

Open Information Extraction for Financial Knowledge Graphs

Sahil Shah

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Data Science and Machine Learning
of
University College London.

Department of Computer Science
University College London

September 12, 2021

Declaration

I, Sahil Shah, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis investigates whether transformer models, such as FinBERT and RoBERTa can be used in the task of Open Information Extraction to improve the quality of financial Knowledge Graphs, working in a multilingual capacity to read articles in both English and German. This project was conducted in collaboration with Deutsche Bank and can potentially be used to obtain information on how events in the news could impact on certain interests from a risk management perspective. Since Deutsche Bank makes press releases in both English and German, this model would provide the ability to build Knowledge Graphs from articles in both languages for users to observe how certain events may affect other interests both directly and indirectly.

Through improving on certain metrics such as precision and recall on a financial dataset by using more finely-tuned transformer based models such as FinBERT and RoBERTa, the quality of the financial Knowledge Graphs will be improved. The visualisation of the Knowledge Graph will provide additional insight as to the relevant topics concerned with a particular event in the news.

The thesis contains the following four sections:

1. **Data and Model Design.** The first section explains how the BERT model can be used to generate feature embeddings and extract predicates and arguments. The triples produced from Open Information Extraction can then be passed into Knowledge Graphs, upon which an algorithm will be used to find related

topics.

2. **Model Implementation.** The second part demonstrates how the model is implemented, first in English and then in German using the multilingual BERT.
3. **Training and Testing.** The third section describes how the models were trained and then tested, including the financial news that was used to create the Knowledge Graph and extract insights from the closest entities.
4. **Results and Discussion.** The final part describes and explains the results obtained from the multilingual Open Information Extraction task, as well as the results of the example articles and events when being visualised in the Knowledge Graph.

The thesis also presents the following contributions to science:

1. Research into which transformer-based model performs best in a financial news context in the overall task of Open Information Extraction
2. Research in the smaller subtasks of Named Entity Recognition and Relation Extraction, as to which transformer-based model performs best in a financial news context
3. The establishment of a financial **multilingual** open information extraction model
4. The use of visualisation of financial knowledge graphs to obtain relevant topics concerned with an topical event

Impact Statement

Risk Management is a hugely important area of finance, seeking to predict and limit potential losses made in the future. It is becoming of increasing significance, due to certain catastrophes such as the COVID-19 pandemic and an increasing number of natural disasters brought upon by climate change, to list but a few instances where risk management is of importance. Knowledge Graphs have known to be highly effective in understanding complex relations between entities seemingly not intuitively unrelated to each other and particularly their application in Risk Management will be of great use. Fundamental to the quality of Knowledge Graphs themselves, are the information extraction methods used to extract triplets from text in order to build Knowledge Graphs. Therefore, by improving the accuracy and versatility of these Triplet Extraction methods, one can hope to improve the quality of Knowledge Graphs. This thesis specifically looks at which transformer-based models can best extract triplets from text and whether multilingual transformer-based models can be used to extract triplets from text in other languages and which performs best.

Acknowledgements

I would like to thank my supervisors at UCL and Deutsche Bank, Prof. Philip Treleaven and John Barclay for all the advice and guidance they have provided during the research project, in terms of technical experiments and thesis writing.

I would also like to thank Dr. Pasquale Minervini of the Natural Language Processing group at UCL for the help and advice he has provided throughout the project.

Finally, I would like to thank my family for all the support they have provided throughout the course of this project.

Contents

1	Introduction	12
1.1	Research Motivation	13
1.2	Research Objectives	15
1.3	Research Experiments	16
1.4	Scientific Contributions	17
1.5	Thesis Structure	18
2	Background and Literature Review	19
2.1	Application Domain	20
2.1.1	Risk, Finance and Treasury (RFT) Technology	20
2.1.2	Knowledge Graphs	21
2.1.3	Financial Knowledge Graphs	22
2.2	Method Development	23
2.2.1	Natural Language Processing	23
2.2.2	Transformers	27
2.2.3	Information Extraction	29
2.2.4	Named Entity Recognition	31
2.2.5	Relation Extraction	32
2.2.6	Open Information Extraction	33

3	Data and Model Design	36
3.1	Introduction	36
3.2	Data	37
3.3	Model Design	39
3.4	Summary	41
4	Model Implementation	42
4.1	Introduction	42
4.2	Implementation	43
4.3	Summary	45
5	Training & Testing	46
5.1	Introduction	46
5.2	Training & Testing	47
5.3	Summary	49
6	Results and Discussion	50
6.1	Introduction	50
6.2	Information Extraction	51
6.3	Multilingual Information Extraction	56
6.4	Knowledge Graphs	57
6.5	Summary	59
7	Conclusions and Future Work	61
7.1	Conclusions	61
7.2	Future Work	63
	References	65

List of Figures

1.1	Flowchart of Typical Triplet Extraction procedure	16
2.1	Knowledge Graph Example [1]	21
2.2	Multi-Layer Perceptron [2]	24
2.3	Recurrent Neural Network Block [3]	25
2.4	Vanilla LSTM Block [4]	26
2.5	GRU Block [5]	26
2.6	Transformer Block [6]	27
2.7	BERT Block [7]	28
2.8	Triplet Structure from Factual Sentence [8]	29
2.9	BIO Tagging Procedure Example [9]	32
2.10	Relation Extraction Example [10]	33
3.1	Flowchart displaying training procedure	39
3.2	Flowchart displaying Triplet Extraction	39
3.3	MultiOIE Triplet Extraction Structure [11]	40
6.1	Named Entity Recognition Triplet Example 1	52
6.2	Named Entity Recognition Triplet Example 2	53
6.3	Named Entity Recognition Triplet Example 3	53
6.4	Named Entity Recognition Triplet Example 4	53

6.5	Named Entity Recognition Triplet Example 5	53
6.6	Named Entity Recognition Triplet Example 6	54
6.7	Named Entity Recognition Triplet Example 7	54
6.8	Named Entity Recognition Triplet Example 8	54
6.9	Named Entity Recognition Triplet Example 9	54
6.10	Named Entity Recognition Triplet Example 10	55
6.11	Apple Knowledge Graph	57
6.12	Deutsche Bank Knowledge Graph	58

List of Tables

2.1	BERT vs GPT-3 comparison [12]	28
4.1	<i>NLTK</i> vs <i>spaCy</i> Comparison [13]	43
6.1	English Model Performance Metrics	51
6.2	Examples Evaluation	55
6.3	Multilingual Model Performance Metrics	56

Chapter 1

Introduction

The aim of this chapter is to provide an introduction to this thesis by explaining the motivation behind the research, the aims, experiments, contributions to science and the overall structure of the thesis. The chapter begins by describing the motivation behind Information Extraction and Knowledge Graphs, as well as their many applications across industries and academia. The chapter then describes the aims, experiments, scientific contributions and thesis structure.

This thesis investigates the effect of transfer learning of different BERT models in the task of Open Information Extraction, in order to create triplets which can be passed into a Knowledge Graph. Upon this, a visualisation is used to find relevant topics affected by a certain event or change to a node in a graph.

This research was carried out in collaboration with Deutsche Bank, under the supervision of John Barclay, who gave expert advice and confirmed the demand for this research within the industry.

1.1 Research Motivation

Information Extraction is a specific task which looks to extract structured information from unstructured or semi-structured machine-readable documents within the field of Natural Language Processing [14]. Natural Language Processing is a field of Linguistics and Computer Science, which enables computers to interpret large amounts of text. Other popular Natural Language Processing tasks would be Sentiment Analysis, Topic Modelling, Searching, Machine Translation, Summarisation, Parts-of-Speech Tagging and Question Answering to name but a few. Information Extraction specifically is used for the automation of certain tasks and data-driven activities like finding patterns, trends and hidden relationships [14].

Examples of information extraction tasks are named entity recognition, where models aim to identify entities within sentences of text and relation extraction, where given two entities in a sentence, models aim to extract the relation between them. For example, the relation linking entities Paris and France would be "is the capital of". The combination of both of these tasks is called Open Information Extraction, where given a sentence, the model will find two entities and the relation between them, called a triplet [15]. Recent activities in multimedia document processing like automatic annotation and content extraction out of images, audio, video or documents could be seen as information extraction tasks. Due to the difficulty of the problem, current approaches to Information Extraction focus on narrowly restricted domains.

The creation of triplets including named entities and the relation between them is particularly useful in the case of Knowledge Graphs. Knowledge Graphs are graphical representations of entities as nodes, where the links between them represent the relations. Knowledge Graphs were first created almost 10 years ago by Google [16] and have been extremely useful in understanding complex relationships between entities along many edges in the graph for entities that were previously unknown to be connected.

Through making improvements on certain metrics such as precision and recall on financial news datasets produced in the task of information extraction, it is therefore possible to improve the quality of financial Knowledge Graphs. Therefore, by using transfer learning in certain pretrained transformer models such as BERT and then fine-tuning on the financial news dataset, I can hope to improve these existing metrics. Different transformer-based models such as BERT, FinBERT, RoBERTa and DistilBERT were used to investigate as to which model could produce the highest performance. Another feature to make the model more widely usable across articles in different languages would be to create additional multilingual capabilities. This capability would be important for Deutsche Bank, since the bank makes press releases in both English and German and so this would avoid the need to create separate Information Extraction models and Knowledge Graphs in each language and instead combine them into one model. Furthermore, news from around the world is often published in the native language around the country and a multilingual model would be able to interpret this as fast as possible instead of having to wait for an English translation to be released.

Following from this, while it is important that the Knowledge Graph produced from the dataset is of the highest quality, it is also imperative that it is easily interpretable. When a brand new article is passed through the Knowledge Graph, it is important to be able to easily recognise which entities will be affected by this particular event or change to an entity within the first few generations of the graph. Therefore, using test dataset specific to a company to produce triplets for the graph, will display how an event in the news might affect a specific client or interest of Deutsche Bank both directly and indirectly.

Knowledge Graphs can be put to many different uses within the financial domain. For example, Knowledge Graphs have been used by traders to predict how certain events in the news might affect share prices, in order to capitalise on market inefficiencies. More specifically, though within Risk Management, risk indicators from text analytics can be used for different asset and risk management decisions and to

identify specific risk types [17]. Given that textual data can be found almost everywhere, extracting meaningful insights from text remains a difficult proposition. However, Knowledge Graphs can be used to determine the risk associated with certain events and which interests it may affect from a bank's perspective, both directly and indirectly without knowing. Examples of events which may be of concern from a risk management perspective may be international crises, natural disasters and humanitarian efforts to name but a few. Therefore, Knowledge Graphs produced in this way can be used beneficially in other areas and industries.

1.2 Research Objectives

The purpose of this research thesis is to investigate whether information extraction pipelines can be expanded to be multilingual using spaCy and whether the metrics of precision and recall can be improved upon by using the FinBERT or RoBERTa transformer models.

The first objective was to create a prototype model, trained using open-source Kaggle datasets. This consisted of using spaCy transformer models to create an Information Extraction pipeline, consisting of a Named Entity Recognition model, followed by a Relation Extraction model, both trained independently of each other.

The second objective was to create an Information Extraction pipeline for a more financial purpose, through training on an internal Deutsche Bank dataset, using a triplet extraction structure, to discover which pretrained transformer model performed best in a financial context. Similarly, the third objective was to create a Multilingual Information Extraction pipeline, using a similar internal Deutsche Bank dataset in a different language, to ascertain which multilingual pretrained transformer model performed best in a financial context. Figure 1.1 displays a flowchart of a typical Information Extraction pipeline as previously described.

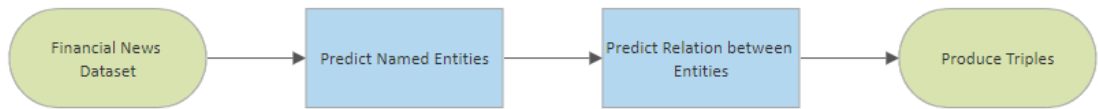


Figure 1.1: Flowchart of Typical Triplet Extraction procedure

The final objective would be to create high quality Knowledge Graphs has been produced from triplets from using the best Information Extraction model on the financial news dataset provided by Deutsche Bank. Finally a graphical structure visualisation would be used to find how certain events might impact on a specific company.

1.3 Research Experiments

The thesis begins with the construction of a prototype joint entity-relation extraction pipeline, consisting of a named entity recognition model, followed by a relation extraction model to create an overall information extraction model. This was constructed using two open-source datasets from Kaggle designed for each task, with each model trained individual.

Using the internal Deutsche Bank dataset which consisted of financial news articles provided by Refinitiv, a new information extraction model, which combined both tasks, was then trained and certain metrics such as precision, recall, F1 score and accuracy were calculated and compared across various different transformer-based models such as BERT, FinBERT, RoBERTa and DistilBERT.

The next part of the information extraction experiment included training multilingual transformer-based models on the same financial news dataset, this time with the capacity to find triplets from articles in other languages, in particular German and would prove to be very useful for Deutsche Bank’s German press releases. The metrics generated by these models using the English financial news dataset were then compared to identify the best information extraction model amongst all previ-

ously mentioned models and Multilingual BERT and XLM-RoBERTa.

Finally, the best information extraction model, RoBERTa, was taken and used on test financial news datasets from Refinitiv, similarly provided by Deutsche Bank in order to create Knowledge Graphs. These were then visualised and analysed to observe how certain events might directly and indirectly affect specific companies.

1.4 Scientific Contributions

This project contributes to the existing research in the following ways:

1. Research into which transformer-based model performs best in a financial context on a financial news dataset in the task of Open Information Extraction
2. Research in the smaller subtasks of Named Entity Recognition and Relation Extraction, as to which transformer-based model performs best in a financial news context
3. Research into whether a multilingual information extraction model can be created using transformers and which type of transformer yields the best metrics on a financial news dataset
4. The visualisation of financial knowledge graphs to obtain relevant topics concerned both directly and indirectly with an event, to learn potential new relations that were not known to have existed
5. Improve Risk Management strategies for the Chief Risk Office (CRO) through the development of a Knowledge Graph to better understand complex relations between entities with large separations
6. Improved strategies would lead to a reduction in losses due to risk optimisation, due to cheap implementation and high scalability with the quality of insights increasing with data volume

1.5 Thesis Structure

The structure of the thesis is as follows:

- Chapter 2 - **Background and Literature Review**. This chapter provides an introduction to the application domain, the gap in Risk Management, Knowledge Graphs in general and how they can be used in Financial Risk Management. The second part of this chapter then describes how Knowledge Graphs are generated using Information Extraction methods and how transformer models are used.
- Chapter 3 - **Data and Model Design**. This chapter first provides a description of the datasets and their origins, followed by an overview of how the model was designed in terms of extracting entities and relations, before being finally used to create a Knowledge Graph.
- Chapter 4 - **Model Implementation**. This chapter describes how this Information Extraction pipeline and Knowledge Graph were implemented, including the specific software and packages used.
- Chapter 5 - **Training and Testing**. This chapter contains information on the procedures used for training and testing, as well as the metrics used for comparing the performance of the different models during testing.
- Chapter 6 - **Results and Discussion**. This chapter looks at the results obtained from the training and testing of the Information Extraction pipeline and attempts to rationalise the metrics obtained using examples.
- Chapter 7 - **Conclusions and Future Work**. The final chapter looks to provide an overall conclusion for the project and summarise all the main findings of the experiments. It finishes with recommended future work which can be carried out, both in terms of the models used for Information Extraction and in terms of more easily interpretable UIs for Knowledge Graphs.

Chapter 2

Background and Literature Review

This chapter presents background information on the main concepts of the fields in which this project is based on. The first part of this chapter looks at the application domain as a whole, in terms of Risk Management and the application of Knowledge Graphs to this area of finance. The second section of this chapter provides background on the key concepts and current state-of-the-art of Information Extraction methods using transformer-based models.

The experiment carried out in this project, first attempts to build a Multilingual Information Extraction pipeline to produce entity-relation triplets, before building Knowledge Graphs for Risk Management purposes with these triplets. Therefore, this chapter provides a high-level overview of the background knowledge including key concepts, and the current state-of-the-art in Information Extraction tasks and Knowledge Graph creation within Natural Language Processing. Each following chapter will therefore focus on some areas within the background and literature which are relevant for that particular topic.

The first part of the background chapter focuses on the application domain of the research, Risk Management more broadly, before highlighting the powerful use case of Knowledge Graphs in this field, to address a specific, as yet unmet, need. The

second part of the background and literature review chapter focuses on method development. This specifically concerns how Natural Language Processing can be used to interpret text and how transformer models improved on the previously used recurrent neural networks. Finally, the chapter examines how different types of models can be used in information extraction tasks and how the quality of these methods in this area has improved significantly in the last few years.

2.1 Application Domain

The first section describes the application domain of the project, first explaining why the application of technology within Risk Management is important. Then the origin of Knowledge Graphs and their usefulness is demonstrated, as well their specific use case within Risk Management.

2.1.1 Risk, Finance and Treasury (RFT) Technology

Within banking, Risk, Finance and Treasury (RFT) Technology is responsible for delivering and maintaining front to back technology solutions to the Risk, Finance and Treasury departments and their safe and controlled implementation and management into the operational business flows. The Risk Office is concerned with efficient and effective governance of significant risks, and related opportunities, aiming to reduce and minimise the risk of generating losses. The Finance Office addresses issues such as tracking cash flow and financial planning, in terms of analysing financial strengths and weaknesses and providing corrective actions. The Treasury Office looks after money management, ensuring the business has the money needed to manage day-to-day business obligations and that there is enough cash available to meet future liabilities. In order to do this successfully, the function analyses the risks that might impact the flow of cash.

For example, a natural disaster, such as an earthquake near a coast could have many different impacts financially. It would cause a tsunami, destroying housing by the

coast and shifting the property market in that area for several years, where a financial institution may have interests. It may also break an oil or gas pipeline in the sea near the coast, again shifting the oil and gas market in that area. The area may not be able to host tourists in the immediate future, meaning that the tourism industry would be heavily affected. All of these direct impacts could have financial repercussions to those with investment interests in those areas [18].

There may also be hidden indirect consequences as a result, if the housing market is impacted, companies may prefer to build offices elsewhere, increasing unemployment in the area. Such indirect correlations can be difficult to predict, since these relationships may be unknown or difficult to interpret and may have financial impacts which were not foreseen. However, these previously unknown or difficult to interpret relationships can be discovered using a Knowledge Graph.

2.1.2 Knowledge Graphs

A Knowledge Graph is a directed acyclic graph (DAG) containing entities at the nodes and relations along the edges, thereby displaying a network of relationships between known entities. An example of this can be seen in Figure 2.1, which describes a network of entities and the relations between them [19].

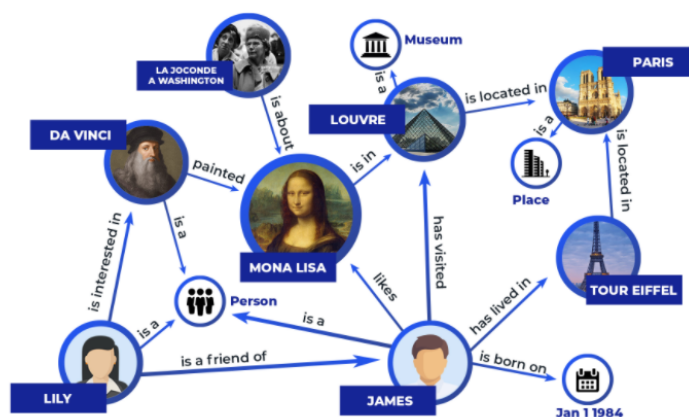


Figure 2.1: Knowledge Graph Example [1]

Knowledge Graphs were first developed by Google almost 10 years ago and have

enabled text to not only be interpreted as strings, but as known entities [20]. The original Knowledge Graph developed by Google was designed for Question Answering purposes and since then much research has been carried out in Knowledge Graphs and especially in the Question Answering problem field. For example, very recently, the approach used in *Yasunaga et al., 2021* [21] aimed to implement concepts such as relevance scoring (some entity nodes being more relevant given the Question Answering context) and joint reasoning (where the Question Answering context is viewed as an explicit node).

Language models can also be seen as Open Knowledge Graphs according to *Wang et al., 2020* [22], since pretrained language models such as BERT and GPT-2/3 have automatically acquired knowledge from large-scale corpora, enabling them to improve downstream tasks. This article shows that Knowledge Graphs constructed with one forward pass of the pretrained language models can outperform standard ones created by humans. This highlights the importance of transformer-based models within this project to obtain high quality Knowledge Graphs.

A particularly interesting application in of Knowledge Graphs was found in *Liu et al., 2020* [23], where language representation models were knowledge-enabled with triples injected into sentences as domain knowledge. This new model, named K-BERT was able to significantly outperform the baseline transformer model BERT, showing great promise in domain-specific tasks such as finance. Therefore, this seems a promising approach to creating high quality Knowledge Graphs and would be worth exploring and investigating in this thesis.

2.1.3 Financial Knowledge Graphs

Knowledge Graphs have been shown to be particularly useful in industries, with many different applications. Within finance, Knowledge Graphs have been used for various different purposes such as stock market prediction and within audit mechanisms. Another very popular use of Knowledge Graphs within finance, lies in the area of fraud management [24]. However, as previously discussed, there is a strong

demand for Knowledge Graphs to be used in Risk Management, since they can be applied in a more versatile manner in this field, due to the many different outcomes in the real world which can impact on interests and investments.

A previous thesis in the creation of Knowledge Graphs, *Elhammadi, 2020* [25] carried out Information Extraction in a pipeline using bidirectional LSTM models, a type of recurrent neural network. This thesis aims to improve on the quality of that Information Extraction pipeline, through the use of transformer-based models, using an attention mechanism to progress on certain metrics such as precision, recall and F1 scores.

In order to obtain meaningful insights from Knowledge Graphs, it is useful to have sorting algorithms to identify the most relevant relations and entities to the entity in question. For example, from a financial standpoint, if there were to be a natural disaster, it would be of importance to be able to sort the Knowledge Graph according to the disaster entity and its most relevant entities and relations, to understand how an event in the news may affect potentially closely linked business interests. This algorithm is called topological sorting and was implemented in *Ajwani et al., 2011* [26] for large scale graphs such as Knowledge Graphs, which are directed acyclic graphs, allowing for useful sorting of the graph by topology.

2.2 Method Development

The second section describes how methods have progressed since the origins of Natural Language Processing and the use of transformer models for tasks. It then goes into more detail about Information Extraction and the different tasks involved in extracting triplets.

2.2.1 Natural Language Processing

Natural Language Processing is a field of machine learning which enables machines to interpret large amounts of text. The basic premise is that words or 'tokens' can

be represented by points in high-dimensional space. As an example, word2vec represents words as 300-dimensional vectors [27]. Therefore, vectors which take one from one point to another can sometimes correspond to a relationship between the tokens. For example, the vector that takes one from the point 'Paris' to the point 'France', would also take one from the point 'Rome' to the point 'Italy', with the vector representing the relation 'is the capital city of'.

Neural Networks can be represented graphically and consist of node layers - an input layer, one or more hidden layers and an output layer. Their name and structure are inspired by the human brain, imitating the way biological neurons signal to each other. Neural Networks are able to make nonlinear decision boundaries linearly separable through transforming the input space using multiple nonlinearities, where each node represents a linear combination of weights on the inputs in addition to a bias, before being acted on by an activation function. This is particularly useful in the case of Natural Language Processing, where words and sentences are given high-dimensional representations.

In the basic case it is possible to pass independent vector representations of tokens through linear layers in a multi-layer perceptron to attempt to detect named entities, but this fails in relation extraction as there is no context given for the relation. An example of a multi-layer perceptron is shown in Figure 2.2.

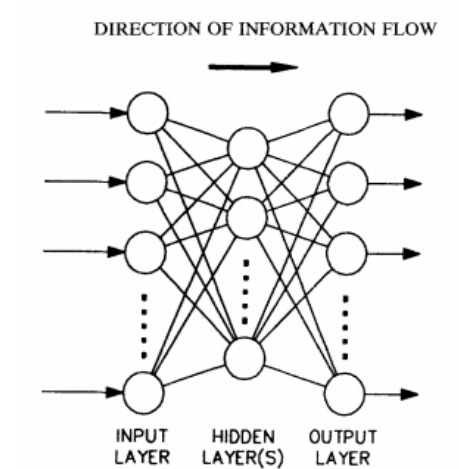


Figure 2.2: Multi-Layer Perceptron [2]

Therefore, it makes sense in Natural Language Processing to use a Recurrent Neural Network which builds an encoded representation of the sentence by passing each token through in a sequential manner. This can then be used as part of a language model, which is then able to decode the encoded representation into text. In the case of Named Entity Recognition, it will produce the BIO tags for each token and for Relation Extraction, the relation between the two entities will be produced. An example of a recurrent neural network block is shown in Figure 2.3, displaying how the hidden state combine with input state to create an output and new hidden states.

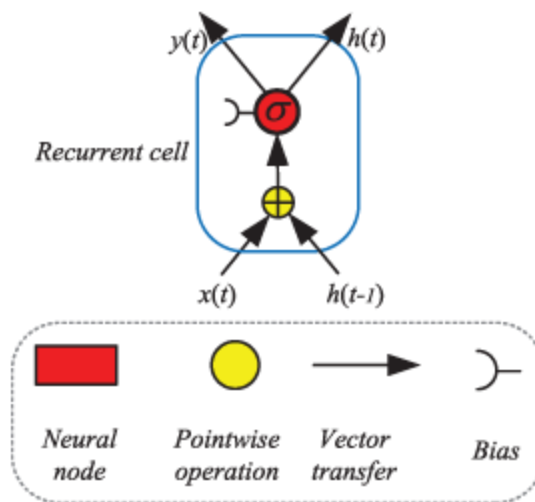


Figure 2.3: Recurrent Neural Network Block [3]

Recurrent Neural Networks however, run into problems where the first tokens passed through the network sequentially, are not well represented in the encoded representation. This is due to the vanishing gradient problem and was originally solved through implementing a cell state alongside the hidden state to ensure that each token within the sentence was well represented. This was called a Long Short-Term Memory (LSTM) model. Figure 2.4 displays the LSTM block, where it contains an additional cell state to maintain long-term memory.

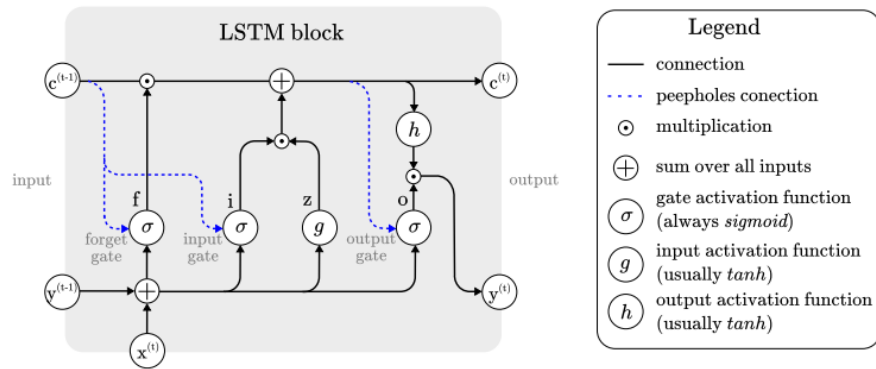


Figure 2.4: Vanilla LSTM Block [4]

Another similar memory retention model called a Gated Recurrent Unit (GRU) was also developed, which contained two values at output (output and hidden states), as opposed to the three values at output (output, hidden and cell states) in the LSTM. Despite fewer output values required to be saved and being faster to execute, LSTM models are more accurate on datasets with long sequences and so are therefore more favoured in industry. Figure 2.5 displays the GRU block, where it contains only two outputs in the output and hidden states.

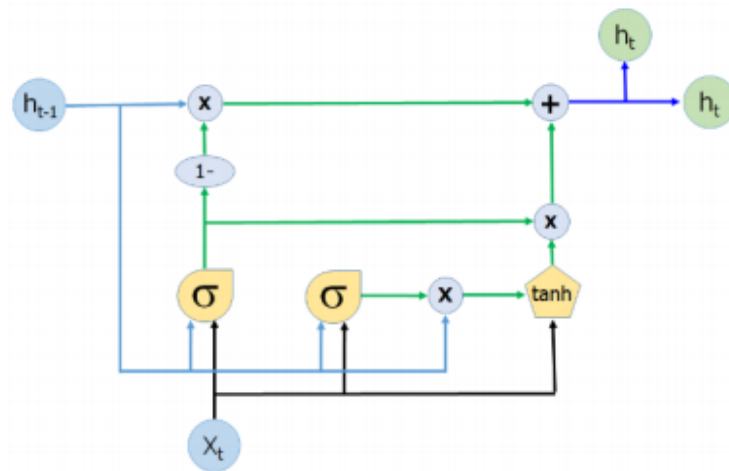


Figure 2.5: GRU Block [5]

2.2.2 Transformers

The field of Transformers then built on this idea, by introducing the concept of attention. Where recurrent neural networks process the data in order, transformers do not. The attention head mechanism provides context for any token in the input sequence and therefore being able to compute the context for the meaning of each token in the sequence. This then reduces training time and therefore computational cost, through more parallelisation than in recurrent neural networks. Transformer models are chosen for Natural Language Processing tasks because of this additional parallelisation. Figure 2.6 shows the structure of a block of a transformer, containing the attention head structure.

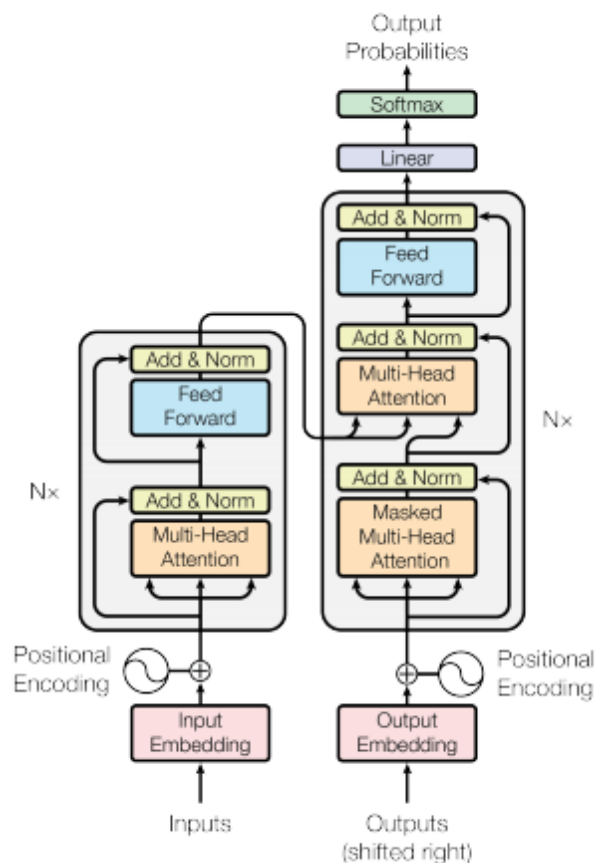


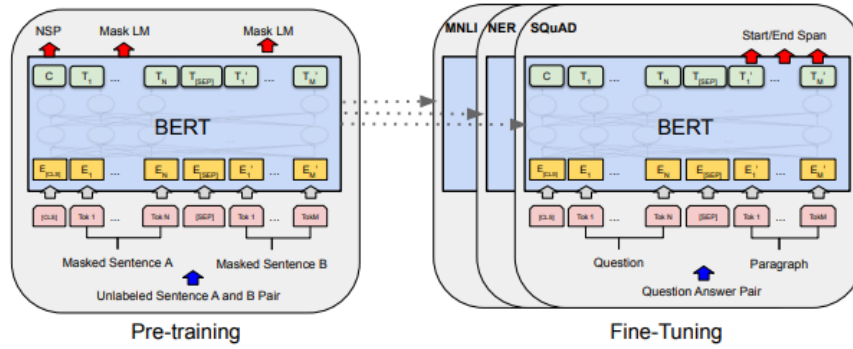
Figure 2.6: Transformer Block [6]

This then led to the development of pretrained models like BERT (Bidirectional En-

	BERT	GPT-3
No. Parameters	110 million	175 billion
Nature	Bidirectional	Autoregressive
Applications	Documents, etc.	News, articles, codes etc.

Table 2.1: BERT vs GPT-3 comparison [12]

coder Representations from Transformers) and GPT (Generative Pretrained Transformer), which can be fine-tuned for specific purposes and use a form of transfer learning to further improve their accuracy.

**Figure 2.7:** BERT Block [7]

However, as can be seen in Table 2.1, BERT contains a far fewer number of parameters and is therefore able to occupy less space and can be trained with less computational complexity.

There are different variants of the BERT model, which differ in terms of their training time, number of parameters and data occupied. For example, RoBERTa is a variant of BERT, with the same number of parameters, but a much longer training time which 4-5 times that of BERT due to the fact that occupies 10x as much data as BERT and yields a 2-20% improvement in performance over BERT [28].

On the other side, there are more compressed versions of BERT such as DistilBERT, which has roughly half the number of parameters of BERT and takes 4 times less time to train than BERT but the cost in performance equates to 3% less than BERT. Another compressed variant of BERT is ALBERT [29], which significantly reduces

the number of parameters by 30% compared to BERT which allows for scaling up of the memory again. By scaling up the number of hidden-layer embeddings by 10-20x, significant performance gains can be observed on the BERT model [28].

More specific BERT variants can also be found. In the case of financial documents, FinBERT [30] is a variant of BERT which has been pretrained on financial documents, for the purpose of financial sentiment analysis. Since the main objective of this project was to create a financial information extraction model, it was very intriguing to investigate whether the financially pretrained model FinBERT, could outperform the original high-performing transformer-based models such as BERT and RoBERTa.

2.2.3 Information Extraction

Information Extraction is a specific task in the field of Natural Language Processing concerned with extracting a required specific entity, event, relationship or other information from a large number of texts and storing them in a structured form [31].

For the purposes of building a Knowledge Graph, it is necessary to produce triplets of information, consisting of two entities and the relation between them. The structure of a triplet extracted from a sentence can be seen in Figure 2.8.

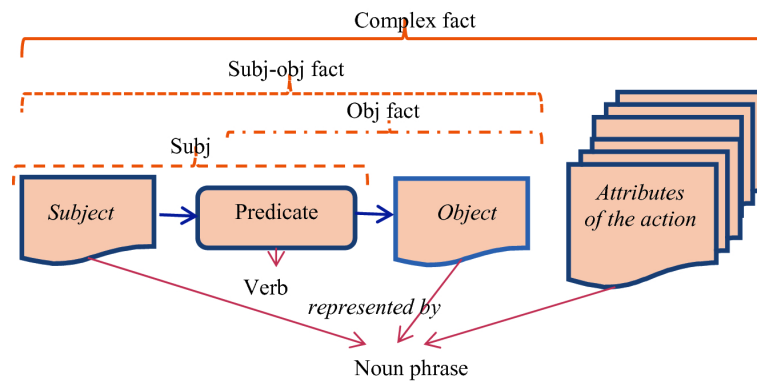


Figure 2.8: Triplet Structure from Factual Sentence [8]

Currently, one very popular way to generate triplets from sentences is through the

task of Open Information Extraction, where the triplets are extracted directly from the text and outputted. However, the most common method would be to use a named entity recognition model, where two named entities are extracted from a sequence, followed by a relation extraction model, where the relation between the two named entities is extracted [32]. Such a model is suggested to be the typical model in the review *Zhu et al., 2020* [33], including other steps such as text segmentation, parts-of-speech tagging and entity disambiguation. The generic information extraction model in that review is very similar to the one implemented in this project.

In the paper *Zhang et al., 2020* [34], the authors attempted to fine-tune BERT in the task of Information Extraction on the specific use case of domain-specific business documents. Fine-tuned BERT models were used to extract structured entities from regulatory filings and property lease agreements in Chinese. The task in this thesis is fairly similar, in that transformer-based models are used in an Information Extraction task upon a dataset of business documents. Key differences are that in this thesis, different variants of BERT models are used to obtain higher metrics and a relation extraction task is carried out too to obtain triplets for a Knowledge Graph. Another key difference is that the model in this thesis is multilingual and able to read articles in English and German amongst other languages and not only Chinese.

Within the financial domain, a different method for Information Extraction was explored in *Lembo et al., 2020* [35], which sought to exploit the semantic knowledge expressed in ontologies to improve question answering over unstructured data in the form of raw text. Whilst a very interesting approach, it is likely that transformer-based models may outperform this method in terms of metrics, but possibly not in terms of generalisability, whilst tackling the slightly different task of question answering within Information Extraction.

The review *Adnan and Akbar, 2019* [36] studies the limitations of Information Extraction methods and techniques for unstructured big data. Some of the key findings within the space of Natural Language Processing and unstructured text data were that some of the limitations of Information Extraction with regards to unstructured

data usability issues are caused by variation in text perspective, semantic understanding, context understanding and user's perspective. Some other issues regarding the language and domain limitations are caused by lack of multilingual systems, poor morphological languages, language ambiguities, modelling and knowledge and domain specific data. These are very common issues within Information Extraction and this project aims to tackle them through using a finance-specific dataset, transformer-based models to overcome semantic and context understanding and creating a multilingual model to interpret articles in many different languages alongside English and German.

The paper *Weischedel and Boschee, 2018* [37] describes the current state-of-the-art in Information Extraction and what can be accomplished in the field. The article identified conditions where Information Extraction from text has proven to be feasible in applications, such as building knowledge bases manually does not provide sufficient coverage of streaming data, noisy output can be tolerated, mitigating the need for very high precision and redundancy in streaming data can be used to overcome recall issues. This highlights one of the issues in using a fixed financial dataset in creating a Knowledge Graph, since factual statements and therefore facts described by the Knowledge Graph may change with time, so it is important ensure there is a live stream of textual data updating the Knowledge Graph.

2.2.4 Named Entity Recognition

Named Entity Recognition is a task within Natural Language Processing that aims to identify named entities within a sequence, using the BIO tagging scheme and forms the first part of the Information Extraction pipeline. Each token within a sequence is given one of three tags - Beginning (B) for tokens which start a named entity, Inside (I) for tokens after tokens with a tag of B which are inside the named entity and Outside (O) for tokens outside the named entity. Figure 2.9 displays an example of how the BIO tagging scheme can be used to identify named entities in a sequence.

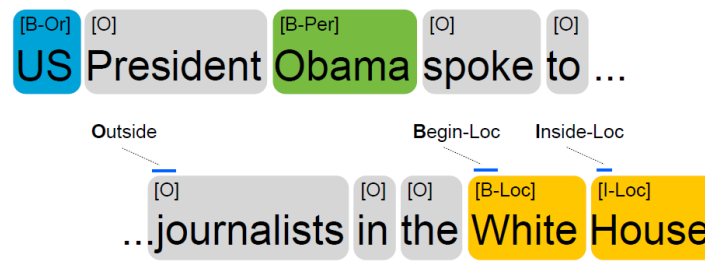


Figure 2.9: BIO Tagging Procedure Example [9]

Named Entity Recognition models have been found to obtain the highest accuracies when transformer-based models are used, due to their attention mechanisms providing greater generalisability. In this way, named entities within sequences can be easily identified and then used further in the task of Relation Extraction to generate triplets for Knowledge Graphs. Transformer-based models are widely known to obtain the highest performance in this task and so justify their use to obtain the best Financial Knowledge Graph in this thesis.

2.2.5 Relation Extraction

Relation Extraction is the task within Natural Language Processing concerned with taking a sequence containing two entities and predicting the relation between them and forms the second part of the Information Extraction pipeline. Again Relation Extraction models have been found to have the highest accuracies when containing transformer-based models. An example of a relation extraction task is shown in Figure 2.10 between different named entities.

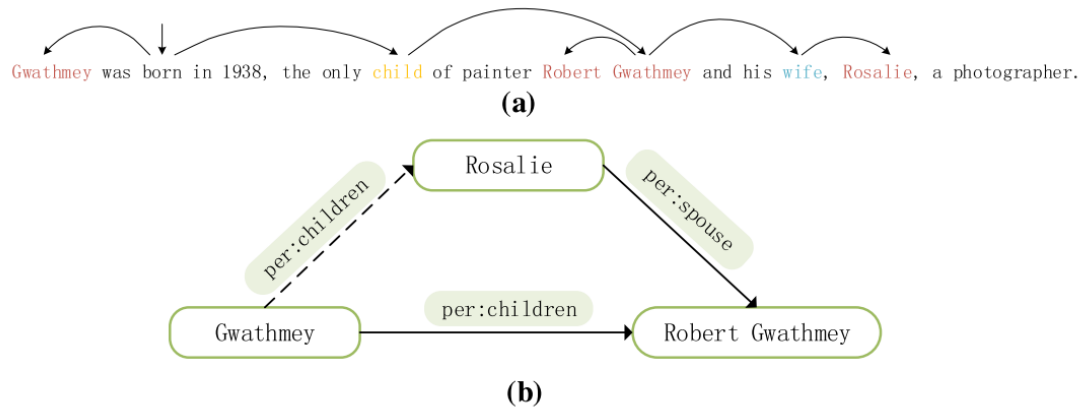


Figure 2.10: Relation Extraction Example [10]

The paper *Zhang et al., 2021* [38] introduced a new relation extraction model, called Open Relation Extraction (OpenRE), which aimed to extract novel relation types between entities from open-domain corpora, which plays an important role in completing the relation schemes of knowledge bases. This model highlights the importance in having a highly accurate relation extraction model, in order to improve the quality Knowledge Graphs.

2.2.6 Open Information Extraction

Open Information Extraction is an unsupervised Natural Language Processing task which focuses purely on the extraction of triplets from text sequences. This differs from Named Entity Recognition and Relation Extraction individually as all are different tasks, but when Named Entity Recognition and Relation Extraction are both supervised tasks but when combined the resulting fully trained model is equivalent to an Open Information Extraction model.

Gashteovski et al., 2017 [39] first introduced the concept of minimising facts in Open Information Extraction (MinIE), through extracting surface relations and their arguments (entities) from natural-language text in an unsupervised, domain-independent manner with high precision and recall. Despite being a slightly different task, the paper highlights the importance of obtaining high performance metrics

in the Open Information Extraction task to generate highly accurate triplets which can form part of a high quality Knowledge Graph.

Stanovsky et al., 2018 [40] presented a supervised learning approach to Open Information Extraction, through formulating the task as a sequence tagging problem. This approach was able to address challenges such as encoding multiple extractions for a predicate, using a bi-directional LSTM transducer. This approach is very similar to that in this thesis, using a supervised method and including sequence tagging. However, this thesis attempts to improve on this approach by using transformer-based models instead of a bi-directional LSTM.

A survey carried out in *Niklaus et al., 2018* [41] provides an overview of approaches used to tackle the task of Open Information Extraction. One of the main conclusions was that there was a lack of a well-defined task definition of what a valid relational triplet is and therefore it is difficult to compare methods and results. Furthermore, there is a lack of a gold standard dataset from a large corpus. Further issues with Open Information Extraction concern assertedness (aims to compress relations through making assertions), minimal propositions (extracting compact propositions that do not combine unrelated facts) and completeness and open lexicon (extracting all relations asserted in the input text). These issues are hoped to be tackled in this thesis through the use of transformer-based models to improve metrics such as precision and recall.

A span model for n-ary Open Information Extraction was implemented in *Zhan and Zhao, 2019* [42], which worked as a two-stage pipeline for predicate (relation) and argument (entity) modelling using bidirectional LSTM models in each step. This approach was somewhat similar to that of this thesis, as the named entity recognition and relation extraction models form the pipeline, except this project aims to use transformer-based models instead of bidirectional LSTM models to obtain more performant metrics. The performance metric under consideration in the article was the confidence score of extraction, but in this project only metrics such as precision, recall and F1 score are measured.

Within the financial domain, a BERT model has been used within Open Information Extraction for learning systematic behaviours in stock markets in *Wu, 2020* [43]. This project looks at using transformer-based models and variants of BERT to further improve performance metrics for the specific use case of risk management, which is different to case of stock market prediction presented in the paper mentioned.

Perhaps the most relevant research for this thesis was conducted very recently, where multilingual Open Information Extraction was carried out using BERT models as opposed to a bidirectional LSTM models for argument and predicate extraction in *Ro et al., 2020* [11]. In order to improve upon the performance metrics for use in a financial news dataset, variants of BERT are used in this thesis to investigate this. Furthermore, the use of a multilingual transformer-based models, tested in Spanish and Portuguese in the paper, will give the model the capacity to read press releases from Deutsche Bank and other articles in German.

Overall, it can be seen that from this chapter, the research attempted within this thesis is applied to a novel downstream task, in terms of financial Knowledge Graphs. Furthermore, this research aims to improve on performance metrics such as precision, recall and F1 score through using variant models with more specific financial pretraining or greater complexity, while also adding multilingual capacity to be able to read texts in other languages to generate triplets for a high quality Knowledge Graph.

Chapter 3

Data and Model Design

This chapter describes and explains the origins of the datasets used within this research and the overall design of the Information Extraction model pipeline and then Knowledge Graph. The datasets for both the prototype and main pipelines are different, with the latter being more financially oriented and collected by Deutsche Bank. The main Information Extraction model consists of two main transformer-based models in the tasks of Named Entity Recognition and Relation Extraction, resulting in triplets to be produced as an input for a Knowledge Graph. The Knowledge Graph can then be sorted using a topological sorting algorithm.

3.1 Introduction

The main experiment of this project aims to build a Multilingual Information Extraction pipeline which produces entity-relation triplets, which are used to build a Knowledge Graph. The information extraction pipelines must be trained on datasets and this chapter aims to describe the origin of the datasets used for training and testing, as well as the overall design of the pipeline and the structure of the Knowledge Graph.

The dataset used to fine-tune the transformer-based models in the prototype pipeline was open-source and freely available on Kaggle. The datasets used for the main and multilingual models were contained inside a MongoDB database and consisted of financial news articles in various languages from Refinitiv, relating to certain companies.

Both the prototype and main model pipelines were designed slightly differently. The prototype consisted of a joint entity-relation extraction model, where the named entity and relation extraction models were trained independently. This led to error propagation and so for the main model, the named entity and relation extraction models were trained together, aiming to identify subjects, objects and the verbs connecting them as a triplet. These factual triplets were then passed into a graphical software to generate Knowledge Graphs.

3.2 Data

This section describes the datasets used in this experiment for the prototype and main models. The first two datasets were labelled, allowing for supervised learning in the prototype model. Furthermore, the datasets for the main models (regular and multilingual) were both labelled and the training procedure was treated as being supervised. The datasets used for the training of the main models were internal and therefore private to Deutsche Bank.

The first dataset used for the prototype model was taken from Kaggle and was called the Annotated Corpus for Named Entity Recognition [44], while the second dataset used for Relation Extraction in the prototype was taken from GitHub [45]. The first Named Entity Recognition dataset, was labelled and each token was assigned a BIO tag and those named entities were given an associated type. The Relation Extraction dataset contained sequences with known named entities which were labelled with the relations between them.

For the main model, an internal Deutsche Bank dataset was used. This consisted of

sentences of factual information taken from financial news documents provided by Refinitiv and specific to different companies between certain dates. For example, the dataset selected initially for training and testing the models, calculating metrics and comparing models consisted of a set of financial news documents from Refinitiv, relating to Apple between the dates of 1st June and 10th August 2021. To obtain this dataset of articles, queries had to be written to extract the articles from a MongoDB database containing the articles.

Similarly, in order to test the multilingual capacity of certain models, another internal Deutsche Bank dataset was used. Again the dataset of choice consisted of a set of financial news documents in French from Refinitiv between the dates of 1st June and 10th August 2021. However, it was later found that the pipeline was too specific to the English language and so the same set of English financial news documents were used as the dataset for fine-tuning in this case.

Finally, the datasets used to create triplets for Knowledge Graphs consisted of a live stream of financial news articles from Refinitiv, referring to different companies. Therefore, this was a continuous and extremely large dataset, where all the articles had been stored in a MongoDB database. The dataset used to specifically obtain example Knowledge Graphs consisted of financial news related to Apple and Deutsche Bank between 30th August and 6th September 2021.

Therefore, the models used for the prototype use specific open-source datasets designed for the tasks of Named Entity Recognition and Relation Extraction. While the main model used datasets consisting of financial news from Refinitiv, related to certain companies, in particular Apple. The multilingual models were fine-tuned on the same set of articles but in a different language to evaluate their ability to interpret text in another language. Finally, large-scale texts were processed in order to create Knowledge Graphs.

3.3 Model Design

This section describes the overall design of the models and their pipelines used in this research. The prototype model contained a joint entity-relation extraction model pipeline, as displayed in Figure 1.1. A more detailed version of the independent training procedures of each of the Named Entity Recognition and Relation Extraction models and the overall pipeline can be seen in the flowchart in Figure 3.1, while the testing procedure can be seen previously in Figure 1.1 in the typical triplet extraction pipeline.

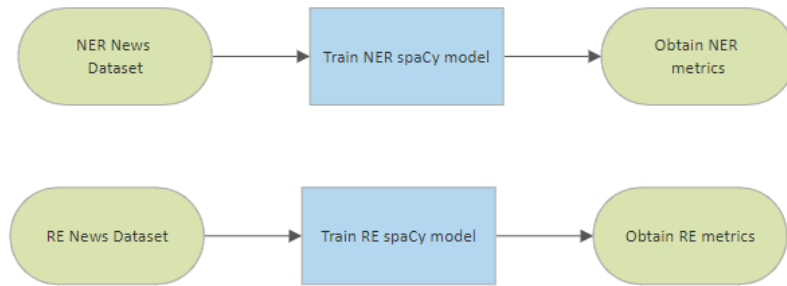


Figure 3.1: Flowchart displaying training procedure

The main model was designed for unsupervised triplet extraction only after it was found that joint entity-relation extraction contained propagation of errors. This was due to the fact that if an incorrect assignment were to be made in the Named Entity Recognition task, this error would be propagated into the Relation Extraction task, leading to an incorrect triplet. This issue was corrected through implementing a triplet extraction model in the form of an Open Information Extraction task, similar to that previously implemented by Deutsche Bank. A flowchart describing the processes involved can be seen in Figure 3.2.

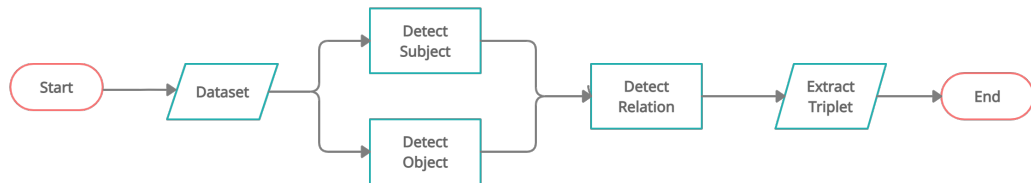


Figure 3.2: Flowchart displaying Triplet Extraction

From Figure 3.2, it can be seen that the financial news dataset of sentences is tokenised and passed into models where the entities are extracted. The subjects and objects specifically are identified in different steps and extracted. Then, using both the subjects and objects, the relations are then identified and extracted, releasing the output as a triplet. In addition to triplets, subject-object pairs where no relation was detected are also released as an output. This system is fairly similar to that implemented in MultiOIE [11] and can be seen in Figure 3.3

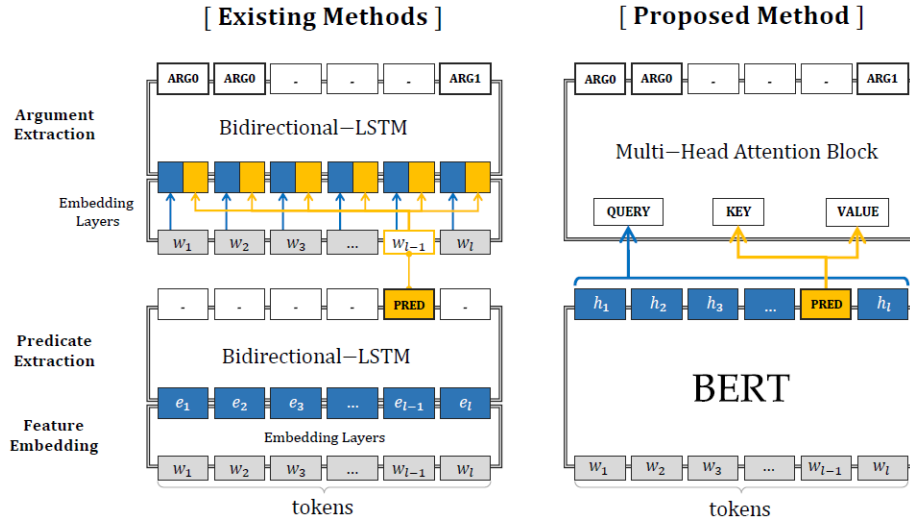


Figure 3.3: MultiOIE Triplet Extraction Structure [11]

From Figure 3.3, it can be seen that MultiOIE follows a similar structure to that implemented in this thesis, through using transformer-based models in the tasks of Predicate and Argument Extraction. However, a key difference lies in the fact that the relation is extracted first before the entities.

Following the triplet extraction, the triplets would be fed into the TigerGraph software to produce a Knowledge Graph, with a topological sorting algorithm used to provide a simplified visualisation of relevant parts of the graph, where news surrounding certain companies could be condensed.

Overall, this section described the designs of the different models. The first prototype model was designed as a joint entity-relation extraction pipeline and since

the two models were trained independently, suffered from error propagation and the solution was to combine the tasks together. Therefore, the main and multilingual models were designed to extract subjects and objects first before extract relations and returning triplets. These triplets were then passed into TigerGraph to produce Knowledge Graphs.

3.4 Summary

Overall, this chapter goes into detail on datasets and overall design of the models covered in this thesis for the task of Information Extraction for the purpose of creating financial Knowledge Graphs.

The prototype model used open-source datasets found on Kaggle in an attempt to create a joint entity-relation extraction pipeline. In this structure, the named entity and relation extraction models were trained independently and combined during the testing procedure to form a complete Information Extraction pipeline. However it was found during the testing phase that this structure gave rise to error propagation.

In order to solve the issue of error propagation, a structure that combined named entity recognition and relation extraction was introduced for the main model. This main model aimed to identify subject and objects in a sentence first, before locating the relation between them in the form of a verb. The datasets used to train the main and multilingual models were internal Deutsche Bank datasets, consisting of financial news articles between 1st June and 10th August, concerning Apple and provided by Refinitiv. The datasets used to produce Knowledge Graphs, consisted of financial news articles concerning Apple and Deutsche Bank provided by Refinitiv.

Chapter 4

Model Implementation

This chapter displays and rationalises the implementation of the Information Extraction model pipeline and Knowledge Graph, in terms of the packages, frameworks and software used to create and compare the models.

4.1 Introduction

The main aim of this research was to implement an Information Extraction pipeline for the purpose of generating triplets to be used in Knowledge Graphs. This chapter describes the implementation choices made while creating the triplet extraction pipeline.

The *spaCy* library was chosen over *NLTK*, largely due to its fast performance, with both packages providing strong advantages explained in a full breakdown. Then a further rationalisation is produced to explain that certain *spaCy* variant packages were used to provide transformer functionality within *spaCy*.

Through using *spaCy*, the text was then tokenized automatically using the pre-trained transformers, generating an encoded representation of the text. The representation is then passed into the named entity and relation extraction models,

Criteria	NLTK	spaCy
Multi-Language Support	Yes	Yes
Input/Output Type	Strings	Objects
Word Vector Support	No	Yes
Performance	Slow	Fast

Table 4.1: *NLTK* vs *spaCy* Comparison [13]

where subjects and objects are attempted to be identified, along with a verb as a relation. This triplet is then given as an output and ready to be used in the creation of Knowledge Graphs.

Finally, the software used to create the Knowledge Graphs was chosen to be Tiger-Graph, as it provides high speed responses for queries with tens of millions of entities and relations amongst other attractive reasons.

4.2 Implementation

The prototype model was implemented in *Google Colaboratory*, using *spaCy*, an advanced Natural Language Processing library. The source code written in the implementation of the main models was confidential and therefore private to Deutsche Bank. *spaCy* was chosen over other well known packages such as *NLTK* due to its fast performance and due to its word vector support. Table 4.1 highlights the key differences between *NLTK* and *spaCy*.

Figure 4.1 shows that while both *NLTK* and *spaCy* provide multi-language support, critical for this research, *spaCy* also provides word vector support in addition, as well as faster performance.

In terms of using transformer-based models, variant packages of *spaCy*, such as *spacy-transformers*, *sentence-transformers* and *spacy-sentence-bert* were utilised. *spacy-transformers* provides *spaCy* model pipelines that wrap Hugging Face’s transformers package and convenient access to state-of-the-art transformer architectures, such as BERT, GPT-2, XLNet, etc. [46]. The *sentence-transformers* frame-

work fine-tunes BERT/RoBERTa/DistilBERT/ALBERT/XLNet with a siamese or triplet network structure to produce semantically meaningful sentence embeddings. The *spacy-sentence-bert* package aims to wrap *sentence-transformers* directly in *spaCy*.

For the preprocessing steps, a pretrained BERT-specific tokenizer for each variant model was used to parse the text into tokens and then into an encoded representation for the BERT model. Tokenizers are responsible for converting tokens strings to IDs and back, as well as encoding/decoding. In addition, they can add new tokens to the vocabulary independently of the underlying structure and manage special tokens, such as masks and beginning-of-sentence tags, through adding them and assigning them to attributes in the tokenizer to ensure they are not split during tokenization [47].

The subject-object named entity recognition model before relation extraction aimed to identify the subjects and objects of sentences. These subjects and objects were identified from different parts of speech such as conjunctions and prepositions. Therefore, in the final relation extraction step, the verb connecting these subjects and objects was extracted as the relation and the whole triplet was returned in the output. If the relation could not be identified, then the subject-object pair was returned in the output.

The software used to build the Knowledge Graph using triplets produced by the RoBERTa triplet extraction model using the financial news dataset provided by Refinitiv, was TigerGraph [48]. TigerGraph is a native parallel graph software, providing high speed responses for queries with tens of millions of entities and relations, using a Graph Query Language for high-performance graph operations. TigerGraph was a good fit for Deutsche Bank's Knowledge Graphs since it would be able to scale out with the company's growing needs. Quite importantly, TigerGraph is able to provide deeper insights through queries which are able to traverse more than ten hops and perform complex analytics. Finally, TigerGraph also has the capability of containing multiple graphs with the same master database and a simple but powerful

graphical user interface. For these reasons, TigerGraph was the graphical software of choice for the Knowledge Graph. Unfortunately, the TigerGraph proof of concept was not implemented by Deutsche Bank engineers in time for this thesis and is therefore room for future work.

4.3 Summary

This chapter has provided a detailed explanation of all the implementation choices made throughout this thesis in the construction of an Information Extraction pipeline for the purpose of building financial Knowledge Graphs.

The first selection choice was to use the *spaCy* library over the *NLTK* package due to its fast performance and word vector support. *spaCy* also had many variant packages which enabled the use of *HuggingFace* transformer models directly within *spaCy*, including multilingual models. It also carried out all the required preprocessing steps in the triplet extraction task, such as stemming, lemmatization and tokenization. This made it a very attractive library for selection.

The triplet extraction model was intended to be implemented in a way which first identified the subject and object within a sentence. The model would then proceed to attempt to locate the relation between the subject and object in the form of a verb. In the case where a relation could not be identified, the subject-object pair would be returned.

The RoBERTa triplet extraction model would be used to produce triplets to be passed into a Knowledge Graph software. The Knowledge Graph software of choice was TigerGraph, due to its ability to provide high speed responses for queries with tens of millions of entities and relations, amongst other reasons.

Chapter 5

Training & Testing

The aim of this chapter is to describe the training and testing procedures. This includes the metrics used for analysing the quality of the Information Extraction pipelines and therefore the best pipeline for creating a Knowledge Graph.

5.1 Introduction

In this experiment, the main goals included training the prototype and main Information Extraction pipelines, through fine-tuning the pretrained transformer models and then testing on established datasets and evaluating performance through metrics.

The prototype model contained a joint entity-relation extraction structure and therefore, each of the two models, named entity extraction and relation extraction, were trained individually. This separate training procedure led to error propagation and therefore a more compact triplet extraction model such as that involved in Open Information Extraction would perform better in the main model. This procedure aimed to identify subjects and objects before searching for a relation involving a verb between them.

The triplet extraction model was tested on a financial news dataset from Refinitiv and certain metrics such as precision, recall, F1 score and accuracy were determined for both the monolingual English models and multilingual models. In order to compare metrics, the same English dataset was used for both types of model.

5.2 Training & Testing

The Information Extraction pipelines were trained, with the prototype in a slightly different way to the main one. One of the significant aims of this project was to find which pretrained BERT model variant performed the best in the pipeline for a financial news dataset and so their key metrics needed to be compared.

The first prototype model was trained by fine-tuning (during training) each of the pretrained Named Entity Recognition and Relation Extraction models individually before combining them for use in the overall Information Extraction pipeline for testing. However, it was found that the overall Information Extraction pipeline when tested, produced low metrics due to the propagation of errors through each individual Named Entity Recognition and Relation Extraction model. Therefore it was decided that a triplet extraction model, similar to that previously implemented by Deutsche Bank and more along the lines of Open Information Extraction may perform better.

The main model initially used a pretrained BERT transformer and fine-tuned during training in an unsupervised way on an internal Deutsche Bank financial news dataset, relating to Apple between the the dates 1st June and 10th August 2021. During the training procedure, the data flows through the pipeline design outlined in Figure 3.2, where the subjects and objects are detected first in the sentences. Following this, the relation is extracted from the sentence using the subject and object, before returning a triplet or subject-object pair if the relation cannot be obtained.

When testing the models, there were certain metrics of interest calculated in the task of Information Extraction, such as precision. This is equivalent to the number

of true positives divided by the sum of true positives and false positives and is therefore a measure of how much of the information is returned as being correct and is displayed in Equation 5.1 [49].

$$Precision = \frac{No. \text{ Correct Predictions}}{No. \text{ Predictions}} \quad (5.1)$$

Another specific metric of interest is recall, which is a measure of how much relevant information the system has extracted (i.e. the coverage of the system). This is equivalent to the number of true positives, divided by the sum of true positives and true negatives and gives an idea of the frequency of triplets in the text. The formula for recall can be seen in Equation 5.2 [49].

$$Recall = \frac{No. \text{ Correct Predictions}}{Total \text{ No. Possible Correct Predictions}} \quad (5.2)$$

Another metric of interest in the task of Information Extraction and particularly popular in Natural Language Processing in general is the value for F1 Score. The F1 Score is calculated as the harmonic mean of precision and recall and is somewhat comparable to the accuracy. This can be calculated in Equation 5.3 [49].

$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (5.3)$$

The final metric of interest in this evaluation of the models is accuracy, which is simply put, the proportion of correct predictions in general.

Using the above metrics, it is possible to gauge and compare the quality of the different transformer-based models. The same training and testing procedures were used for the multilingual models, except they were trained on datasets in a different language and compared using the same metrics described above.

5.3 Summary

Overall, this chapter describes the training and testing procedures involved in this project in creating an Information Extraction pipeline for the purpose of building a financial Knowledge Graph in order to provide information in Risk Management.

Initially, the prototype model consisted of two models, named entity recognition and relation extraction, which were trained separately. After testing on a small-scale, it was found this model was prone to error propagation and therefore it was thought that a model that combined both tasks would perform better.

Therefore, the main model consisted of a triplet extraction model where subjects and objects were identified, followed by finding relations between them as verbs. Various different transformer models were investigated and fine-tuned on financial news datasets provided by Refinitiv. They were then tested on further datasets of a similar type and certain metrics such as precision, recall, F1 score and accuracy were all calculated to compare the performance between the models.

In the case of the multilingual models, in order to compare their performance with that of the English models, the same English financial news dataset was used. This enabled the multilingual transformer models to produce comparable metrics.

Chapter 6

Results and Discussion

This chapter attempts to describe and explain all of the results obtained in each of the experiments for different transformer-based models. This includes performance metrics as well as examples of where the models succeed and fail. Finally, examples of the working topological sorting algorithm are used to show the usefulness of such an algorithm in a domain-specific task.

6.1 Introduction

During this project, a prototype and main models were created, with the prototype taking on the structure of a joint entity-relation extraction pipeline and was trained on open-source Kaggle datasets. However, after small-scale evaluation it was found that this model was insufficient and suffered from error propagation issues and so could not accurately produce meaningful triplets. Therefore, for the main model, which was trained on internal Deutsche Bank financial news datasets provided by Refinitiv, a triplet extraction model along the lines of Open Information Extraction was used.

This pipeline was able to identify subjects, objects and the relations between them more reliably. Different transformer-based models, such as BERT, FinBERT and

Model	Precision	Recall	F1 Score	Accuracy
BERT	0.762	0.196	0.312	0.510
DistilBERT	0.751	0.187	0.300	0.502
RoBERTa	0.778	0.212	0.333	0.525
FinBERT	0.765	0.201	0.319	0.514

Table 6.1: English Model Performance Metrics

RoBERTa, were used in the pipeline to assess which performed the best. In order to evaluate the overall performance, different metrics were compared such as precision, recall, F1 score and accuracy and it was found that RoBERTa outperformed the other models.

In order to test the multilingual capacity of the pipeline, multilingual transformer-based models such as Multilingual BERT and XLM-RoBERTa were placed in the pipeline. A French financial news dataset from Refinitiv was used to fine-tune and test the pipeline but it was found that the relation extraction model was too specific to the English language. Therefore, in order to compare performance with the previous monolingual models, the same English financial news dataset was used and their metrics compared. Surprisingly, it was found that while XLM-RoBERTa was able to outperform the Multilingual BERT, it was unable to outperform the monolingual RoBERTa model.

Some examples of Knowledge Graphs produced using triplets obtained using the RoBERTa triplet extraction pipeline were created and visualised to highlight the strong use case of triplet extraction within the domain of Risk Management.

6.2 Information Extraction

Following the training and testing of the different variants of the BERT models in the Information Extraction pipeline, certain metrics were calculated in order to compare the performance of each of them against each other. These performance metrics can be observed in Table 6.1.

From Table 6.1, it can be seen that the RoBERTa model outperforms the rest with an F1 score of 0.333 and an accuracy of 0.525, while the more compressed variant, DistilBERT, while being faster to train, had to compromise on their accuracy. For the purposes of this research, computational cost was not the most important factor when comparing models, whereas the above performance metrics were and this was where the smaller models came last. As expected, BERT performed reasonably well and can be used as the benchmark for this comparison, since it had been pre-trained on an English dataset prior. Interestingly, despite the relevant pretraining on financial documents, FinBERT did not produce the highest metrics when fine-tuned and came second best to RoBERTa. This was due to the high amount of data storage capacity, allowing for greater learning, transferability and generalisability. In general, the high precision and low recall metrics imply that the financial news datasets from Refinitiv, include text where there are few identifiable triplets, but when there is one, there is a high probability that the model will correctly identify it. The low F1 scores in general can be attributed to both the likelihood that there is often more than one triplet in a sentence, making them hard to extract and the fact that the relation extraction model was limited to finding relations only in the form of verbs.

Examples of where the best model succeeded and failed can be seen in Figures 6.1 to 6.10. Analysis of the triplets produced from these examples can be seen in Table 6.2.

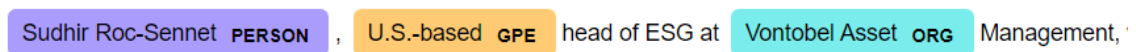


Figure 6.1: Named Entity Recognition Triplet Example 1

Figure 6.1 displays the first example, where named entities are extracted successfully but the model fails to extract triplets linked without the use of verbs as relations, as shown in Table 6.2. It therefore highlights the limitations of this model in being able to extract complex relations between named entities, not involving verbs.

Google **ORG** , whose European **NORP** headquarters is in the Irish **NORP** capital Dublin **GPE** , says it pays taxes where it is required to do so by law.

Figure 6.2: Named Entity Recognition Triplet Example 2

Figure 6.2 shows the second example, where again the named entities are extracted, but the complex relations between them cannot be identified by the model. This is again as a consequence of the lack of a verb connecting them.

Thomson Reuters Corp **ORG** <TRI.TO>, the parent of Reuters News **ORG** , offers Pondera **PRODUCT** software, which spots suspicious patterns among

Figure 6.3: Named Entity Recognition Triplet Example 3

The third example in Figure 6.3, correctly identifies the named entities and the relation between them. However this triplet is not factual and cannot be used in a Knowledge Graph. The factual triplet would incorporate the name "Pondera" in the "software" object, therefore being more specific. Another possible factual triplet to be identified in this sentence, would be to have the relation "is the parent of" between the named entities "Thomson Reuters Corp" and "Reuters News".

ID.me **ORG** has this out-of-the-box solution."

Figure 6.4: Named Entity Recognition Triplet Example 4

The fourth example in Figure 6.4, shows that the model has correctly identified the named entity involved here and produced a satisfactory triplet. However, this statement is vague and the triplet is not a representation of a fact for a Knowledge Graph, despite being an accurately extracted triplet.

Market researcher Juniper **ORG** expects global annual **DATE** sales of online identity verification services to reach nearly \$16.7 billion **MONEY** in 2025 **DATE** , up 77% **PERCENT** from this year **DATE** .

Figure 6.5: Named Entity Recognition Triplet Example 5

Figure 6.5 highlights a successful example, where the triplet and named entities are successfully extracted and the triplet is factual. An area for improvement for the model in this case would be to add more detail to the triplet extracted.

James Gorman PERSON , chief executive at Morgan Stanley ORG <MS.N>, said
Monday DATE that if most employees are not back at the bank's
Manhattan GPE headquarters in September DATE , he will be "very
disappointed." [nL2N2NW243]

Figure 6.6: Named Entity Recognition Triplet Example 6

Figure 6.6 is another example similar to the first and second, where the named entities are successfully extracted, but the model fails to identify triplets due to the complex relations between named entities which do not involve a verb.

Arizona GPE credited ID.me for reducing fraud and ensuring
timely benefits delivery. New Jersey GPE described ID.me as "a

Figure 6.7: Named Entity Recognition Triplet Example 7

Figure 6.7 provides an interesting example, where the model correctly predicts a factual triplet, but could again do with more detail regarding what the organisation can be credited with exactly.

G7 ORG could agree minimum 15% PERCENT corporation tax rate

Figure 6.8: Named Entity Recognition Triplet Example 8

The eighth example in Figure 6.8 shows again how the model can successfully identify a triplet and the entities involved. The triplet produced is also factual and therefore able to be used in a Knowledge Graph.

KLP ORG , APG among investors stepping up pressure on companies

Figure 6.9: Named Entity Recognition Triplet Example 9

Example	Triplet	Correct?	Factual?
1	('Sudhir Roc-Sennet', PERSON), ('US-based', GPE)	No	No
2	('Google', ORG), ('Dublin', GPE)	No	No
3	('Thomson Reuters Corp', ORG), 'offers', ('software', ' ')	Yes	No
4	('ID.me', ORG), 'has', ('solution', ' ')	Yes	No
5	('Juniper', ORG), 'reach', ('\$16.7 billion', MONEY)	Yes	Yes
6	('James Gorman', PERSON), ('Morgan Stanley', ORG)	No	No
7	('Arizona', GPE), 'credited', ('ID.me', ORG)	Yes	Yes
8	('G7', ORG), 'agree', ('15% corporation tax rate', ' ')	Yes	Yes
9	('KLP', ORG)	No	No
10	('ID.me', 'ORG'), 'founded as', ('Craigslist', ORG)	Yes	Yes

Table 6.2: Examples Evaluation

The penultimate example in Figure 6.9 is a good example of where the model identifies a named entity, but since there is no object or distinct relation for triplet, no triplet is extracted. This represents an example of a true negative result.

No company may be benefiting more than ID.me, founded in
 2010 DATE as a Craigslist ORG for verified military veterans and valued
 at \$1.5 billion MONEY in financing this year DATE by funds including
 Alphabet Inc's <GOOGL.O> CapitalG.

Figure 6.10: Named Entity Recognition Triplet Example 10

The final example in Figure 6.10 shows a factual triplet was able to be extracted successfully from this sentence, which can be used in a Knowledge Graph to visually represent connections between named entities.

Overall, from this experiment, it was found that factual triplets are fairly infrequent in financial news articles in general, but that when relevant factual sentences containing named entities do appear, the model rarely fails to identify the correct factual triplet. Where the model usually fails to identify factual triplets, is where there are complex relations connecting the entities without use of a verb to connect subject and object and therefore, such as possessive or colloquial phrases, the model fails to identify the factual triplet correctly if at all.

Multilingual Model	Precision	Recall	F1 Score	Accuracy
Multilingual BERT	0.754	0.189	0.302	0.501
XLM-RoBERTa	0.763	0.204	0.322	0.512

Table 6.3: Multilingual Model Performance Metrics

6.3 Multilingual Information Extraction

In the next part of the investigation, the multilingual capacity of the models was tested, through attempted fine-tuning on a German dataset of financial news applying to certain companies. However, it was found that the nature of the triplet extraction pipeline was too specified to the English language, making it difficult to be able obtain meaningful triplets in another language such as German. Therefore, the models were fine-tuned on the original English financial news dataset, to be able to evaluate their performance specifically compared to the English pretrained models. The different multilingual models selected were BERT (trained on 102 languages in the uncased case) and XLM-RoBERTa (trained on 100 languages in the regular case) [50]. Following training and testing, the performance metrics for the two multilingual models were calculated and can be seen in Figure 6.3.

As can be seen from Figure 6.3, XLM-RoBERTa outperformed the multilingual version of BERT, with the F1 score metrics being 0.322 for XLM-RoBERTa and 0.302 for the Multilingual BERT, highlighting the improved performance using XLM-RoBERTa. This is likely due to the fact that RoBERTa is able to store large amounts of data, roughly ten times more than BERT and so is able to fine-tune itself better to the dataset, while also being more transferable as a result. The English RoBERTa model likely outperformed the XLM-RoBERTa model due to the model's pretraining on text from 100 different languages and therefore being less specific to English.

The multilingual Information Extraction pipeline could be improved to obtain meaningful triplets from text in another language, by making the relation extraction procedure more transferable across languages. This would enable improved triplet extraction in other languages for use in Knowledge Graphs.

6.4 Knowledge Graphs

Finally, the optimal RoBERTa triplet extraction model was taken and used to produce triplets from different test datasets. Each test dataset consisted of a set of financial news articles from Refinitiv belonging to specific companies and was used to create their own individual Knowledge Graph.

These triplets containing subjects, objects and relations from financial news articles from the dates 30th August to 6th September 2021 were then used to create Knowledge Graphs. News relating to different companies was used to produce Knowledge Graphs specific to each company and examples can be seen in Figures 6.11 and 6.12.



Figure 6.11: Apple Knowledge Graph

Figure 6.11 shows a Knowledge Graph representing named entities mentioned in financial news provided by Refinitiv, concerning Apple in the week from 30th August to 6th September 2021. The relations have not been included for easy interpretability.

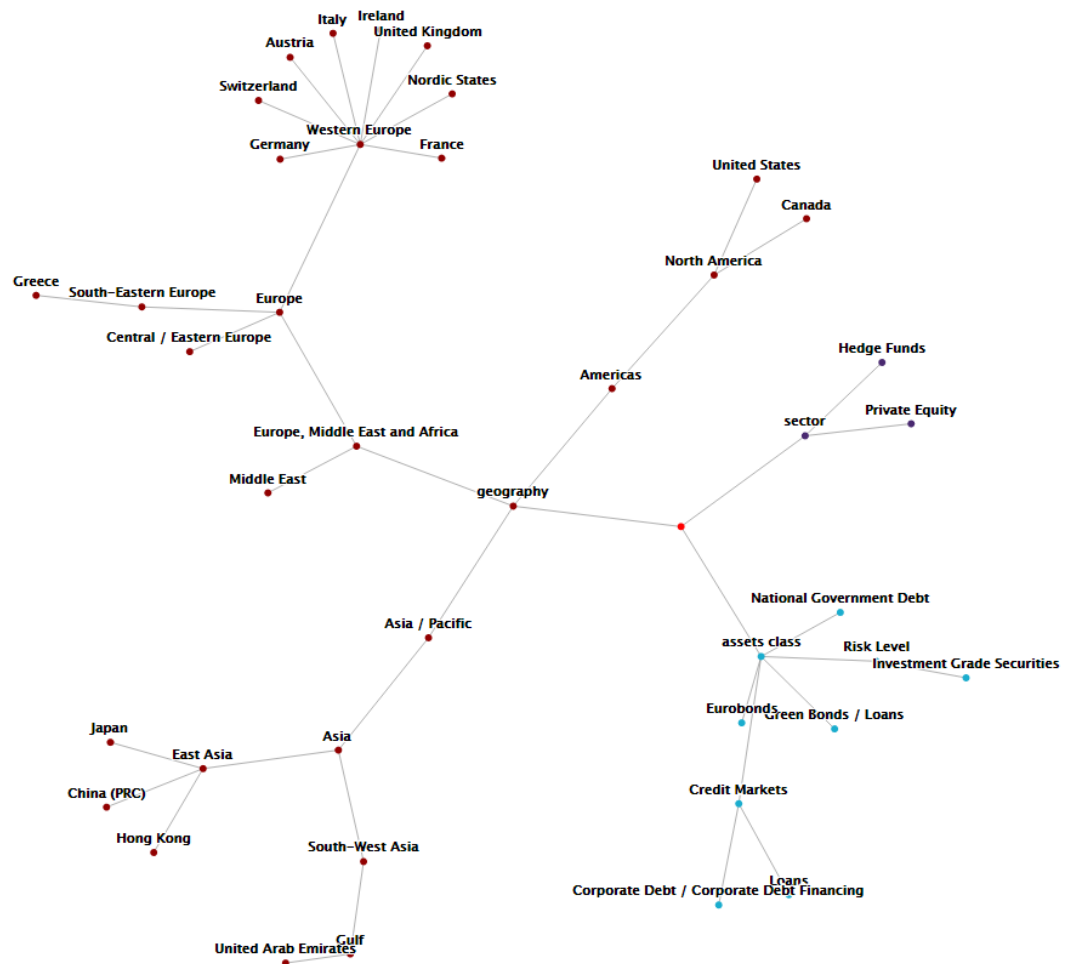


Figure 6.12: Deutsche Bank Knowledge Graph

On the other hand, Figure 6.12 displays a Knowledge Graph representing named entities mentioned in financial news provided by Refinitiv, concerning Deutsche Bank from 30th August to 6th September. It can be seen that Deutsche Bank is spoken about in more detail within finance, with more asset classes listed, whereas Apple is spoken about with influences more widely across the globe, in many more regions.

At the time of producing the Knowledge Graphs, development of the TigerGraph software at Deutsche Bank was not complete, therefore the Knowledge Graphs in Figures 6.11 and 6.12 were visualised using the Deutsche Bank NLP Analytics Dashboard. The use of TigerGraph for the visualisations of Knowledge Graphs will be implemented in future work.

6.5 Summary

The main objective of this thesis was to identify the highest performing transformer-based model in the case of Information Extraction of financial news texts for the purpose of building Knowledge Graphs. This chapter provides the results obtained for the English and Multilingual Information Extraction pipelines, as well as examples of where the best performing transformer-based model, RoBERTa, succeeds and fails in this task. It finally shows the use case of Knowledge Graphs in being able to highlight named entities of recent discussion in the news associated with a particular company and how they are all connected.

In the main model, the Refinitiv financial news dataset was used to fine-tune the transformer-based Information Extraction pipeline, before being tested on a smaller number of articles to provide performance metrics. From these metrics, it was found that the RoBERTa model produced the highest metrics with an F1 score of 0.333 and an accuracy of 0.525. Interestingly, it was able to outperform FinBERT, despite it being pretrained on financial news data prior. As expected it was able to improve upon the state-of-the-art BERT model in financial Information Extraction. Analysis of examples of where the best model failed showed that the triplet extraction model struggled to identify complex relations that did not consist of verbs, such as those that consisted of possessive or colloquial phrases.

For the multilingual model, initially the same Refinitiv financial news dataset was used as before except in German, to test the multilingual capacity. Unfortunately, due to the pipeline being too specific to the English language, it was unable to

extract meaningful triplets in its current state. Therefore, in order to test the multilingual transformer models against their English counterparts, the same English datasets as before were used to train and test the models in the pipeline. It was found that the RoBERTa model was able to outperform both multilingual models, likely due to the multilingual models being less specific to English as they had been pretrained on 100 different languages prior. Therefore, the RoBERTa model was chosen to produce Knowledge Graphs from financial news triplets obtained from further test financial news datasets.

The Knowledge Graphs shown in the cases of Apple and Deutsche Bank highlight that entities with large separations can still be visualised and their relations understood and rationalised (although not displayed for clarity purposes), even when they may not seem intuitive. Therefore, the usefulness of Knowledge Graphs can be seen in a field such as Risk Management, where unintuitive relations between entities could be discovered earlier in order to minimise potential losses to be incurred. However, this depends on the quality of the Knowledge Graph and this research has improved upon the state-of-the-art through providing improvements on the baseline BERT in Financial Information Extraction using RoBERTa and FinBERT and through the initial implementation of multilingual financial Information Extraction pipelines using transformer-based models such as XLM-RoBERTa and Multilingual BERT.

Chapter 7

Conclusions and Future Work

This chapter aims to describe the main findings of this thesis and any future work that it may lead to. The first part explains the conclusions of the research conducted, what was learnt and where there is room for improvement. The second part goes into more detail in the room for improvement, as well as further work that this research would lead to, both from a scientific and business perspective.

The experiments carried out in this thesis first attempted to build Triplet Extraction pipelines with different transformer-based models and compare them, while the second took two multilingual Triplet Extraction pipelines with transformer-based models and compared their performance on a dataset from a different language. These triplets were then displayed graphically as Knowledge Graphs, in order to obtain meaningful insights. This section aims to summarise the overall findings discovered during this research, as well as the next steps to progress this work further in the field of financial Knowledge Graphs.

7.1 Conclusions

In summary, this research highlights the very powerful use case for Knowledge Graphs within Finance and specifically Risk Management. In the first prototype

model, open-source Kaggle datasets were used to train a joint entity-relation extraction pipeline. However, it was found that given that the two models were trained independently, this led to the propagation of errors through the model.

The main model however was trained internal Deutsche Bank datasets consisting from financial news from Refinitiv related to certain companies. This model was designed in a way which combined both entity and relation tasks, aiming to identify the subject and object of sentences first, before extracting the relation and returning the triplet or subject-object pair if the relation could not be identified. Different transformer-based models were experimented with and it was found that RoBERTa performed the best with metrics of an F1 score of 0.333 and accuracy of 0.525, due to the additional storage capacity provided by RoBERTa. The financially pretrained FinBERT transformer surprisingly fell short of this with metrics of an F1 score of 0.319 and accuracy of 0.514. Examples of where the models succeeded and failed were identified, displayed and rationalised and it was found that the model pipeline in general failed to identify triplets containing complex relations which did not include verbs.

The multilingual information extraction models were designed in a similar fashion to the main information extraction model, with subjects and objects extracted first from sentences and relations being identified afterwards and returned as a triplet. The difference lay in the fact that they were trained on financial news articles from Refinitiv but in different languages and different multilingual transformer-based models were used. It was found that XLM-RoBERTa outperformed Multilingual BERT, with F1 scores of 0.322 and 0.302 respectively and accuracies of 0.512 and 0.501 respectively, due to the additional storage capacity provided by the model. The English RoBERTa model likely outperformed the multilingual XLM-RoBERTa model due to the multilingual model's pretraining on text from 100 different languages and therefore being less specific to English.

Finally, the best model on the English financial news dataset, RoBERTa, was used to produce triplets and these were used to produce Knowledge Graphs for news

related to Apple and Deutsche Bank. The differences between news concerned with the companies was highlighted, where Deutsche Bank was related more to financial markets and Apple contained interests in many more regions of the world. Both of these Knowledge Graphs demonstrated the success of Open Information Extraction in this case.

These findings were significant as the current state-of-the-art in financial Knowledge Graphs has been improved upon, through using different transformer-based models such as RoBERTa and FinBERT which improved upon the baseline set by BERT. Multilingual Information Extraction models have also been created for financial Knowledge Graphs, with XLM-RoBERTa performing best in English.

7.2 Future Work

In order to improve on the work carried out in this thesis, the triplet extraction pipeline can be further improved by enabling extraction of complex relations which include not only verbs, but possessive and colloquial phrases for example. Furthermore, an adaption to the multilingual model would be required through making the model less specific to English. This would require a more detailed study of lexical patterns across languages to ensure multilingual models can perform equivalently across languages. Finally, the development of the TigerGraph software will allow for higher performance and improved sorting and Knowledge Graph manipulation than previously created.

An interesting extension to this work, would be to attempt to implement the KBERT model [23], discussed earlier in Chapter 2 in the case of Information Extraction for the use of Knowledge Graphs. This relates to the idea of injecting triplets into sentences as domain knowledge and using this to train language representation models, since they would then be knowledge-enabled. This would increase the accuracy of the models even further and in general would lead to more accurate Knowledge Graphs.

In the case of Knowledge Graphs, from a business perspective, it would be highly advantageous to have quasi-static Knowledge Graphs, where facts can change dynamically. In the world of finance, things are constantly changing, meaning that the information and facts displayed in Knowledge Graphs must always be up to date. Therefore, by implementing dynamically changing Knowledge Graphs using quasi-static reference data, it would be possible to always be in possession of the most up to date information in Knowledge Graphs, with long-standing facts remaining in the graphs.

With regards to Knowledge Graphs and interpreting them, it seems there would still be significant advances to be made in improving their user interfaces. This would include not only visualisations, but through the addition of graph-based interactions with humans, to allow even greater ease of use.

Another potentially interesting area of research within using Natural Language Processing in Risk Management would include studies into the volume and sentiment of speech of press releases in the public domain. This would involve taking second-order time derivatives of certain Natural Language Processing metrics such as sentiment, to understand the rate of growth of sentiment in the public domain. This would inevitably give an idea of which news would be trending in the media and to what extent and would provide a stronger understanding of crowd psychology.

Bibliography

- [1] Yashu Seth. Introduction to Question Answering over Knowledge Graphs. <https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/>. Accessed: 01/8/2021.
- [2] B. Wu. An introduction to neural networks and their applications in manufacturing. *Journal of Intelligent Manufacturing*, 3:391–403, 1992.
- [3] Y. Yu et al. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31:1235–1270, 2019.
- [4] Gonzalo Nápoles Greg Van Houdt, Carlos Mosquera. A review on the long short-term memory model. *Artificial Intelligence Review*, 53:5929–5955, 2020.
- [5] Hung Viet Ho Xuan-Hien Le and Giha Lee. Application of Gated Recurrent Unit (GRU) Network for Forecasting River Water Levels affected by Tides. 2019.
- [6] Ashish Vaswani et al. Attention Is All You Need. *Neural Information Processing Systems*, 2017.
- [7] Jacob Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2019.

- [8] N. Khairova et al. Open Information Extraction as Additional Source for Kazakh Ontology Generation. *Intelligent Information and Database Systems*, 2020.
- [9] T. Rocktaschel and S. Riedel. Sequence Labelling. *Statistical Natural Language Processing*, 2020.
- [10] Z. Li et al. Improve relation extraction with dual attention-guided graph convolutional networks. *Neural Computing and Applications*, 2020.
- [11] X. Wu. Event-Driven Learning of Systematic Behaviours in Stock Markets. 2020.
- [12] GPT-3 Vs BERT For NLP Tasks. <https://analyticsindiamag.com/gpt-3-vs-bert-for-nlp-tasks/>. Accessed: 28/7/2021.
- [13] spaCy: Advanced NLP in Python. <https://www.kaggle.com/sanikamal/spacy-advanced-nlp-in-python/>. Accessed: 08/8/2021.
- [14] ontotext. What is Information Extraction? <https://www.ontotext.com/knowledgehub/fundamentals/information-extraction/>. Accessed: 12/7/2021.
- [15] E. Farjana et al. Identification of Correct Triples on Open Information Extraction. *The 34th Annual Conference of the Japanese Society for Artificial Intelligence*, 2020.
- [16] Amit Singhal. Introducing the Knowledge Graph: Things, Not Strings. <https://blog.google/products/search/introducing-knowledge-graph-things-not/>, May 2012. Accessed: 12/7/2021.
- [17] Mohammed Zaki and Aparna Gupta. III: EAGER: Knowledge Graph Mining for Financial Risk Analytics. *Rensselaer Polytechnic Institute, Troy, NY, United States*, May 2017.

- [18] W. J. Wouter Botzen et al. The Economic Impacts of Natural Disasters: A Review of Models and Empirical Studies. *Review of Environmental Economics and Policy*, pages 177–178, 2019.
- [19] L. Ehrlinger and W. Wöß. Towards a Definition of Knowledge Graphs. *SEMANTICS 2016: Posters and Demos Track*, 2016.
- [20] A. Hogan et al. Knowledge Graphs. 2021.
- [21] M. Yasunaga et al. QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering. 2021.
- [22] C. Wang et al. Language Models are Open Knowledge Graphs. 2020.
- [23] W. Liu et al. K-BERT: Enabling Language Representation with Knowledge Graph. 2020.
- [24] Franz Inc. Optimizing Fraud Management with AI Knowledge Graphs. <https://allegrograph.com/articles/optimizing-fraud-management-with-ai-knowledge-graphs/>. Accessed: 03/8/2021.
- [25] S. Elhammadi. Financial Knowledge Graph Construction. 2020.
- [26] D. Ajwani et al. Engineering a Topological Sorting Algorithm for Massive Graphs. 2011.
- [27] Kenneth Ward Church. Natural Language Engineering. pages 155–162, December 2016.
- [28] Suleiman Khan. BERT, RoBERTa, DistilBERT, XLNet — which one to use?s. <https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8>. Accessed: 28/7/2021.

- [29] Radu Soricut and Zhenzhong Lan. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. <https://ai.googleblog.com/2019/12/albert-lite-bert-for-self-supervised.html>. Accessed: 21/7/2021.
- [30] Dogu Araci. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. June 2019.
- [31] Dengyun Zhu et al. Information Extraction Research Review. *J. Phys.*, pages 155–162, 2021.
- [32] S. Ji et al. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. 2021.
- [33] D. Zhu et al. Information Extraction Research Review. *Journal of Physics*, 2020.
- [34] R. Zhang et al. Rapid Adaptation of BERT for Information Extraction on Domain-Specific Business Documents. 2020.
- [35] D. Lembo et al. Ontology Mediated Information Extraction in Financial Domain with Mastro System-T. 2020.
- [36] K. Adnan and R. Akbar. Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, pages 1–23, 2019.
- [37] R. Weischedel and E. Boschee. What Can Be Accomplished with the State of the Art in Information Extraction? A Personal View. *Computational Linguistics*, pages 651–658, 2018.
- [38] K. Zhang et al. Open Hierarchical Relation Extraction. *Association for Computational Linguistics*, page 5682–5693, 2021.

- [39] K. Gashteovski et al. MinIE: Minimizing Facts in Open Information Extraction. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, page 2630–2640, 2017.
- [40] G. Stanovsky et al. Supervised Open Information Extraction. *Proceedings of NAACL-HLT*, page 885–895, 2018.
- [41] C. Niklaus et al. A Survey on Open Information Extraction. 2018.
- [42] J. Zhan and H. Zhao. Span Model for Open Information Extraction on Accurate Corpus. 2019.
- [43] Y. Ro et al. Multi2OIE: Multilingual Open Information Extraction Based on Multi-Head Attention with BERT. 2020.
- [44] Abhinav Walia. Annotated Corpus for Named Entity Recognition. <https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus>. Accessed: 03/8/2021.
- [45] Walid Amamou. Relation Extraction Transformer. https://github.com/walidamamou/relation_extraction_transformer. Accessed: 03/8/2021.
- [46] Spacy Transformers Documentation. <https://spacy.io/universe/project/spacy-transformers>. Accessed: 25/8/2021.
- [47] Tokenizer. https://huggingface.co/transformers/main_classes/tokenizer.html. Accessed: 09/8/2021.
- [48] TigerGraph. <https://www.tigergraph.com/product/>. Accessed: 08/8/2021.
- [49] A. Fraser. Rule-based Named Entity Recognition. *Information Extraction*, 2016.
- [50] Multilingual Transformer Models. <https://huggingface.co/transformers/multilingual.html>. Accessed: 10/8/2021.