Fake statements
Zara Phillips is the ancestor of Princess Beatrice.
Elizabeth Bowes_Lyon is the ancestor of Cecilia Bowes_Lyon.
Elizabeth IIis the descendent of Prince Harry.
Prince Harry is the descendent of Prince William.
Autumn Phillips is the Wife of Prince Harry.
Peter Phillipsis the Wife of Sarah Fergie Ferguson.
Sarah Fergie Ferguson is the husband of Peter Phillips.
Prince Harry is the husband of Zara Phillips.
Prince Harry is the child of Isla Elizabeth Phillips.
Prince Louis is the child of Princess Eugenie.
Prince William is the child of Elizabeth II.
Prince Harry is the child of Princess Anne.
Prince Charles is the child of Prince William.
David Armstrong_Jones is the child of Prince William.
Lady Sarah Chatto is the child of David Armstrong_Jones.
Prince Louisis the Son of Elizabeth II.
Princess Beatrice is the son of Elizabeth II.
Mary Elphinstone is the son of Cecilia Bowes_Lyon.
Kate Middleton is the daughter of Prince William.
Prince Louis is the daughter of Prince William.
Samuel Chatto is the daughter of Lady Sarah Chatto.
Cecilia Bowes_Lyon is the father of Elizabeth Bowes_Lyon.
Princess Charlotte is the father of Prince George.
Prince Harry is the father of Kate Middleton.
Meghan Markle is the mother of Lena Elizabeth Tindall.
Prince William is the mother_Of Prince Louis.
David Armstrong_Jones is the mother of Serena Armstrong_Jones.
Prince George is the siblings of Princess Beatrice.
Princess Margaret is the siblings of Daniel Chatto.
Samuel Chatto is the siblings of Lady Sarah Chatto.
Arthur Chatto is the siblings of Charles Armstrong_Jones.

Isla Elizabeth Phillips is the sister of Princess Beatrice.
Elizabeth II is the sister of Prince Charles.
Prince Harry is the sister of Prince William.
Princess Beatrice is the brother of Princess Eugenie.
Prince George is the brother of Prince William.
Isla Elizabeth Phillips is the brother of Prince William.
Prince Louis is the aunt of Isla Elizabeth Phillips.
Lady Sarah Chatto is the aunt of Samuel Chatto.
Princess Beatrice is the aunt of Isla Elizabeth Phillips.
Kate Middleton is the aunt_In Law of Prince William.
Lady Sarah Chatto is the aunt_In Law Lady Margarita Armstrong_Jones.
Prince George is the uncle of Prince Edward.
Princess Beatrice is the uncle of Prince Edward.
Lady Sarah Chatto is the uncle of Charles Armstrong_Jones.
Prince George is the Uncle_In_Law of Prince Edward.
Prince Edward is the Uncle_In_Law of Prince William.
David Armstrong_Jones is the Uncle_In_Law of Arthur Chatto.
Isla Elizabeth Phillips is the niece of James.
Prince Louis is the niece of Prince Harry.
Arthur Chatto is the niece of David Armstrong_Jones.
Princess Charlotte is the niece_In_Law of Prince Harry.
Arthur Chatto is the nephew of Prince Harry.
Princess Beatrice is the nephew of Princess Eugenie.
Princess Beatrice is the nephew of Prince Charles.
Prince Louis is the nephew_In_Law of Prince Charles.
Princess Beatrice is the nephew_In_Law of Prince Charles.
Prince George is the nephew_In_Law of Lady Sarah Chatto.
Prince Louis is the cousin of Prince Charles.
Prince Louis is the cousin of Prince Harry.
Isla Elizabeth Phillips is the cousin of Savannah Phillips.
Prince Louis is the cousin of Lady Sarah Chatto.
Isla Elizabeth Phillips is the cousin_In_Law of Savannah Phillips.
Lady Sarah Chatto is the cousin_In_Law of Princess Anne.
Elizabeth II is the child_In_Law of Princess Anne.
Prince Harry is the son_In_Law of Princess Anne.
Princess Beatrice is the son_In_Law of Prince William.
Arthur Chatto is the son_In_Law of Lady Sarah Chatto.

Isla Elizabeth Phillips is the daughter_In_Law of Princess Beatrice.
Prince Charles is the daughter_In_Law of Princess Beatrice.
Lady Sarah Chatto is the daughter_In_Law of Princess Margaret.
James is the parent_In_Law of Prince Harry.
Princess Margaret is the mother_In_Law of Zara Phillips.
Prince Charles is the mother_In_Law of Zara Phillips.
Elizabeth II is the mother_In_Law of Zara Phillips.
Savannah Phillips is the father_In_Law of Prince William.
Peter Phillips is the father_In_Law of Prince William.
Arthur Chatto is the father_In_Law of Princess Margaret.
Prince Harry is the sibling_In_Law of Princess Beatrice.
Mia Grace Tindall is the sister_In_Law of Princess Eugenie.
Prince Harry is the sister_In_Law of Princess Eugenie.
Princess Margaret is the sister_In_Law of Princess Eugenie.
Elizabeth II is the brother_In_Law of Savannah Phillips.
Peter Phillips is the brother_In-Law of Savannah Phillips.
Prince Harry is the grand child of Peter Phillips.
Prince William is the grand son of Princess Anne.
Lady Sarah Chatto is the grand son of Princess Anne.
Savannah Phillips is the grand son of Princess Anne.
Peter Phillips is the grand daughter of Elizabeth II.
Princess Anne is the grand daughter of Elizabeth II.
Prince Harry is the grand parent of Mia Grace Tindall.
Arthur Chatto is the grand parent of Princess Beatrice.
Lady Sarah Chatto is the grand parent of Mia Grace Tindall.
Prince William is the grand father of Elizabeth II.
Princess Anne is the grand father Isla Elizabeth Phillips.
Princess Beatrice is the grand mother of Peter Phillips.
Prince Philip is the grand mother of Prince William.
Prince Philip is the grand mother of Arthur Chatto.
Princess Margaret is the grand mother of Prince William.

APPENDIX D

APPENDIX D: LOGICAL RULES

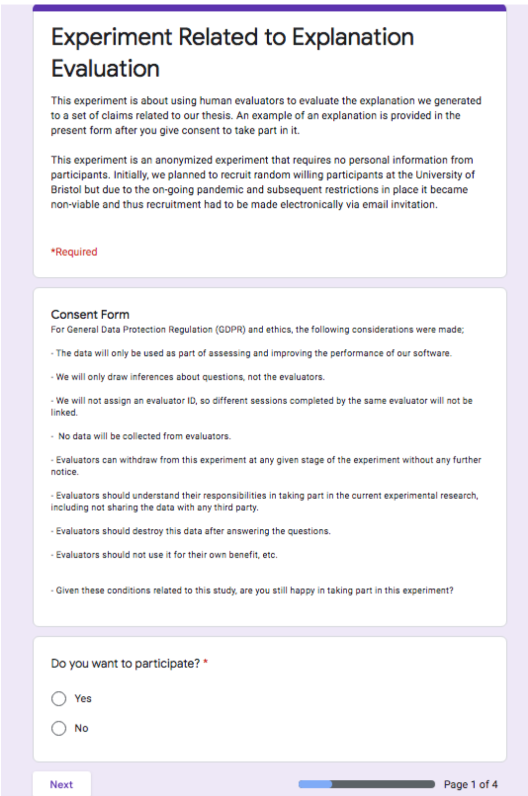
Here is a list of the logical rules we use in our PSL model:

Logical Rules
$Male(X) \Rightarrow \neg Female(X)$
$Parent\_Of(X,Y) \wedge X \neq Y \Rightarrow Ancestor\_Of(X,Y)$
$Ancestor\_Of(X,Y) \wedge Ancestor\_Of(Y,Z) \wedge Y \neq X \wedge Y \neq Z \Rightarrow Ancestor\_Of(X,Z)$
$Descendent\_Of(Y,X) \wedge Y \neq X \Rightarrow Ancestor\_Of(X,Y)$
$Parent\_Of(X,Y) \wedge Y \neq X \Rightarrow Descendent\_Of(Y,X)$
$Descendent\_Of(X,Y) \wedge Descendent\_Of(Y,Z) \wedge Y \neq X \wedge Y \neq Z \Rightarrow Descendent\_Of(X,Z)$
$Ancestor\_Of(X,Y) \wedge Y \neq X \Rightarrow Descendent\_Of(Y,X)$
$Parent\_Of(X,Y) \wedge Male(X) \wedge Y \neq X \Rightarrow Father\_Of(X,Y)$
$Parent\_Of(X,Y) \wedge Female(X) \wedge X \neq Y \Rightarrow Mother\_Of(X,Y)$
$Spouse\_Of(X,Y) \wedge Female(X) \wedge Y \neq X \Rightarrow Wife\_Of(X,Y)$
$Spouse\_Of(X,Y) \wedge Male(X) \wedge Y \neq X \Rightarrow Husband\_Of(X,Y)$
$Parent\_Of(X,Y) \wedge Y \neq X \Rightarrow Child\_Of(Y,X)$
$Parent\_Of(X,Y) \wedge Male(Y) \wedge Y \neq X \Rightarrow Son\_Of(Y,X)$
$Parent\_Of(X,Y) \wedge Female(Y) \wedge Y \neq X \Rightarrow Daughter\_Of(Y,X)$
$Parent\_Of(X,Y) \wedge Parent\_Of(X,Z) \wedge Y \neq Z \Rightarrow Sibling\_Of(Y,Z)$
$Sibling\_Of(Y,Z) \wedge Male(Y) \wedge (Y \neq Z) \Rightarrow Brother\_Of(Y,Z)$
$Sibling\_Of(Y,Z) \wedge Female(Y) \wedge (Y \neq Z) \Rightarrow Sister\_Of(Y,Z)$
$Parent\_Of(X,Z) \wedge Sibling\_Of(Y,X) \wedge Female(Y) \wedge X \neq Z \wedge Y \neq X \Rightarrow Aunt\_Of(Y,Z)$
$Uncle\_Of(X,Z) \wedge Spouse\_Of(Y,X) \wedge X \neq Z \wedge Y \neq X \Rightarrow AuntInLaw\_Of(Y,Z)$
$Aunt\_Of(X,Z) \wedge Spouse\_Of(Z,Y) \wedge X \neq Z \wedge Y \neq X \Rightarrow AuntInLaw\_Of(X,Y)$
$Parent\_Of(X,Z) \wedge Brother\_Of(Y,X) \wedge X \neq Z \wedge Y \neq X \Rightarrow Uncle\_Of(Y,Z)$
$Aunt\_Of(X,Z) \wedge Spouse\_Of(Y,X) \wedge X \neq Z \wedge Y \neq X \Rightarrow UncleInLaw\_Of(Y,Z)$
$Uncle\_Of(X,Z) \wedge Spouse\_Of(Z,Y) \wedge X \neq Z \wedge Y \neq X \Rightarrow UncleInLaw\_Of(X,Y)$
$Aunt\_Of(Z,X) \wedge Female(X) \wedge X \neq Z \Rightarrow Niece\_Of(X,Z)$
$Uncle\_Of(Z,X) \wedge Female(X) \wedge X \neq Z \Rightarrow Niece\_Of(X,Z)$
$UncleInLaw\_Of(Y,X) \wedge Female(X) \wedge X \neq Y \Rightarrow NieceInLaw\_Of(X,Y)$
$AuntInLaw\_Of(Y,X) \wedge Female(X) \wedge X \neq Y \Rightarrow NieceInLaw\_Of(X,Y)$
$Uncle\_Of(Y,X) \wedge Male(X) \wedge X \neq Y \Rightarrow Nephew\_Of(X,Y)$

Logical Rules
$Aunt\_Of(Y,X) \wedge Male(X) \wedge X \neq Y \Rightarrow Nephew\_Of(X,Y)$
$UncleInLaw\_Of(Y,X) \wedge Male(X) \wedge X \neq Y \Rightarrow NephewInLaw\_Of(X,Y)$
$AuntInLaw\_Of(Y,X) \wedge Male(X) \wedge X \neq Y \Rightarrow NephewInLaw\_Of(X,Y)$
$Parent\_Of(X,B) \wedge Parent\_Of(Z,A) \wedge A \neq B \wedge Sibling\_Of(X,Z) \wedge X \neq Z \Rightarrow Cousin\_Of(B,A)$
$Parent\_Of(X,B) \wedge Parent\_Of(Z,A) \wedge A \neq B \wedge SiblingInLaw\_Of(X,Z) \wedge X \neq Z \Rightarrow Cousin\_Of(B,A)$
$Cousin\_Of(Z,X) \wedge Spouse\_Of(X,Y) \wedge X \neq Z \wedge Y \neq X \Rightarrow CousinInLaw\_Of(Z,Y)$
$Parent\_Of(X,Z) \wedge Spouse\_Of(Y,Z) \wedge X \neq Z \wedge Y \neq Z \Rightarrow ChildInLaw\_Of(Y,X)$
$ChildInLaw\_Of(Y,X) \wedge Male(Y) \wedge Y \neq X \Rightarrow SonInLaw\_Of(Y,X)$
$ChildInLaw\_Of(Y,X) \wedge Female(Y) \wedge Y \neq X \Rightarrow DaughterInLaw\_Of(Y,X)$
$Spouse\_Of(X,Y) \wedge Parent\_Of(Z,X) \wedge Y \neq X \wedge Z \neq X \Rightarrow ParentInLaw\_Of(Z,Y)$
$ParentInLaw\_Of(Z,Y) \wedge Female(Z) \wedge Y \neq Z \Rightarrow MotherInLaw\_Of(Z,Y)$
$ParentInLaw\_Of(Z,Y) \wedge Male(Z) \wedge Y \neq Z \Rightarrow FatherInLaw\_Of(Z,Y)$
$Spouse\_Of(X,Y) \wedge Sibling\_Of(X,Z) \wedge Y \neq X \wedge X \neq Z \Rightarrow SiblingInLaw\_Of(Y,Z)$
$Spouse\_Of(X,Y) \wedge Sibling\_Of(Z,X) \wedge Y \neq X \wedge X \neq Z \Rightarrow SiblingInLaw\_Of(Z,Y)$
$SiblingInLaw\_Of(Z,Y) \wedge Female(Z) \wedge Z \neq Y \Rightarrow SisterInLaw\_Of(Z,Y)$
$SiblingInLaw\_Of(Z,Y) \wedge Male(Z) \wedge Z \neq Y \Rightarrow BrotherInLaw\_Of(Z,Y)$
$Parent\_Of(X,Y) \wedge Parent\_Of(Y,Z) \wedge X \neq Y \wedge Z \neq Y \Rightarrow GrandChild\_Of(Z,X)$
$GrandChild\_Of(Y,X) \wedge Male(Y) \wedge X \neq Y \Rightarrow Grandson\_Of(Y,X)$
$GrandChild\_Of(Y,X) \wedge Female(Y) \wedge X \neq Y \Rightarrow Granddaughter\_Of(Y,X)$
$Parent\_Of(X,Y) \wedge Parent\_Of(Y,A) \wedge X \neq Y \wedge A \neq Y \Rightarrow GrandParent\_Of(X,A)$
$GrandParent\_Of(X,Y) \wedge Male(X) \wedge X \neq Y \Rightarrow Grandfather\_Of(X,Y)$
$GrandParent\_Of(X,Y) \wedge Female(X) \wedge X \neq Y \Rightarrow Grandmother\_Of(X,Y)$

Appendix E

Appendix E: Evaluator Google Form



Experiment Related to Explanation Evaluation

This experiment is about using human evaluators to evaluate the explanation we generated to a set of claims related to our thesis. An example of an explanation is provided in the present form after you give consent to take part in it.

This experiment is an anonymized experiment that requires no personal information from participants. Initially, we planned to recruit random willing participants at the University of Bristol but due to the on-going pandemic and subsequent restrictions in place it became non-viable and thus recruitment had to be made electronically via email invitation.

*Required

Consent Form

For General Data Protection Regulation (GDPR) and ethics, the following considerations were made;

- The data will only be used as part of assessing and improving the performance of our software.
- We will only draw inferences about questions, not the evaluators.
- We will not assign an evaluator ID, so different sessions completed by the same evaluator will not be linked.
- No data will be collected from evaluators.
- Evaluators can withdraw from this experiment at any given stage of the experiment without any further notice.
- Evaluators should understand their responsibilities in taking part in the current experimental research, including not sharing the data with any third party.
- Evaluators should destroy this data after answering the questions.
- Evaluators should not use it for their own benefit, etc.
- Given these conditions related to this study, are you still happy in taking part in this experiment?

Do you want to participate? *

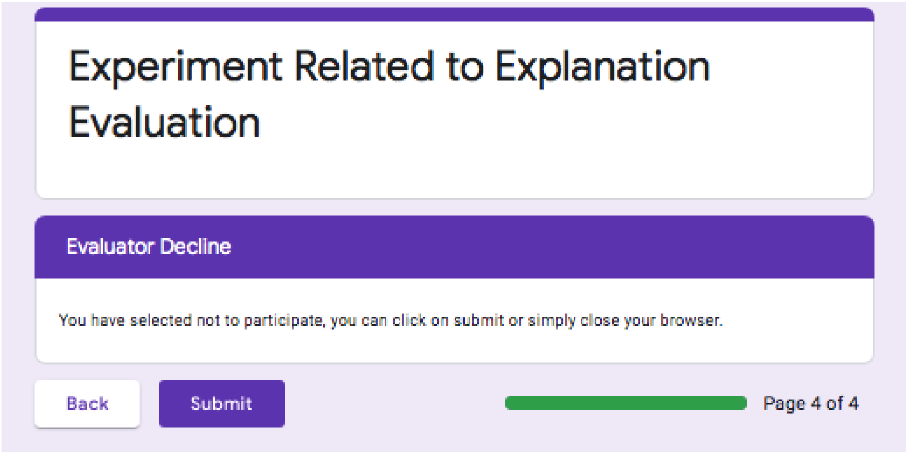
Yes

No

Next

Page 1 of 4

Figure E.1: Consent form and confirmation of participation in study



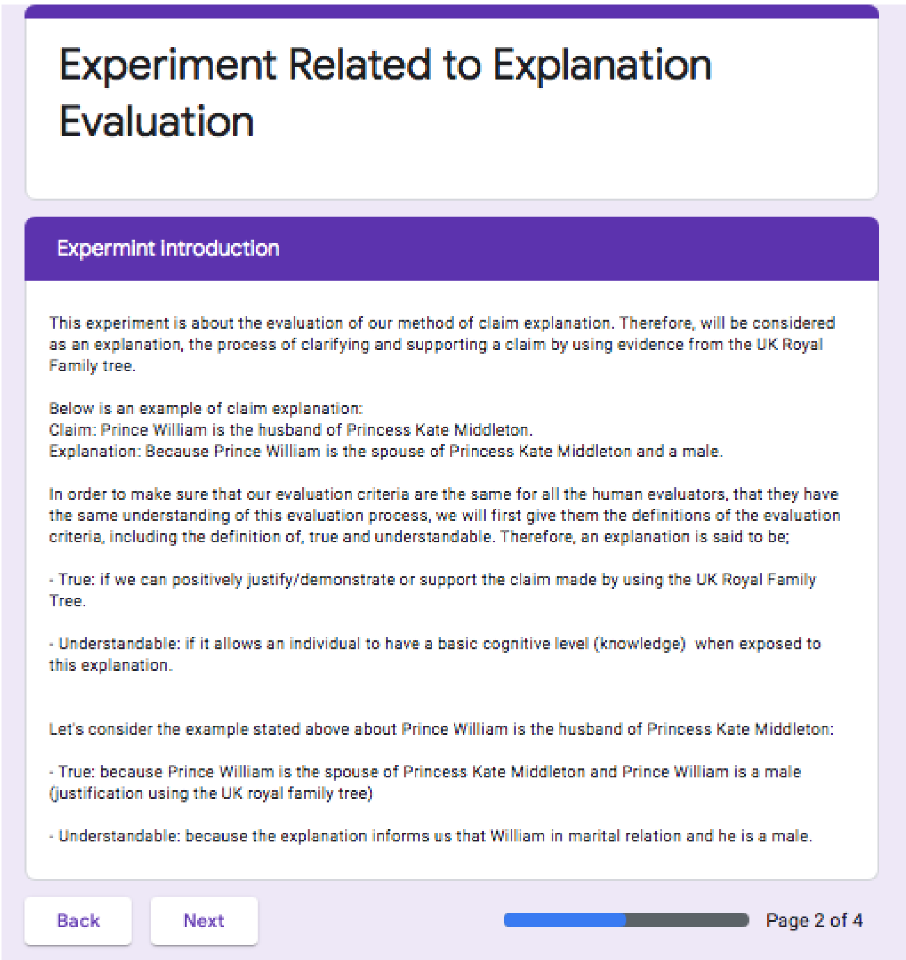
Experiment Related to Explanation Evaluation

Evaluator Decline

You have selected not to participate, you can click on submit or simply close your browser.

Back Submit Page 4 of 4

Figure E.2: Case 1: Response if answer is 'NO'



Experiment Related to Explanation Evaluation

Expermint Introduction

This experiment is about the evaluation of our method of claim explanation. Therefore, will be considered as an explanation, the process of clarifying and supporting a claim by using evidence from the UK Royal Family tree.

Below is an example of claim explanation:
Claim: Prince William is the husband of Princess Kate Middleton.
Explanation: Because Prince William is the spouse of Princess Kate Middleton and a male.

In order to make sure that our evaluation criteria are the same for all the human evaluators, that they have the same understanding of this evaluation process, we will first give them the definitions of the evaluation criteria, including the definition of, true and understandable. Therefore, an explanation is said to be;

- True: if we can positively justify/demonstrate or support the claim made by using the UK Royal Family Tree.

- Understandable: if it allows an individual to have a basic cognitive level (knowledge) when exposed to this explanation.

Let's consider the example stated above about Prince William is the husband of Princess Kate Middleton:

- True: because Prince William is the spouse of Princess Kate Middleton and Prince William is a male (justification using the UK royal family tree)

- Understandable: because the explanation informs us that William in marital relation and he is a male.

Back Next Page 2 of 4

Figure E.3: Case 2: Response if answer is 'YES'

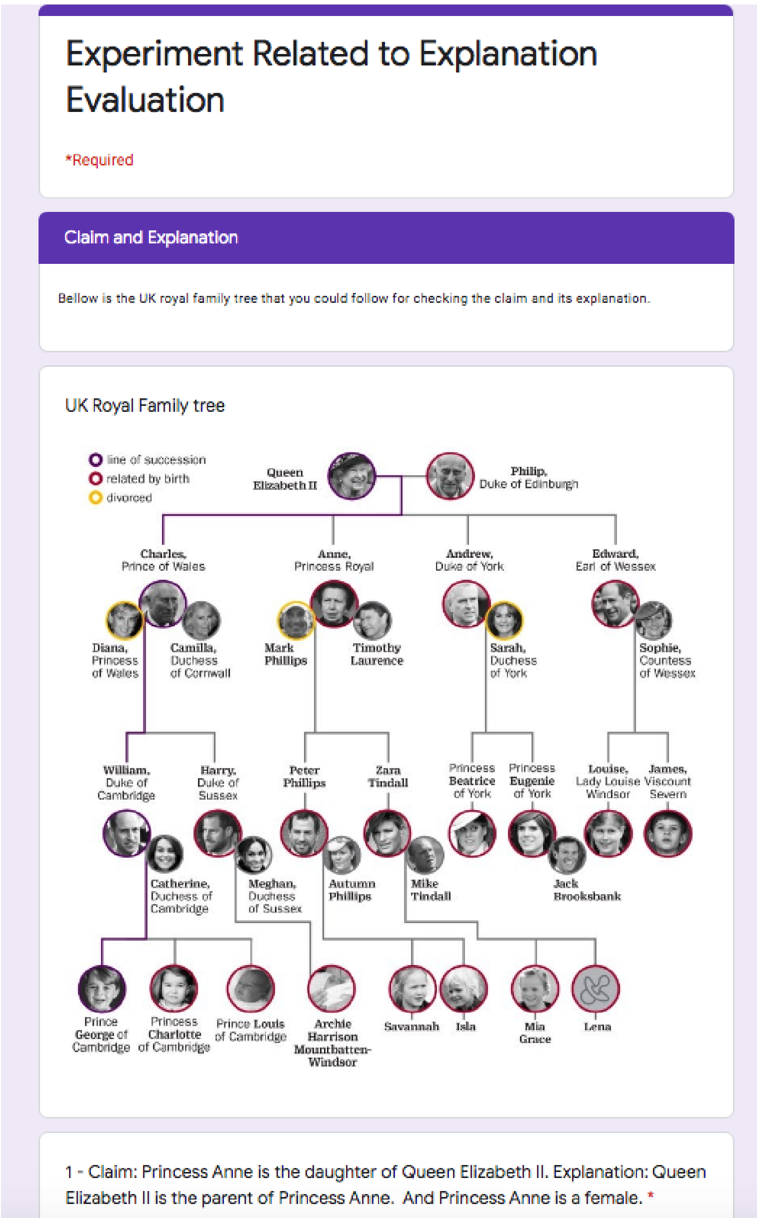
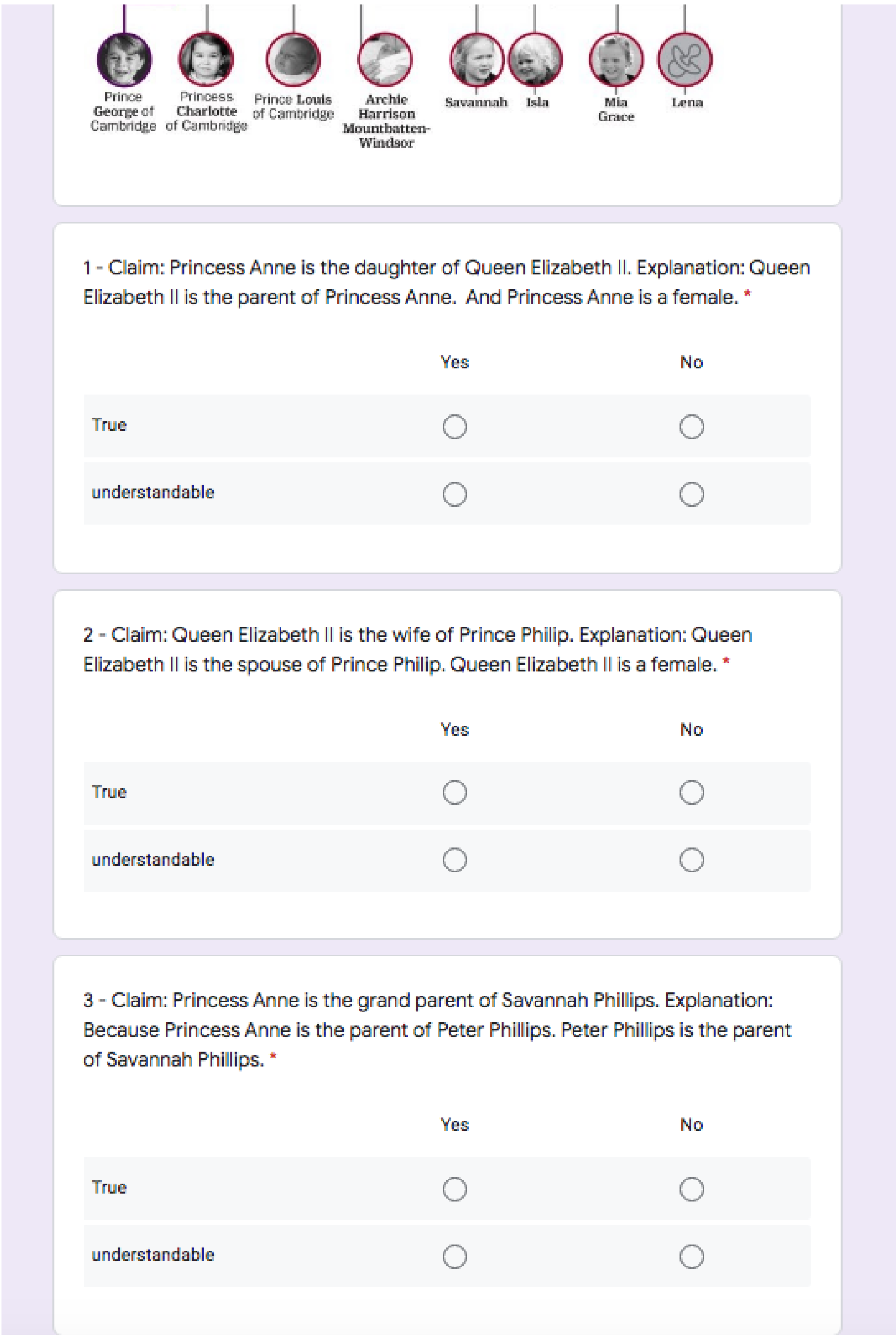


Figure E.4: The UK royal family tree needs to be followed by the Evaluators



Prince George of Cambridge
Princess Charlotte of Cambridge
Prince Louis of Cambridge
Archie Harrison Mountbatten-Windsor
Savannah
Isla
Mia Grace
Lena

1 - Claim: Princess Anne is the daughter of Queen Elizabeth II. Explanation: Queen Elizabeth II is the parent of Princess Anne. And Princess Anne is a female. *

	Yes	No
True	○	○
understandable	○	○

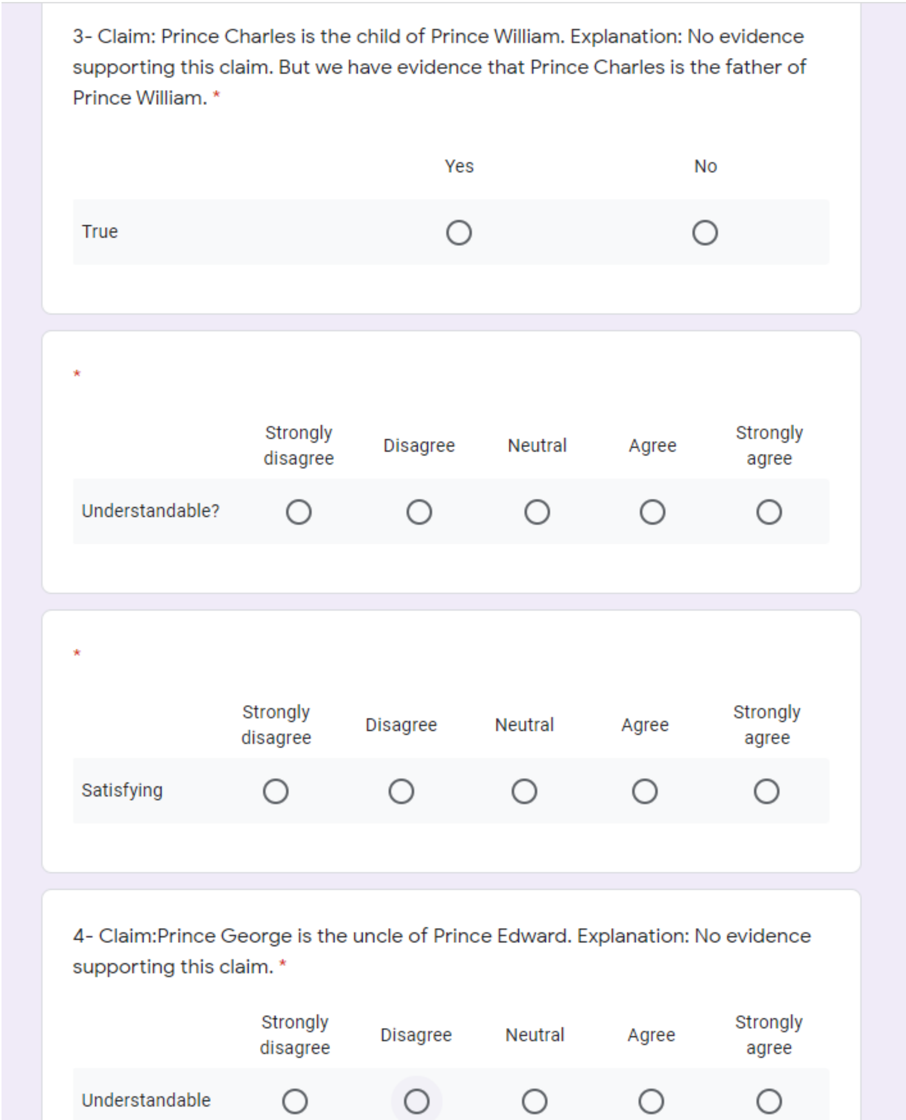
2 - Claim: Queen Elizabeth II is the wife of Prince Philip. Explanation: Queen Elizabeth II is the spouse of Prince Philip. Queen Elizabeth II is a female. *

	Yes	No
True	○	○
understandable	○	○

3 - Claim: Princess Anne is the grand parent of Savannah Phillips. Explanation: Because Princess Anne is the parent of Peter Phillips. Peter Phillips is the parent of Savannah Phillips. *

	Yes	No
True	○	○
understandable	○	○

Figure E.5: Experiment 1 claims and explanations example in Google form



3- Claim: Prince Charles is the child of Prince William. Explanation: No evidence supporting this claim. But we have evidence that Prince Charles is the father of Prince William. *

	Yes	No
True	○	○

*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Understandable?	○	○	○	○	○

*

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Satisfying	○	○	○	○	○

4- Claim:Prince George is the uncle of Prince Edward. Explanation: No evidence supporting this claim. *

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Understandable	○	○	○	○	○

Figure E.6: Experiment 2 claims and explanations example in Google form

APPENDIX

F

APPENDIX F: CLAIM AND EXPLANATION IN GOOGLE FORM

This table bellow contains the claims and explanations for experiment 1 and 2

Claim and Explanation for Experiment 1		
Explana-tionID	Claim	Explanation
Ex1	Autumn Phillips is the child-In-Law of Princess Anne.	Princess Anne is the parent of Peter Phillips. And Autumn Phillips is the spouse of Peter Phillips.
Ex2	Princess Eugenie is the sister of Princess Beatrice.	Princess Eugenie is the sibling of Princess Beatrice. Because Prince Andrew is the parent of Princess Eugenie and Prince Andrew is the parent of Princess Beatrice. And Princess Eugenie is a female.
Ex3	Prince George is the son of Catherine Middleton.	Catherine Middleton is the parent of Prince George. And Prince George is a male.
Ex4	Queen Elizabeth II is the grand-mother of Prince William.	Queen Elizabeth II is the parent of Prince Charles and Prince Charles is the parent of Prince William. And Queen Elizabeth II is a female.
Ex5	Princess Anne is the grandparent of Savannah Phillips.	Peter Phillips is the parent of Savannah Phillips and Princess Anne is the parent of Peter Phillips.
Ex6	Princess Beatrice is the grand-daughter of Queen Elizabeth II.	Queen Elizabeth II is the parent of Prince Andrew and Prince Andrew is the parent of Princess Beatrice. And Princess Beatrice is a female.
Ex7	Queen Elizabeth II is the mother-In-Law of Princess Diana.	Prince Charles is the spouse of Princess Diana and Queen Elizabeth II is the parent of Prince Charles. And Queen Elizabeth II is a female.
Ex8	Catherine Middleton is the sister-In-Law of Prince Harry.	Catherine Middleton is the spouse of Prince William. Prince Charles is the parent of Prince William and Prince Charles is the parent of Prince Harry.
Ex9	Queen Elizabeth II is the wife of Prince Philip.	Queen Elizabeth II is the spouse of Prince Philip and Queen Elizabeth II is a female.
Ex10	Princess Anne is the daughter of Queen Elizabeth II.	Queen Elizabeth II is the parent of Princess Anne and Princess Anne is female.

Claim and Explanation for Experiment 2		
ExplanationID	Claim	Explanation
Ex1	Princess Anne is the daughter of Queen Elizabeth II.	Queen Elizabeth II is the parent of Princess Anne. And Princess Anne is a female.
Ex2	Queen Elizabeth II is the wife of Prince Philip.	Queen Elizabeth II is the spouse of Prince Philip. Queen Elizabeth II is a female.
Ex3	Princess Anne is the grand parent of Savannah Phillips.	Princess Anne is the parent of Peter Phillips. Peter Phillips is the parent of Savannah Phillips.
Ex4	Prince William is the nephew of Princess Anne.	Queen Elizabeth II is the parent of Prince Charles. Queen Elizabeth II is the parent of Princess Anne. Prince Charles is the parent of Prince William. and Prince William is male.
Ex5	Prince Harry is the child of Prince Charles.	Prince Charles is the parent of Prince Harry.
Ex6	Prince Edward is the brother-In-Law of Princess Diana.	Princess Diana is the spouse of Prince Charles. Queen Elizabeth II is the parent of Prince Charles. Queen Elizabeth II is the parent of Prince Prince Edward. Prince Edward is a male.
Ex7	Prince Harry is the grand child of Queen Elizabeth II.	Queen Elizabeth II is the parent of Prince Charles. Prince Charles is the parent of Prince Harry.
Ex8	Prince Charles is the uncle of Princess Eugenie.	Prince Andrew is the parent of Princess Eugenie. Queen Elizabeth II is the parent of Prince Andrew. Queen Elizabeth II is the parent of Prince Charles. Prince Charles is a male.
Ex9	Princess Anne is the aunt of Prince William.	Prince Charles is the parent of Prince William. Queen Elizabeth II is the parent of Prince Charles. Queen Elizabeth II is the parent of Princess Anne. Princess Anne is a female.
Ex10	Prince William is the cousin of Princess Beatrice.	Prince Charles is the parent of Prince William. Prince Andrew is the parent of Princess Beatrice. Queen Elizabeth II is the parent of Prince Charles. Queen Elizabeth II is the parent of Prince Andrew.

APPENDIX G

APPENDIX G: EVALUATORS EXPERIMENT RESULTS

G.1 Evaluators Evaluation results for the First and Second experiment

Table G.1: Responses of human evaluators to the process of evaluating explanations, where 'Exs' represents an explanation of a claim in both the First and Second Experiments (see Appendix F for more details)

FACT Explanation Evaluation for the First Experiment					
Evaluator #	Explanation #	True		Understandable	
30 Evaluators	Ex1	23	76.67%	28	93.33%
	Ex2	29	96.67%	28	93.33%
	Ex3	28	93.33%	29	96.67%
	Ex4	29	96.67%	29	96.67%
	Ex5	26	86.67%	29	96.67%
	Ex6	29	96.67%	28	93.33%
	Ex7	26	86.67%	26	86.67%
	Ex8	28	93.33%	29	96.67%
	Ex9	29	96.67%	28	93.33%
	Ex10	30	100%	30	100%
	Total	277		284	
FACT Explanation Evaluation for the Second Experiment					
Evaluator #	Explanation #	True		Understandable	
31 Evaluators	Ex1	31	100%	31	100%
	Ex2	31	100%	31	100%
	Ex3	27	87.10%	24	77.42%
	Ex4	31	100%	25	80.65%
	Ex5	31	100%	30	96.77%
	Ex6	26	83.87%	23	74.19%
	Ex7	31	100%	25	80.65%
	Ex8	31	100%	29	93.55%
	Ex9	31	100%	23	74.19%
	Ex10	27	87.10%	28	90.32%
	Total	297		269	

G.2 Evaluators Evaluation results for the third experiment

G.2.1 First criteria results

Criteria	Claim	FACT	CredEye
Truthfulness	ex1	96.6	16.6
	ex2	96.6	93.3
	ex3	96.6	90
	ex4	93.3	3.3

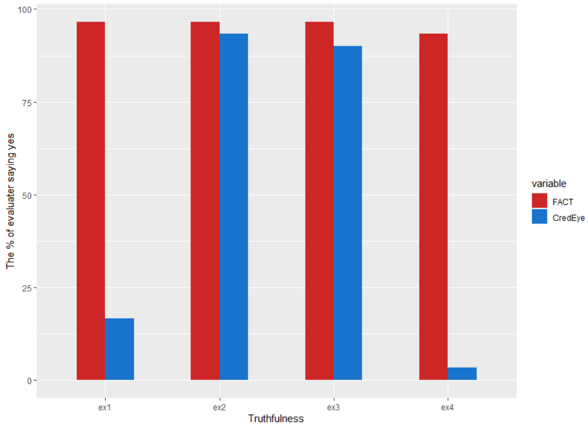


Table G.3: Result for the 'Truthfulness' criterion

G.2.2 Second criteria results

Criteria	Claim	Scale	FACT	CredEye
Understandable	ex1	Totally disagree	3.3	50
		Disagree	3.3	33.3
		Neutral	3.3	13.3
		Agree	36.6	3.3
		Totally agree	53.3	0
	ex2	Totally disagree	3.3	33.3
		Disagree	3.3	26.6
		Neutral	0	30
		Agree	33.3	10
		Totally agree	60	0
	ex3	Totally disagree	0	33.3
		Disagree	3.3	23.33
		Neutral	3.3	26.6
		Agree	40	16.6
		Totally agree	53.3	0
	ex4	Totally disagree	10	66.66
		Disagree	0	13.3
		Neutral	0	6.6
		Agree	46.6	10
		Totally agree	43.3	3.3

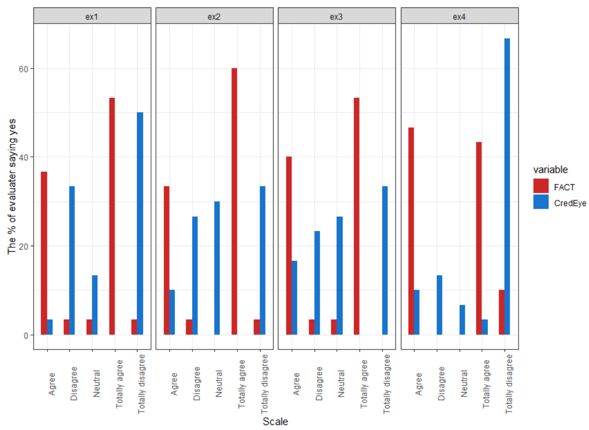


Table G.4: Result for the 'Understandable' criterion

G.2.3 Third criteria results

Criteria	Claim	Scale	FACT	CredEye
Satisfying	ex1	Totally disagree	3.3	53.33
		Disagree	0	40
		Neutral	3.3	3.3
		Agree	46.6	3.3
		Totally agree	46.6	0
	ex2	Totally disagree	0	43.3
		Disagree	3.3	30
		Neutral	3.3	20
		Agree	40	6.6
		Totally agree	53.3	0
	ex3	Totally disagree	3.3	30
		Disagree	3.3	40
		Neutral	10	16.6
		Agree	33.3	13.3
		Totally agree	50	0
	ex4	Totally disagree	0	76.6
		Disagree	16.6	10
		Neutral	43.3	3.3
		Agree	20	6.6
		Totally agree	20	3.3

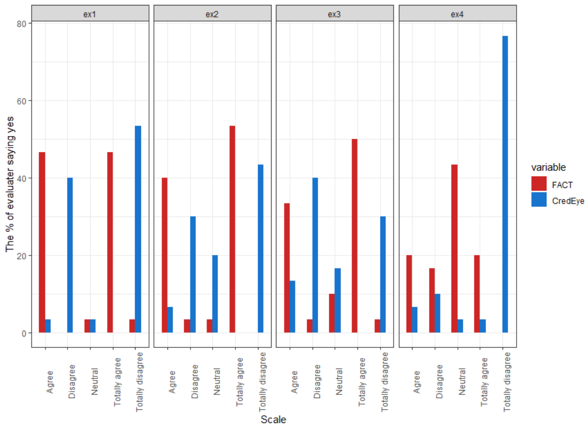


Table G.5: Result for the ‘Understandable’ criterion

Bibliography

[1] J. Hendler and T. Berners-Lee, “From the semantic web to social machines: A research challenge for ai on the world wide web,” *Artificial intelligence*, vol. 174, no. 2, pp. 156—-161, Feb. 2010. [Online]. Available: https://doi.org/10.1016/j.artint.2009.11.010

[2] A. Majchrzak, S. Faraj, G. C. Kane, and B. Azad, “The contradictory influence of social media affordances on online communal knowledge sharing,” *Journal of Computer-Mediated Communication*, vol. 19, no. 1, pp. 38–55, Oct. 2013. [Online]. Available: https://doi.org/10.1111/jcc4.12030

[3] B. Reese, *Infinite progress: How the internet and technology will end ignorance, disease, poverty, hunger, and war*. Austin, Texas: Greenleaf Book Group Press, 2013.

[4] E. Yamato, S. E. Krauss, E. Tamam, H. Hassan, and M. N. Osman, “It’s part of our lifestyle: Exploring young malaysians’ experiences with japanese popular culture,” *Keio Communication Review*, no. 33, pp. 199–223, Mar. 2011.

[5] E. B. Mandinach, E. S. Gummer, and R. D. Muller, “The complexities of integrating data-driven decision making into professional preparation in schools of education: It’s harder than you think,” in *Report from an invitational meeting*. Alexandria, VA, Portland, OR, and Washington, DC: CNA Education, Education Northwest, and WestEd, May 2011.

[6] V. Jayagopal and K. Basser, “Data management and big data analytics: Data management in digital economy,” in *Optimizing Big Data Management and Industrial Systems With Intelligent Techniques*. IGI Global, 2019, pp. 1–23.

[7] D. P. Acharjya and K. A. P, “A survey on big data analytics: challenges, open research issues and tools,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 511–518, 2016. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2016.070267

[8] F. Piccialli, S. Cuomo, and G. Jeon, “Parallel approaches for data mining in the internet of things realm,” pp. 807–811, Mar. 2018. [Online]. Available: https://doi.org/10.1007/s10766-018-0565-y

[9] R. R. Larson, “Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace,” in *Proceedings of the Annual Meeting-American Society for Information Science*, vol. 33, 1996, pp. 71–78. [Online]. Available: http://hdl.handle.net/10150/106530

[10] X. Yin, J. Han, and S. Y. Philip, “Truth discovery with multiple conflicting information providers on the web,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, Jun. 2008.

[11] C. Shao, G. L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, and F. Menczer, “The spread of low-credibility content by social bots,” *Nature communications*, vol. 9, no. 1, p. 4787, Nov. 2018. [Online]. Available: https://doi.org/10.1038/s41467-018-06930-7

[12] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, “Credibility assessment of textual claims on the web,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM ’16, ACM. New York, NY, USA: Association for Computing Machinery, 2016, pp. 2173–2178. [Online]. Available: https://doi.org/10.1145/2983323.2983661

[13] A. K. Chaudhry, D. Baker, and P. Thun-Hohenstein, “Stance detection for the fake news challenge: identifying textual relationships with deep neural nets,” *CS224n: Natural Language Processing with Deep Learning*, 2017. [Online]. Available: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760230.pdf

[14] R. Masood and A. Aker, “The fake news challenge: Stance detection using traditional machine learning approaches.” in *KMIS*, 2018, pp. 126–133. [Online]. Available: https://www.scitepress.org/Papers/2018/68988/68988.pdf

[15] S. Zhi, F. Yang, Z. Zhu, Q. Li, Z. Wang, and J. Han, “Dynamic truth discovery on numerical data,” in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 817–826. [Online]. Available: https://doi.org/10.1109/ICDM.2018.00097

[16] V. Woloszyn, E. G. Cortes, R. Amantea, V. Schmitt, D. A. Barone, and S. Möller, “Towards a novel benchmark for automatic generation of claimreview markup,” in *13th ACM Web Science Conference 2021 (WebSic ’21)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 29–35. [Online]. Available: https://doi.org/10.1145/3447535.3462640

[17] N. Vo and K. Lee, “Learning from fact-checkers: Analysis and generation of fact-checking language,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’19*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 335–344. [Online]. Available: https://doi.org/10.1145/3331184.3331248

[18] M. L. Ba, L. Berti-Equille, K. Shah, and H. M. Hammady, "Vera: A platform for veracity estimation over web data," in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW '16 Companion, International World Wide Web Conferences Steering Committee. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 159–162. [Online]. Available: https://doi.org/10.1145/2872518.2890536

[19] N. Hassan, B. Adair, J. T. Hamilton, C. Li, M. Tremayne, J. Yang, and C. Yu, "The quest to automate fact-checking. world (2015)," *Proceedings of the 2015 Computation + Journalism Symposium*, Oct. 2015.

[20] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak *et al.*, "Claimbuster: the first-ever end-to-end fact-checking system," *Proceedings of the VLDB Endowment*, vol. 10, no. 12, pp. 1945–1948, Aug. 2017. [Online]. Available: https://doi.org/10.14778/3137765.3137815

[21] Y. Wu, B. Walenz, P. Li, A. Shim, E. Sonmez, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "icheck: computationally combating "lies, d–ned lies, and statistics"," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: Association for Computing Machinery, Jun. 2014, pp. 1063–1066. [Online]. Available: https://doi.org/10.1145/2588555.2594522

[22] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," *arXiv preprint arXiv:1806.07687*, 2018.

[23] T. Mihaylova, P. Nakov, L. Marquez, A. Barron-Cedeno, M. Mohtarami, G. Karadzhov, and J. Glass, "Fact checking in community forums," in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, Mar. 2018.

[24] N. Bindris, S. Sudhahar, and N. Cristianini, "Fact checking from natural text with probabilistic soft logic," in *Advances in Intelligent Data Analysis XVII*, W. Duivesteijn, A. Siebes, and A. Ukkonen, Eds. Cham: Springer International Publishing, 2018, pp. 52–61.

[25] N. Bindris, N. Cristianini, and J. Lawry, "Claim consistency checking using soft logic," *Machine Learning and Knowledge Extraction*, vol. 2, no. 3, pp. 147–171, 2020.

[26] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2931–2937. [Online]. Available: https://aclanthology.org/D17-1317

[27] N. Nakashole and T. M. Mitchell, “Language-aware truth assessment of fact candidates,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1009–1019. [Online]. Available: https://aclanthology.org/P14-1095

[28] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti, “Automatic stance detection using end-to-end memory networks,” *arXiv preprint arXiv:1804.07581*, pp. 767–776, Jun. 2018. [Online]. Available: https://aclanthology.org/N18-1070

[29] A. Vlachos and S. Riedel, “Fact checking: Task definition and dataset construction,” in *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*. Baltimore, MD, USA: Association for Computational Linguistics, Jun. 2014, pp. 18–22. [Online]. Available: https://aclanthology.org/W14-2508

[30] J. Thorne, M. Chen, G. Myrianthous, J. Pu, X. Wang, and A. Vlachos, “Fake news stance detection using stacked ensemble of classifiers,” in *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 80–83. [Online]. Available: https://aclanthology.org/W17-4214

[31] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, “Credeye: A credibility lens for analyzing and explaining misinformation,” in *Companion Proceedings of the The Web Conference 2018*, ser. WWW ’18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2018, pp. 155–158. [Online]. Available: https://doi.org/10.1145/3184558.3186967

[32] J. Thorne and A. Vlachos, “An extensible framework for verification of numerical claims,” in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 37–40. [Online]. Available: https://aclanthology.org/E17-3010

[33] N. Hassan, F. Arslan, C. Li, and M. Tremayne, “Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1803–1812. [Online]. Available: https://doi.org/10.1145/3097983.3098131

[34] G. Karadzhov, P. Nakov, L. Màrquez, A. Barrón-Cedeño, and I. Koychev, “Fully automated fact checking using external sources,” in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*.

Varna, Bulgaria: INCOMA Ltd., Sep. 2017, pp. 344–353. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_046

[35] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, and P. Nakov, "Integrating stance detection and fact checking in a unified corpus," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 21–27. [Online]. Available: https://aclanthology.org/N18-2004

[36] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini, "Computational fact checking from knowledge networks," *PLOS ONE*, vol. 10, no. 6, p. e0128193, Jun. 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0128193

[37] B. Shi and T. Weninger, "Fact checking in heterogeneous information networks," in *Proceedings of the 25th International Conference Companion on World Wide Web*, ser. WWW '16 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2016, pp. 101–102. [Online]. Available: https://doi.org/10.1145/2872518.2889354

[38] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, pp. 1003–1012. [Online]. Available: https://doi.org/10.1145/3041021.3055133

[39] A. Vlachos and S. Riedel, "Identification and verification of simple claims about statistical properties," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2596–2601. [Online]. Available: https://aclanthology.org/D15-1312

[40] J. Leblay, "A declarative approach to data-driven fact checking." in *AAAI*, 2017, pp. 147–153.

[41] J. Leblay, W. Chen, and S. Lynden, "Exploring the veracity of online claims with backdrop," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 2491–2494.

[42] A. Patwari, D. Goldwasser, and S. Bagchi, "Tathya: A multi-classifier system for detecting check-worthy statements in political debates," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 2259–2262.

[43] S. Jo, I. Trummer, W. Yu, D. Liu, and N. Mehta, "The factchecker: Verifying text summaries of relational data sets," *arXiv preprint arXiv:1804.07686*, 2018.

[44] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn *et al.*, "Clinical information extraction applications: a literature review," *Journal of biomedical informatics*, vol. 77, pp. 34–49, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046417302563

[45] J. Piskorski and R. Yangarber, "Information extraction: past, present and future," in *Multi-source, multilingual information extraction and summarization*, T. Poibeau, H. Saggion, J. Piskorski, and R. Yangarber, Eds. Springer, Berlin, Heidelberg, 2013, pp. 23–49. [Online]. Available: https://doi.org/10.1007/978-3-642-28569-1_2

[46] F. Hogenboom, F. Frasincar, U. Kaymak, F. De Jong, and E. Caron, "A survey of event extraction methods from text for decision support systems," *Decision Support Systems*, vol. 85, pp. 12–22, May 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923616300173

[47] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173 111–173 137, Nov. 2019.

[48] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 39–48. [Online]. Available: https://aclanthology.org/W15-1506

[49] M. Garcia, "Semantic relation extraction. resources, tools and strategies," in *International Conference on Computational Processing of the Portuguese Language*, J. Silva, R. Ribeiro, P. Quaresma, A. Adami, and A. Branco, Eds. Cham: Springer International Publishing, 2016, pp. 141–152. [Online]. Available: https://doi.org/10.1007/978-3-319-41552-9_15

[50] L. Eikvil, "Information extraction from world wide web-a survey," Technical Report 945, Norwegian Computing Center, Tech. Rep., Jul. 1999. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.217.5237&rep=rep1&type=pdf

[51] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 10, pp. 1411–1428, Oct. 2006.

[52] K. Adnan and R. Akbar, "An analytical study of information extraction from unstructured and multidimensional big data," *Journal of Big Data*, vol. 6, no. 1, p. 91, Oct. 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0254-8

[53] L. Chiticariu, Y. Li, and F. Reiss, “Rule-based information extraction is dead! long live rule-based information extraction systems!” in *Proceedings of the 2013 conference on empirical methods in natural language processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 827–832. [Online]. Available: https://aclanthology.org/D13-1079

[54] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, “Named entity recognition: fallacies, challenges and opportunities,” *Computer Standards & Interfaces*, vol. 35, no. 5, pp. 482–489, Sep. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548912001080

[55] A. Goyal, V. Gupta, and M. Kumar, “Recent named entity recognition and classification techniques: a systematic review,” *Computer Science Review*, vol. 29, pp. 21–43, Jun. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013717302782

[56] M. Song, W. C. Kim, D. Lee, G. E. Heo, and K. Y. Kang, “PKDE4J: Entity and relation extraction for public knowledge discovery,” *Journal of Biomedical Informatics*, vol. 57, pp. 320–332, Aug. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046415001756

[57] D. Deng, G. Li, J. Feng, Y. Duan, and Z. Gong, “A unified framework for approximate dictionary-based entity extraction,” *The VLDB Journal*, vol. 24, no. 1, pp. 143–167, 2015. [Online]. Available: https://doi.org/10.1007/s00778-014-0367-9

[58] R. Grishman, “Information extraction: Techniques and challenges,” in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, T. Pazienza, Ed. Springer, Berlin, Heidelberg, 1997, pp. 10–27. [Online]. Available: https://doi.org/10.1007/3-540-63438-X_2

[59] A. Esuli and F. Sebastiani, “Evaluating information extraction,” in *Multilingual and Multimodal Information Access Evaluation*, M. Agosti, N. Ferro, C. Peters, M. de Rijke, and A. Smeaton, Eds. Springer, Berlin, Heidelberg, 2010, pp. 100–111. [Online]. Available: https://doi.org/10.1007/978-3-642-15998-5_12

[60] S. Jiang, S. Baumgartner, A. Ittycheriah, and C. Yu, “Factoring fact-checks: Structured information extraction from fact-checking articles,” in *Proceedings of The Web Conference 2020*, ser. WWW ’20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1592–1603. [Online]. Available: https://doi.org/10.1145/3366423.3380231

[61] S. Cazalens, P. Lamarre, J. Leblay, I. Manolescu, and X. Tannier, “A content management perspective on fact-checking,” in *Companion Proceedings of the The Web Conference*

2018, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2018, pp. 565–574. [Online]. Available: https://doi.org/10.1145/3184558.3188727

[62] Y. Wu, P. K. Agarwal, C. Li, J. Yang, and C. Yu, "Toward computational fact-checking," *Proceedings of the VLDB Endowment*, vol. 7, no. 7, pp. 589–600, Mar. 2014. [Online]. Available: https://doi.org/10.14778/2732286.2732295

[63] D. Graves, "Understanding the promise and limits of automated fact-checking," Web page, https://reutersinstitute.politics.ox.ac.uk/our-research/understanding-promise-and-limits-automated-fact-checking, 2018.

[64] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, May 2017.

[65] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3846–3912, Jan. 2017.

[66] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1, pp. 107–136, Jan. 2006. [Online]. Available: https://doi.org/10.1007/s10994-006-5833-1

[67] I. Beltagy, S. Roller, G. Boleda, K. Erk, and R. Mooney, "UTexas: Natural language semantics using distributional semantics and probabilistic logic," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 796–801. [Online]. Available: https://aclanthology.org/S14-2141

[68] J. Lee and Y. Wang, "On the semantic relationship between probabilistic soft logic and markov logic," *arXiv preprint arXiv:1606.08896*, Jun. 2016.

[69] A. Kimmig, S. Bach, M. Broecheler, B. Huang, and L. Getoor, "A short introduction to probabilistic soft logic," in *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*. Mansinghka, Vikash, 2012, pp. 1–4. [Online]. Available: https://lirias.kuleuven.be/retrieve/204697

[70] L. Deng and J. Wiebe, "Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 179–189. [Online]. Available: https://aclanthology.org/D15-1018

[71] S. Fakhraei, B. Huang, L. Raschid, and L. Getoor, “Network-based drug-target interaction prediction with probabilistic soft logic,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 5, pp. 775–787, Sept. 2014. [Online]. Available: https://doi.org/10.1109/TCBB.2014.2325031

[72] M. Bröcheler and L. Getoor, “Computing marginal distributions over continuous markov networks for statistical relational learning,” in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS’10. Red Hook, NY, USA: Curran Associates Inc., Dec. 2010, pp. 316–324.

[73] D. Sridhar, “Learning structured and causal probabilistic models for computational science,” Ph.D. dissertation, UC Santa Cruz, 2018. [Online]. Available: https://escholarship.org/uc/item/0xf1t2zr

[74] S. Bach, M. Broecheler, L. Getoor, and D. O’leary, “Scaling mpe inference for constrained continuous markov random fields with consensus optimization,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012, pp. 2654–2662. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf

[75] R. West, H. S. Paskov, J. Leskovec, and C. Potts, “Exploiting social network structure for person-to-person sentiment analysis,” *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 297–310, Oct. 2014. [Online]. Available: https://doi.org/10.1162/tacl_a_00184

[76] B. Huang, A. Kimmig, L. Getoor, and J. Golbeck, “A flexible framework for probabilistic models of social trust,” in *Proceedings of the 6th international conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, vol. 7812. Springer, Apr. 2013, pp. 265–273.

[77] T. Khot, N. Balasubramanian, E. Gribkoff, A. Sabharwal, P. Clark, and O. Etzioni, “Exploring markov logic networks for question answering,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 685–694. [Online]. Available: https://aclanthology.org/D15-1080

[78] P. Clark, P. Harrison, and N. Balasubramanian, “A study of the AKBC requirements for passing an elementary science test,” in *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, ser. AKBC ’13. New York, NY, USA: Association for Computing Machinery, Oct. 2013, pp. 37–42. [Online]. Available: https://doi.org/10.1145/2509558.2509565

[79] I. Beltagy, C. Chau, G. Boleda, D. Garrette, K. Erk, and R. Mooney, “Montague meets markov: Deep semantics with probabilistic logical form,” in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 11–21. [Online]. Available: https://aclanthology.org/S13-1002

[80] I. Beltagy and R. J. Mooney, “Efficient markov logic inference for natural language semantics,” in *Proceedings of the Fourth International Workshop on Statistical Relational AI at AAAI (StarAI-2014)*, Quebec City, Canada, Jul. 2014, pp. 9–14. [Online]. Available: http://www.cs.utexas.edu/users/ai-labpub-view.php?PubID=127446

[81] G. Farnadi, L. Getoor, M.-F. Moens, and M. De Cock, “User profiling using hinge-loss markov random fields,” *arXiv preprint arXiv:2001.01177*, Jan. 2020.

[82] S. Fakhraei, J. Foulds, M. Shashanka, and L. Getoor, “Collective spammer detection in evolving multi-relational social networks,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 1769–1778. [Online]. Available: https://doi.org/10.1145/2783258.2788606

[83] P. D. Turney and P. Pantel, “From frequency to meaning: Vector space models of semantics,” *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, Jan. 2010.

[84] J. Mitchell and M. Lapata, “Composition in distributional models of semantics,” *Cognitive science*, vol. 34, no. 8, pp. 1388–1429, Nov. 2010.

[85] A. Lenci and G. Benotto, “Identifying hypernyms in distributional semantic spaces,” in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, Jun. 2012, pp. 75–79. [Online]. Available: https://aclanthology.org/S12-1012

[86] S. Roller, K. Erk, and G. Boleda, “Inclusive yet selective: Supervised distributional hypernymy detection,” in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 1025–1036. [Online]. Available: https://aclanthology.org/C14-1097

[87] J. Bos, “Wide-coverage semantic analysis with boxer,” in *Semantics in Text Processing. STEP 2008 Conference Proceedings*. College Publications, 2008, pp. 277–286. [Online]. Available: https://aclanthology.org/W08-2222

[88] S. Clark and J. R. Curran, “Parsing the WSJ using CCG and log-linear models,” in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, ser. ACL ’04. USA: Association for Computational Linguistics, Jul. 2004, pp. 103–110. [Online]. Available: https://doi.org/10.3115/1218955.1218969

[89] I. Beltagy, K. Erk, and R. Mooney, “Probabilistic soft logic for semantic textual similarity,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1210–1219. [Online]. Available: https://aclanthology.org/P14-1114

[90] S. H. Bach, B. Huang, B. London, and L. Getoor, “Hinge-loss markov random fields: Convex inference for structured prediction,” *arXiv preprint arXiv:1309.6813*, pp. 32–41, Aug. 2013.

[91] A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor, “Modeling learner engagement in MOOCs using probabilistic soft logic,” in *NIPS workshop on data driven education*, vol. 21, 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.477.4319&rep=rep1&type=pdf

[92] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, “Opinionfinder: A system for subjectivity analysis,” in *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 34–35. [Online]. Available: https://aclanthology.org/H05-2018

[93] M. Samadi, P. Talukdar, M. Veloso, and M. Blum, “Claimeval: Integrated and flexible framework for claim evaluation using credibility of sources,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. AAAI Press, Feb. 2016, p. 222–228.

[94] A. Holzinger, “From machine learning to explainable ai,” in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*. IEEE, 2018, pp. 55–66.

[95] A. Duval, “Explainable artificial intelligence (xai),” *MA4K9 Scholarly Report, Mathematics Institute, The University of Warwick*, Apr. 2019.

[96] R. K. Sheh, “Different xai for different hri,” in *2017 AAAI Fall Symposium Series*, 2017, pp. 114–117. [Online]. Available: http://hdl.handle.net/20.500.11937/67997

[97] L. Ljung, “Black-box models from input-output measurements,” in *IMTC 2001. Proceedings of the 18th IEEE instrumentation and measurement technology conference. rediscovering*

measurement in the age of informatics (cat. no. 01CH 37188), vol. 1. IEEE, Jun. 2001, pp. 138–146.

[98] R. Schmelzer, “Understanding explainable ai,” WebPage, Available at https://www.forbes.com/sites/cognitiveworld/2019/07/23/understanding-explainable-ai/#1155fd267c9e (Accessed: 14 Jun. 2020).

[99] M. T. Ribeiro, S. Singh, and C. Guestrin, “" why should i trust you?" explaining the predictions of any classifier,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, Feb. 2016, pp. 97–101. [Online]. Available: https://aclanthology.org/N16-3020

[100] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (xai),” *IEEE Access*, vol. 6, pp. 52 138–52 160, Sept. 2018.

[101] W. Samek, T. Wiegand, and K.-R. Müller, “Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models,” *ITU Journal: ICT Discoveries - Special Issue 1 - The Impact of Artificial Intelligence (AI) on Communication Networks and Services*, vol. 1, pp. 1–10, Oct. 2017.

[102] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 1721—-1730. [Online]. Available: https://doi.org/10.1145/2783258.2788613

[103] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, “Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai,” *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://doi.org/10.1016/j.inffus.2019.12.012

[104] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019. [Online]. Available: https://doi.org/10.1016/j.artint.2018.07.007

[105] A. Alase, “The interpretative phenomenological analysis (ipa): A guide to a good qualitative research approach,” *International Journal of Education and Literacy Studies*, vol. 5, no. 2, pp. 9–19, Apr. 2017. [Online]. Available: https://www.journals.aiac.org.au/index.php/IJELS/article/view/3400

[106] D. Amaratunga, D. Baldry, M. Sarshar, and R. Newton, “Quantitative and qualitative research in the built environment: application of “mixed” research approach,” *Work study*, vol. 51, no. 1, pp. 17–31, Feb. 2002. [Online]. Available: https://doi.org/10.1108/00438020210415488

[107] C. Hughes, “Qualitative and quantitative approaches,” 2012. [Online]. Available: https://www.scribd.com/document/135552364/Quantitative-and-Qualitative-Approaches

[108] R. B. Burns, “Introduction to research methods,” Frenchs Forest, NSW, 2000.

[109] S. L. McGregor and J. A. Murnane, “Paradigm, methodology and method: Intellectual integrity in consumer scholarship,” *International journal of consumer studies*, vol. 34, no. 4, pp. 419–427, Jun. 2010. [Online]. Available: https://doi.org/10.1111/j.1470-6431.2010.00883.x

[110] M. Bamba, *Development and Application of Novel Computational Intelligence Techniques to the Multivariate Analysis of Metabolomics Biofluids Datasets*. De Montfort University, Dec. 2017. [Online]. Available: https://books.google.co.uk/books?id=DQExwAEACAAJ

[111] D. Wahyuni, “The research design maze: Understanding paradigms, cases, methods and methodologies,” *Journal of applied management accounting research*, vol. 10, no. 1, pp. 69–80, Jun. 2012. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2103082

[112] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla, “Markov logic,” in *Probabilistic inductive logic programming*. Springer, Berlin, Heidelberg, 2008, pp. 92–117. [Online]. Available: https://doi.org/10.1007/978-3-540-78652-8_4

[113] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic *et al.*, “Developing language processing components with gate version 7 (a user guide),” Feb. 2012. [Online]. Available: https://gate.ac.uk/releases/gate-7.0-build4195-ALL/doc/

[114] K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, and H. Cunningham, “Shallow methods for named entity coreference resolution,” in *Chaınes de références et résolveurs d’anaphores, workshop TALN*, Jan. 2002, pp. 24–27.

[115] H. Cunningham, “GATE, a general architecture for text engineering,” *Computers and the Humanities*, vol. 36, no. 2, pp. 223–254, May 2002. [Online]. Available: https://doi.org/10.1023/A:1014348124664

[116] W. M. Soon, H. T. Ng, and D. C. Y. Lim, “A machine learning approach to coreference resolution of noun phrases,” *Computational Linguistics*, vol. 27, no. 4, pp. 521–544, Dec. 2001. [Online]. Available: https://aclanthology.org/J01-4004

[117] M. A. Yosef, J. Hoffart, I. Bordino, M. Spaniol, and G. Weikum, “AIDA: An online tool for accurate disambiguation of named entities in text and tables,” *Proceedings of the VLDB Endowment*, vol. 4, no. 12, pp. 1450–1453, Aug. 2011. [Online]. Available: https://doi.org/10.14778/3402755.3402793

[118] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic *et al.*, “Developing language processing components with gate version 6 (a user guide),” Nov. 2010. [Online]. Available: https://gate.ac.uk/releases/gate-6.0-build3764-ALL/doc/tao/tao.pdf

[119] D. Thakker, T. Osman, and P. Lakin, “GATE JAPE grammar tutorial,” *Nottingham Trent University, UK, Phil Lakin, UK, Version*, vol. 1, Feb. 2009.

[120] H. Cunningham, D. Maynard, and V. Tablan, “Jape: a java annotation patterns engine,” 1999.

[121] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, “Robust disambiguation of named entities in text,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 782–792. [Online]. Available: https://aclanthology.org/D11-1072

[122] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell, “Toward an architecture for never-ending language learning,” in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI'10, vol. 5, no. 3. AAAI Press, Jul. 2010, pp. 1306–1313. [Online]. Available: https://dl.acm.org/doi/10.5555/2898607.2898816

[123] G. Lucko and E. M. Rojas, “Research validation: Challenges and opportunities in the construction domain,” *Journal of construction engineering and management*, vol. 136, no. 1, pp. 127–135, Jan. 2010. [Online]. Available: https://doi.org/10.1061/(ASCE)CO.1943-7862.0000025

[124] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel *et al.*, “Performance measures for information extraction,” in *In Proceedings of DARPA broadcast news workshop*. Herndon, VA, Aug. 2000, pp. 249–252.

[125] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, “Web data extraction, applications and techniques: A survey,” *Knowledge-based systems*, vol. 70, no. C, pp. 301–323, Nov. 2014. [Online]. Available: https://doi.org/10.1016/j.knosys.2014.07.007

[126] J. Cowie and W. Lehnert, “Information extraction,” *Communications of the ACM*, vol. 39, no. 1, pp. 80–91, Jan. 1996. [Online]. Available: https://doi.org/10.1145/234173.234209

[127] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "A brief survey of text mining: Classification, clustering and extraction techniques," *arXiv preprint arXiv:1707.02919*, Jul. 2017.

[128] D. De Vaus, *Research design in social research*, 1st ed. SAGE, Feb. 2001.

[129] B. v. Wyk, "Research design and methods: Part 1," *Post graduate enrolment and throughput*, 2015.

[130] Wikipedia contributors, "JAPE (linguistics) — Wikipedia, the free encyclopedia," 2017, [Online; accessed 22-January-2021]. [Online]. Available: https://en.wikipedia.org/wiki/JAPE_(linguistics)

[131] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI'10, vol. 5, no. 3. AAAI Press, Jul. 2010, pp. 1306–1313. [Online]. Available: https://dl.acm.org/doi/10.5555/2898607.2898816

[132] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, Oct. 2020. [Online]. Available: https://doi.org/10.1155/2020/8885861

[133] Y. Tsfati, H. Boomgaarden, J. Strömbäck, R. Vliegenthart, A. Damstra, and E. Lindgren, "Causes and consequences of mainstream media dissemination of fake news: literature review and synthesis," *Annals of the International Communication Association*, vol. 44, no. 2, pp. 157–173, May 2020. [Online]. Available: https://doi.org/10.1080/23808985.2020.1759443

[134] D. al Shekaili, *Integrating Linked Data Search Results Using Statistical Relational Learning Approaches*. The University of Manchester (United Kingdom), Aug. 2017. [Online]. Available: https://www.research.manchester.ac.uk/portal/en/theses

[135] S. Magliacane, P. Stutz, P. Groth, and A. Bernstein, "foxpsl: A fast, optimized and extended psl implementation," *International Journal of Approximate Reasoning*, vol. 67, pp. 111–121, Jun. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888613X15000845

[136] E. Augustine, T. Rekatsinas, and L. Getoor, "Tractable probabilistic reasoning through effective grounding," in *Third ICML workshop on Tractable Probabilistic Modeling*, Jan. 2019. [Online]. Available: https://par.nsf.gov/biblio/10110919

[137] F. Niu, C. Ré, A. Doan, and J. Shavlik, "Tuffy: Scaling up statistical inference in markov logic networks using an rdbms," *Proceedings of the VLDB Endowment*, vol. 4, no. 6, p. 373–384, Mar. 2011. [Online]. Available: https://doi.org/10.14778/1978665.1978669

[138] V. Embar, D. Sridhar, G. Farnadi, and L. Getoor, "Scalable structure learning for probabilistic soft logic," *arXiv preprint arXiv:1807.00973*, Jul. 2018.

[139] N. Jeyaraj. (2018, Oct.) A high level overview of the probabilistic soft logic. [Online]. Available: https://medium.com/datadriveninvestor/a-high-level-overview-of-the-probabilistic-soft-logic-fbafad146b69

[140] E. Augustine and L. Getoor, "A comparison of bottom-up approaches to grounding for templated markov random fields," 2018. [Online]. Available: https://mlsys.org/Conferences/doc/2018/87.pdf

[141] B. Godefroy and C. Potts, "Modeling drug-disease relations with linguistic and knowledge graph constraints," *arXiv preprint arXiv:1904.00313*, Mar. 2019.

[142] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011. [Online]. Available: https://doi.org/10.1561/2200000016

[143] I. Sabek, "Adopting markov logic networks for big spatial data and applications," Ph.D. dissertation, University of Minnesota, Jan. 2020. [Online]. Available: https://hdl.handle.net/11299/213095

[144] S. Natarajan, V. Bangera, T. Khot, J. Picado, A. Wazalwar, V. S. Costa, D. Page, and M. Caldwell, "Markov logic networks for adverse drug event extraction from text," *Knowledge and information systems*, vol. 51, no. 2, pp. 435–457, May 2017.

[145] P. Singla and P. Domingos, "Entity resolution with markov logic," in *Sixth International Conference on Data Mining (ICDM'06)*, IEEE. IEEE, 2006, pp. 572–582.

[146] D. Lowd and P. Domingos, "Efficient weight learning for markov logic networks," in *European conference on principles of data mining and knowledge discovery*, vol. 4702. Springer, 09 2007, pp. 200–211.

[147] D. D. Lewis, "Evaluating text categorization i," in *Proceedings of the Workshop on Speech and Natural Language*, ser. HLT '91. USA: Association for Computational Linguistics, Feb. 1991, p. 312–318. [Online]. Available: https://doi.org/10.3115/112405.112471

[148] D. Lewis, "Representation quality in text classification: An introduction and experiment," in *Speech and Natural Language: Proceedings of a Workshop Held*

at Hidden Valley, Pennsylvania, June 24-27, 1990, 01 1990. [Online]. Available: https://aclanthology.org/H90-1057

[149] S. Sudhahar, G. De Fazio, R. Franzosi, and N. Cristianini, "Network analysis of narrative content in large corpora," *Natural Language Engineering*, vol. 21, no. 1, pp. 81–112, 2015.

[150] I. Flaounas, T. Lansdall-Welfare, P. Antonakaki, and N. Cristianini, "The anatomy of a modular system for media content analysis," *arXiv preprint arXiv:1402.6208*, 2014.

[151] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[152] D. Cartwright and F. Harary, "Structural balance: a generalization of heider's theory." *Psychological review*, vol. 63, no. 5, p. 277, 1956.

[153] R. Dale, "Nlp in a post-truth world," *Natural Language Engineering*, vol. 23, no. 2, pp. 319–324, Mar. 2017.

[154] K. Kertysova, "Artificial intelligence and disinformation: How ai changes the way disinformation is produced, disseminated, and can be countered," *Security and Human Rights*, vol. 29, no. 1-4, pp. 55–81, Dec. 2018.

[155] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, p. 2, 2017. [Online]. Available: https://www.documentcloud.org/documents/5794867-National-Security-Archive-David-Gunning-DARPA

[156] N. Ahmadi, J. Lee, P. Papotti, and M. Saeed, "Explainable fact checking with probabilistic answer set programming," *arXiv preprint arXiv:1906.09198*, Jun. 2019.

[157] K. Sokol and P. Flach, "Explainability fact sheets: a framework for systematic assessment of explainable approaches," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM, Jan. 2020, pp. 56–67. [Online]. Available: http://dx.doi.org/10.1145/3351095.3372870

[158] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *2016 IEEE symposium on security and privacy (SP)*, IEEE. IEEE, May 2016, pp. 598–617.

[159] J. E. Mercado, M. A. Rupp, J. Y. Chen, M. J. Barnes, D. Barber, and K. Procci, "Intelligent agent transparency in human–agent teaming for multi-uxv management," *Human factors*, vol. 58, no. 3, pp. 401–415, May 2016. [Online]. Available: https://doi.org/10.1177/0018720815621206

[160] W. R. Swartout and J. D. Moore, "Explanation in second generation expert systems," in *Second generation expert systems*, J.-M. David, J.-P. Krivine, and R. Simmons, Eds. Springer Berlin Heidelberg, 1993, pp. 543–585. [Online]. Available: https://doi.org/10.1007/978-3-642-77927-5_24

[161] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *The Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, Sept. 2015. [Online]. Available: http://dx.doi.org/10.1214/15-AOAS848

[162] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[163] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Generating and understanding personalized explanations in hybrid recommender systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 10, no. 4, pp. 1–40, Nov. 2020. [Online]. Available: https://doi.org/10.1145/3365843