Knowledge Graphs in the Age of GPT-3

Subtitle: Using large language models to derive facts

Master's Thesis submitted to the
Faculty of Informatics of the *Università della Svizzera Italiana*
in partial fulfillment of the requirements for the degree of
Master of Science in Informatics
Neural Networks and Knowledge Graphs

presented by
Michael Mazourik

under the supervision of
Prof. Cesare Alippi
co-supervised by
Prof. Mark James Carman

June 2022

I certify that except where due acknowledgement has been given, the work presented in this thesis is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; and the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program.

Michael Mazourik
Lugano, 12 June 2022

To my loving family and friends.

...

"Education is the most powerful weapon
which you can use to change the world."
-Nelson Mandela

Abstract

Throughout the last few decades, the field of machine learning leaped in advanced automation and understanding in natural language processing. One of the key limitations for training for sequential data was that neural network training was not parallelizable. The development in transformers-based architecture shifted the focus of the field towards language model embeddings allowing for better results on down-stream tasks by first being trained in an unsupervised manner and allowing for parallel training. This change proved to be important as natural language tasks can now improve with the help of unstructured text, as opposed to often scarce or even noisy datasets. By transferring part of the computational graph to a specific task, state of the art natural language processing models have become available to the public, which reduces computational time and effectively democratising access to large language models.

At the same time in the particular task of knowledge graph generation, it becomes evident that high quality datasets without ontologies are difficult to produce as multiple works attempt to gradually improve them. Therefore, the language model-based approaches are of utmost relevance for their ability to increase performances with respect to the amount of unstructured data rather than the currently lacking high quality supervised data.

We hope to define this opportunity in the thesis by investigating the ability of large language models to generate knowledge graphs. We take a hands-on approach, with methods tested against established standards. Furthermore, because extracting highly structured graphs from unstructured text is a difficult process, it is critical to provide tools that can quantitatively as well as qualitatively analyze the process. We investigate the task of Open Information Extraction, which aims to transform general text into lists of fact triplets, from both a theoretical and practical standpoint.

Overall, the models demonstrate an extremely high level of accuracy in learning the structure and wording of the output triplet fact lists. They also show promising results when fine-tuned, achieving the highest recall in some configurations. However, when being queried with few-shot learning, it still appears that the models are performing at baseline levels. As discussed from a theoretical perspective, evaluating language models against benchmarks can quickly become at risk of subjectivity. The reasons being that meta-physical concept of 'facts' can have multiple valid and correct multiple forms in the knowledge graph. Moreover, drawing the line between objective and subjective text fragments can even difficult for humans evaluators which is why the numerical evaluation is further enhanced with qualitative analysis.

Developing a language model to generate knowledge graphs approach becomes valuable because it can provide extensions and lower labor costs for downstream knowledge graph applications such as search engine optimizations, allowing for more accountability and clearer communication.

Acknowledgements

This master's program and thesis have been challenging and an immense opportunity to refine the skill-set in Artificial Intelligence. I would like to thank Professor Cesare Alippi, Professor Carman Mark James and Daniele Grattarola for helping in the supervision of this thesis as well as well as with their technical expertise.

Moreover, it would be hard to compose a exhaustive list with whom the discussion of the thesis topic was over viewed. Some of the key discussions took thanks to classmates Manav Chakraborty, Brian Pulfer, Clemente Pasti and researchers Imanol Schlag.

Finally, I would like to thank the fantastic facilities of USI for providing a proper environment to write, code and review this thesis.

Contents

Figures

Tables

Chapter 1

Introduction

The field of machine learning has brought advanced automation and understanding in natural language processing during the recent decades. Knowledge graphs generated using language models have the potential to extend multi-relational reasoning capabilities as well as provide explanations for the outputs of models which are hard to interpret.

In this master thesis, we aim to explore large language models and their abilities to generate knowledge graphs. We will focus on a practical approach with the methods being tested on established benchmarks. Moreover, as extracting highly structured graphs from unstructured text is a difficult process, it is important to provide tools that can analyse the process in both a quantitative and qualitative aspects. The aim of this thesis is twofold:

To investigate the ability of language models to generate knowledge graphs from unstructured text, and compare different approaches for doing so.

To develop tools that can be used to analyse and evaluate the quality of the generated knowledge graphs in a quantitative and qualitative manner.

1.1 Machine Learning in Natural Language Processing

"Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications." Liddy [2001]

1.2 Knowledge Graphs

Knowledge graphs have been a key component of modern technology in multitude of ways. Google's knowledge graph was responsible for finding the correct search result on one-third of the 100 billion monthly queries in 2016 Goo. Moreover, the utility of knowledge graphs has been shown in academic works by helping academia create structured relations between a variety of papers related to the recent pandemic of Covid-19 Dessì et al. [2021] Hope et al. [2020].

1.3 Use of Machine Learning in Knowledge Graphs

Knowledge graphs (KG) are a standard data format to organise multi-relational data. However, with the recent success of transformer-based language models such as GPT-3 and BERT, the future of KG comes under question. In this research paper, we investigate how these two technological interact under a practical use case while enforcing a formal procedure using concrete datasets.

1.4 Research Questions

Throughout our research, we aim to generate knowledge graphs using advanced machine learning techniques enhanced with large language models.

In this study, we aim to show the potential of unsupervised methods in the task of knowledge graph creation by comparing hypothetical models to control models.

For this purpose, we compared performance metrics such as F1 scores amongst control models based on published articles and our novel architectures.

The research firstly focused on how to create generate knowledge graphs from text written a natural language such as English. Since this is a broad investigation, the research needs to be broken down into several smaller sub-questions:

1.4.1 Primary Research Goal

- To evaluate the potential of large language models such as GPT-3 and T5 to produce for the creation of knowledge graphs.

1.4.2 Secondary Research Goal

- To help provide open-source tools to evaluate and debug the process of the aforementioned task

1.5 Thesis Composition

The thesis organisation follows the following structure:

- Introduction leads the thesis with motivation on the work.
- Related Works takes a look into previous research in the field of the construction of knowledge graphs including datasets, benchmarks, systems, supervised and unsupervised models.
- Background showcases the relevant information to understand the technical perspective of the thesis.
- Material and Methods shows the concrete manner in which the experiments are conducted.
- Results demonstrates the valuable metrics and charts derived from the aforementioned experiments.

- Discussion consolidates the thesis in a final chapter to discuss the findings, shortcomings and potential future directions to follow up this work.

Chapter 2

Related Works

This chapter primarily focuses on the works that have been used for understanding the task of the knowledge graphs construction in the age of large language machine learning models.

2.1 Preface to Literature Review

Researching the topic of knowledge graphs construction ties strongly with many similar but distinctive terms. For instance, the topic can also be referred to as a subset class of knowledge bases with graphs that are limited to the algorithmic data-structure whereas the super-class of knowledge bases can include other data structures as well. Due to a wide array of terms used for similar tasks, the related work section aims to cover the most relevant works for the thesis despite the evolving terminology through time and expansive literature.

The task of Open Information Extraction (OIE) is also synonymous with Automatic Knowledge Base Construction. The "input is ... a corpus, and its output is a set of extracted relations" Banko et al. [2007]. The authors state that this technique has advantages over ontology-based extraction techniques because the relations can also be directly extracted from the text unlike previously proposed methods.
In a similar fashion relation extraction (RE), deals with the similar idea of extracting relations in the data-structure of triplets. In the case of RE, whether the entities are pre-specified depends on the benchmark.

2.2 Algorithmic Non-Learning Approaches

The algorithmic approaches are distinguished due to their ability to predict triplets without training data. One of the first traces of open information extraction came with the need to avoid pre-encoding knowledge and ability to apply it across a wide array of text. In the article presented by Hearst [1992], an initial model is introduced that is based on recognisable lexico-syntactic patterns.

A decade later, an autonomous system by the name of *Baseline KnowItAll* was introduced to extract large collections of facts from the web with preliminary results Etzioni et al. [2004]. Further improvements by the author in the pattern learning, subclass extraction and list extraction created the new model *KnowItAll*.This method was the first to connect the task of information extraction with the ability to make it independent of a fixed schema for the knowledge graph Etzioni et al. [2005].

In 2008, the same group formalised the task of Open Information Extraction to which they also provided the system called *TEXTRUNNER* Banko et al. [2007]. In comparison with *KnowItAll*, *TEXTRUNNER* achieved a 33% reduction in error rate and improved recall of fact extraction whilst reducing running time.

To contrast to their previous work with shallow models, the group created deeper syntax-models to increase performance. These models, *SRL-IE-UIUC* and *SRL-IE-Lund* focused on semantic role labelling. Although they achieved higher performance, the value of the statistics based *TEXTRUNNER* was still essential which why they created a mix between the two models called the "Smart Union". This ensemble model is the predecessor to OpenIE which is the most popular system for the task of open information extraction Christensen et al. [2011].

Another short coming to OpenIE was the interpretation of noun compound phrases which also contain facts and attributes. *RelNoun2.2*, a sub-component of OpenIE 5.1 received a major update which focused on demonyms and compound relational nouns Pal et al. [2016]. To deal with numerical facts, it was important to introduce a deep syntax model which could reason information from numbers or quantity-unit phrases. The group introduces *BONIE*, a model that bootstraps using deep syntactical patterns to generate numerical relations. Through this switch, the model increases 1.5x the yield compared to OpenIE and 15 points gain on numerical facts on the ClueWeb12 dataset. Due to these advantages, *BONIE* was also merged into the OpenIE system Saha et al. [2017].

To complete the upgrade to OpenIE 5, *CalmIE* was appended to the system to increase performances of complex sentences. The group designed the model to reducing the conjunctivity of the sentences by splitting them into several simpler clauses. Saha et al. [2018]

However, this was not the first presentation of using simpler sentences for the task of OpenIE, as *ClauseIE* focused on linguistic traits of clauses in the sentence to perform the task of open information extraction. ClauseIE depends on parsing sentences into grammatical dependencies and domain-independent lexica. At the time of its publication, the model achieved between 53.41-59.74 % precision on datasets of the Reverb, Wikipedia, and NYT. Del Corro and Gemulla [2013]

An issue arising with *ClauseIE* is the over-specificity of the extracted triplet. A new model, *MinIE* attempts to reduce the unnecessary arguments surrounding the entities and relations while retaining high re-call and precision Gashteovski et al. [2017].

2.3 Supervised Learning Approaches

One of the first papers outlining the supervised approaches to build knowledge graphs describes models by treating the problem as a classification of each word in an extraction to be part of specific arguments Stanovsky et al. [2018]. In this paper the *RNN-OIE* system uses a bi-LSTM transducer to predict relations between words in a sentence, by identifying words that can be syntactic heads of relations, and performing a single labeling to get the extractions.

As a futher iteration to *RNN-OIE*, the model is upgraded to *SpanOIE* by treating the data as a span labelling problem. The model consists of two parts: a predicate module and an argument module. The predicate module is used to find potential predicate spans in a sentence, while the argument module is used to classify all possible spans in the sentence as subject or object. The model provides a confidence score for every extraction. Zhan and Zhao [2020]

The *Multi2OIE* is a sequence-labeling system that uses a query, key, and value setting inspired by the Multi-modal Transformer replacing the bi-LSTM used in *SpanOIE* Ro et al. [2020]. It follows the pattern to extract relational tuples from a given sentence in two steps. The first step is to find all predicates in the sentence. The second step is to extract the arguments associated with each identified predicate.

Connecting back to algorithmic based models, *OpenIE6* connects the labelling and sequences approaches with an iterative labeling-based system Kolluru et al. [2020a]. Iterative Grid Labeling (IGL) architecture, which treats OpenIE as a 2-D grid labeling task. The grid's dimensions are $M \times N$, M is a pre-defined maximum number of extractions and N is the sentence length. Moreover, OpenIE6 employs BERT embeddings and is trained on a dataset generated by OpenIE4.

On the other hand, *IMOJIE* uses an LSTM decoder to generate a tuple one word at a time, producing $< rel >$ and $< obj >$ tokens to indicate the start of relation and object Kolluru et al. [2020b]. The generated extractions are concatenated with the original input sentence and passed back through *IMOJIE* to generate the next extraction. This process is repeated until the $< EndOfExtractions >$ token is generated.

In terms of domain specific knowledge graph generation, Dessì et al. [2020] have performed extensive research on implementing methods and creating datasets with methodologies to analyse the accuracy, recall and F1 scores of models using a "gold standard" in the topics of scientific paper.

2.4 Unsupervised Learning Approaches

2.4.1 Language Models

In 2017, a new network architecture coined the transformer achieves state of the art performance without the need of convolution or recurrence by using the attention method. This advance led to the production of more capable models as now these networks can be trained in a more parallelizable manner as opposed to recurrent neural networks Vaswani et al. [2017].

Generative Pre-Training

Using this architecture, a new version of embeddings is also implemented using unlabeled text from 7'000 unpublished books. A particular type of these embeddings, coined generative pre-training (GPT) shows to be of great use to out-perform state of the art in specific natural language processing tasks such as natural question answering and common sense reasoning, semantic similarity, language inference and classification Radford et al. [2018].

By transferring more of the generatively pre-trained layers, the task-specific performance also experiences gradual increase after supervised fine-tuning. Moreover, it shows zero-shot behavior, which implies that the model can adapt to its dataset with only a forward pass rather than demanding additional architecture modifications or specific datasets.

Another iteration of the generative pre-training model is published which demonstrates much stronger zero-shot behavior due to its larger size. GPT-2 contains 1.5 billion parameters and is trained on 40 GB of text from the WebText dataset Radford et al. [2019].

GPT-3 set of models are trained on a weighted mix of 300 billion tokens including a dataset with a trillion words stored in 45TB of data. The largest of the models, GPT-3 contains a whole 175.0 billion parameters and achieves extremely strong performances in natural language processing tasks without fine-tuning, demonstrating zero-shot behaviour Brown et al. [2020].

Bidirectional Encoder Representations from Transformers

In another case, the model of BERT, Bidirectional Encoder Representations from Transformers is trained by both conditioning the left and right context. The BERT model is trained on BooksCorpus and Wikipedia which totals up to 3,3 billion words. By concatenating one additional layer to the pre-trained BERT model, it is possible to achieve state of the art performance in a wide array of natural processing language tasks. Devlin et al. [2018].

A further iteration of BERT, RoBERTa, has optimised the methods for training language models and achieved higher performance. By adjusting to larger batches, adapting the the sentence objective, creating longer text sequences, and dynamically changing the masking pattern during training, RoBERTA achieves state-of-the-art results on SQuAD and RACE datasets. Liu et al. [2019]

The advance in language models has also enabled a strong advantage in data augmentation, the task related to increasing training data Papanikolaou and Pierleoni [2020]. In combination with both *BERT* and *GPT-2*, the data augmented relation extraction model *DARE* has enabled for the ability to increase training data in the relation extraction which is similar to the open information extraction task.

Text-to-Text Transformer Transformer

Using the encoder-decoder, Text-to-Text Transformer Transformer(T5) aims to create a more effective pre-trained language model to fine-tune on NLP downstream tasks. To optimise the procedure of language models, T5 evaluates transfer learning performance using various configurations such as different objective function, architectures, datasets, transfer approaches and other configurations. This methodology enables the achievement state-of-the-art on many downstream NLP benchmarks. The group discovers methods to reduce to the computational cost by drawing conclusions on experiments that language models architecture can achieve similar performance to task-specific architectures. Moreover, that the most efficient configurations share encoder-decoder weights and the objective function should be that of denoising further helps efficiency.

Regarding the size of the model and dataset, training on large and diverse text benefits is important and the method could reach the size of 11 billion models which is the largest at the time of publication. Finally, it is becoming apparent that unsupervised pre-training is starting to produce comparable results to fine-tuning after pre-training and the ensemble models can outperform individual models Raffel et al. [2019].

2.4.2 Knowledge Graph Generation Models

Contrasted to supervised methods, the unsupervised methods aim to generate knowledge graphs without the use of labels but by using unstructured text directly. The benefit of these approaches is that the learning capabilities are increased with the quantity of abundant unstructured text rather than scarce structured datasets.

Match and Mapping

In this method, the authors show that the knowledge graphs information is found in language models through their ability to learn general facts Wang et al. [2021b]. The algorithm constitutes of two stages. The first aims to generate candidate facts by implementing a modified beam search across the weights of attention matrix. The secondary stage re-integrates the likely facts into the knowledge graph found in a pre-existing knowledge graph.

Zero-shot translation

Contrasted to previous methods, a new formulation of the task was created by viewing the knowledge graph construction problem as a text-to-triple translation framework Wang et al. [2021a]. The paper combines three different datasets types, open information extraction, relation classification and knowledge probe. Therefore, they are introducing a new type of model called the DeepEx. The model uses this framework to create knowledge graph in two stages: "Generating" and "Matching".

The first stage is generation which helps find candidate sequence of tokens for noun phrase pairs. The stage does finds the sequence of tokens by creating beam-search across the attention scores which are part of the transformer architecture. The k-best scores are then elected to proceed to the following stage.

In the second stage of ranking, the scores are compared to the sentences using a trained model which employs language model embeddings. The model is tasked with minimising the loss between the ideal candidate and maximising it with the incorrect candidates.

2.5 Evaluations and Datasets of Knowledge Graph Constructions

2.5.1 Datasets

OpenIE Datasets

The Open Information Task was introduced to the literature as a first paradigm which does not require manual annotations to automatically generate knowledge graphs. The motivation behind moving away from traditional IE datasets which contain fixed classes is that the quality of the models scales linearly with manual input . Banko et al. [2007]

During advances in machine learning, labelled datasets with golden triplets were introduced. Stanvosky et al. developed a large scale dataset and performance evaluation for open information extraction called "OIE2016" Stanovsky and Dagan [2016].

From another perspective, the Wire57 focuses on the quality of annotations by implementing annotation guidelines Léchelle et al. [2018] . Moreover, the evaluation script is publicly available. The functions return the evaluation metrics across the complete dataset composed of 57 sentences associated with 343 extracted triplets.

Using the same source sentences as in OIE2016, a more accurate and standardised annotation, Crowd automatic open Relation extraction Benchmark (CarB) was enabled through platforms such as Amazon Turk. The authors, who are also major contributors to the OpenIE system, mentioned that the previous datasets have been too small or lacked standardisation. Bhardwaj et al. [2019].

Supervised OpenIE Datasets

With the growing influence of machine learning in the field of data analysis, the first dataset enabling supervised learning approaches was generated. Moreover, the conversion to machine learning framework is non trivial due to the generative and subjective aspects of knowledge graph generation. Stanovsky et al. introduce a variant of "OIE2016" based on sequences Stanovsky et al. [2018]. The novelty in this work in viewing the problem as a sequence tagging problem allowed for systems to learn from data rather than rule specific algorithms. Moreover, with matching evaluation functions the framework allowed for more efficient validations.

In a similar approach, a further iteration of "OIE2016" nicknamed "Re-OIE2016", Zhan et al. Zhan and Zhao [2020] improved the dataset in both the training and evaluating splits. In the training part, the authors expanded the volume using bootstrapping techniques including lower confidence triplet lists. From the evaluation side, they accurately re-annotated the test split of the dataset. Moreover, in the complete dataset, they added additional indices to be able to frame the task as a sequence to sequence problem.

In a similar approach to how the original "OIE2016" was transformed from "QA-SRL" dataset, the large scale open information extraction benchmark, "LSIOE" derived its facts from "QA-SRL2" Solawetz and Larson [2021]. Additionally, the authors contributed additional conversion heuristics to ensure data quality. With a larger source, the dataset reached a 10-fold volume compared to its largest predecessor "OPEIC."

The latter dataset focused on extracting a list of triplets using an algorithm based approach. The source text is given by Wikipedia articles and then compared to already established knowledge graph datasets such as DBPedia and YAGO. Moreover, due to the computer-based extraction, this dataset has over 341 millions of triplets. Due to such a large amount, the dataset is further sub-divided into "OPIEC-Clean" and "OPIEC-Linked" sub-corpora. The clean retains triplets between entities and concepts and the linked dataset retains only entities which have Wikipedia articles linked.

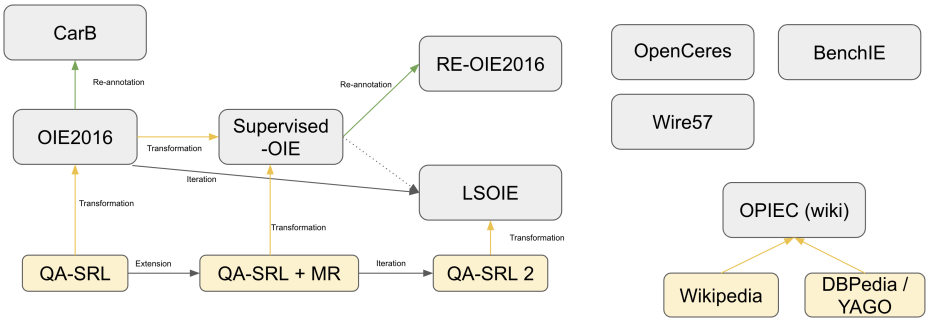

Figure 2.1. Diagram showing the evolution of open information extraction datasets

As a further comment to the state of datasets in the Open Information Extraction Field, *BenchIE* aims to increase the completeness of benchmarks Gashteovski et al. [2021]. To do so, *BenchIE* focuses on being fact-based and lists exhaustively all fact synsets related to a given a sentence. This approach permits for the evaluation of other aspects such as compactness and minimality.

In addition to datasets, additional tools such as annotators have also been introduced. *AnnIE* is an interactive web based application that helps manually label triplets from unstructured text

Friedrich et al. [2021]. In their work, the authors also introduce two datasets composed of 150 sentences sourced from "NYT10k" dataset which is formed with random sentences from New York Times articles. The first dataset is a subset with the focus on triplets containing named entity and the second one based on verb-mediated relations.

Chapter 3

Background

3.1 Natural Language Processing

It is an area of focus in artificial intelligence which addresses the subject of processing natural languages such as Italian, English used for human communication.

In computer science, a series of of natural language characters is denoted as a string. To analyse the string, it needs to be further divided into tokens which can be encoded either on the character, word or morphemes level.

In this paper, we will use the following notation for word encoding:

- Words: $w \in W$

3.1.1 Spanning a wide of array of subjects

The field is difficult as it crosses many areas of research departments such as communications, philosophy, psychology, culture, and languages. Specifically knowledge graphs deal with the notion of ontologies which is a science in philosophy aiming to describe "the kinds and structures of objects, properties, events, processes and relations in every area of reality" Floridi and Smith [2004]. As stated in Smith's work, Ontologies do not seek not "predication, but rather taxonomy." In the following section, we will explore the current implementations of open information extraction's approaches to derive facts from text.

3.2 Knowledge Graphs

3.2.1 Forms of knowledge graphs

A knowledge graph or base is a type of data structure which aims to contain valuable information. In computer science, traditional graphs are defined as a set of vertices and edges. Vertices are connected via edges which allows for advanced representations of various systems. Their mathematical notations are:

- Edges: $e \in E$
- Vertices: $v \in V$

The two main forms of representations of graphs are adjacency matrices or adjacency lists. Matrices are not necessarily superior to the list and vice versa. Each has its advantages when it comes to specific computational tasks.

For instance, retrieving a specific edge takes O(1) time steps in the matrix form of graphs, but O(log(|E|)) time steps in list form. This means querying specific edges is a lot less computational expensive, but it comes with more space allocation.

Knowledge graphs are a specific form of directed graphs in which the vertices and edges are respectively called "entities" and "relations". This type of graph has a feature that the edges/ relations can be of different types. Moreover, because it is a directed graph the vertices could be denoted as "heads" or "tails", with respect to a given relation. Formally speaking, the knowledge graph data notations are:

- Entities: $e \in L$
- Relations: $r_i \in R$
- Tails: $t_j \in T \subseteq L$
- Heads: $h_k \in H \subseteq L$

To store a knowledge graph, it is common practice to employ the list form of graph storage because it is more efficient with respect to memory. The data is composed of a list of triplets in the following format:

$$[(h_1, r_1, t_1), (h_2, r_2, t_2), ..., (h_n, r_n, t_n)].$$

Furthermore, the entities and relations names tend to be converted to index values with a separate dictionary to reduce the data size. This is important because some of the largest datasets include millions of triplets with millions of different types of relations and entities. On this scale of large data, it would be ineffective to store the data in standard tensor form. The knowledge graph in tensor format, $\mathbf{T}$ has the dimension of $R \times T \times H$. Unlike traditional graphs which can be transformed into a matrix (tensor of order 2), knowledge graphs are of order 3 due to the relation having an additional label.

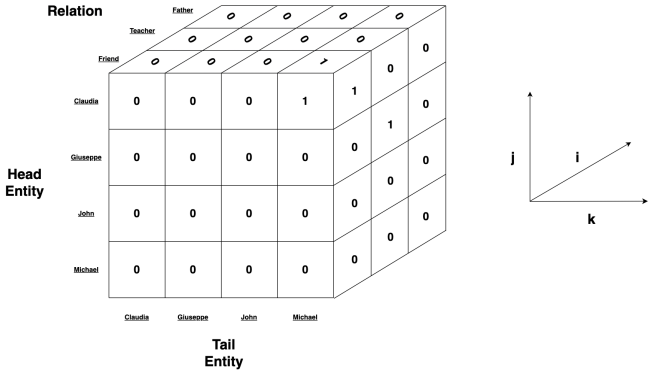

Figure 3.1. Knowledge graph adjacency tensor with the respective dimensions of head, tail and relation

Converting the knowledge graph from a list structure into a tensor format has disadvantages that it tends to be a sparse tensor, but it enables it to be included in certain types of operations where a fast retrieval of values is also necessary. In certain cases of automatic knowledge graph construction such as open information extraction, the entities are not standardised which proves this storage format to be difficult to handle as it requires additional post-processing to cluster entities to improve quality of the data structure.

3.3 Non-triviality of Evaluating Knowledge Graph Generation

Although a knowledge graph is extremely versatile to contain relevant information in form of facts, evaluating its quality proves to be a task of its own. In this section we aim to investigate why it is not clear whether the generated knowledge graph is of high quality or not.

Contrasted to standard machine learning tasks, where we can tell with extreme certainty whether the picture is a cat or dog, it becomes evident that giving a number for accuracy and recall has to be further researched in the task of knowledge graph creation.

Knowledge graphs expressed a list of triplets containing three arguments (head, relation, tail), help bridge the reasoning between humans whilst remaining useful to computer. As a trade-off to this level of expressiveness, the data structure can have infinite permutations in its instances, knowing which one is the ideal and objective knowledge graph (if there is one) is not trivial. Therefore, there are no obvious target labels in general so machine learning evaluation techniques need to adapt.

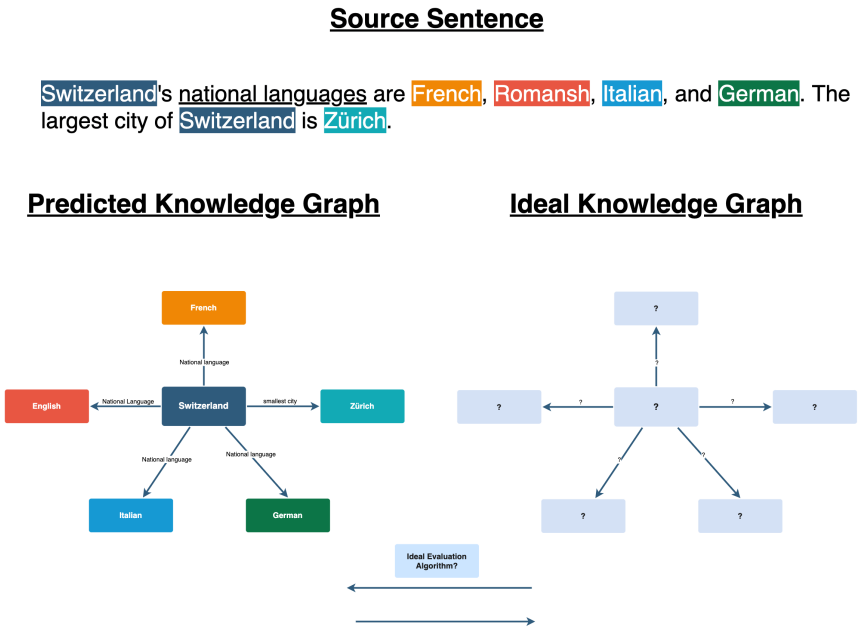

Figure 3.2. Diagram depicting the evaluation and ideal standard for a knowledge graph remains unclear

To start answering the question of evaluation, current academia suggests that we need to undertake several constraining assumption.

Assumption 1: Extracted triplets shared the vocabulary with the source text

In the task of open information extraction, we aim to formally facilitate "domain-independent discovery of relations extracted from text" Banko et al. [2007]. Despite the fact that there is no theoretically "pre-specified domain or vocabulary" Christensen et al. [2011], the practical implementation of knowledge graphs requires a given vocabulary to express them.

- Vocabulary: $w \in V$
- Knowledge Graph Vocabulary: $w \in V_{kg}$
- Source Text Vocabulary: $w \in V_{text}$
- Application Base Vocabulary: $w \in V_{application}$

From an applied perspective, works in literature aim to generate triplets in which the arguments are subsets of words from the source text. In other words, to simplify the evaluation and task, the extractions only contain sequences of words found in the original text limiting the vocabulary of the knowledge graph. In the original supervised open information extraction work, the authors refer to a similar idea as 'assertedness' which refers to the fact that the phrases are parts of the original sentence although it may appear to be more direct to include them in other ways. Stanovsky et al. [2018]. Thus, in current evaluations $V_{kg} = V_{text}$.

For example in *RE-OIE2016* Zhan and Zhao [2020], we notice the following sentence - triplet match:

```
Sentence: "There were 143 households out of which 30.1 % had children under the age of 18
living with them , 49.7 % were married couples living together , 11.9 % had a female house-
holder with no husband present , and 36.4 % were non-families ."

Triplets: ["30.1 %", "had", "children under the age of 18 living with them"]
```

Although a fact can be also phrased using alternate to symbolise similar sense, deviating from the source text vocabulary increases the difficulty of the task. In future work, it would be of value to generate datasets with a given vocabulary in mind which could have practical task in the downstream tasks for which $V_{kg} \neq V_{text}$, but rather $V_{kg} = V_{application}$. As an example, the vocabulary of words could be used to appeal to targeted audience to help communicate with more clarity while also providing useful analysable information.

Assumption 2: Human manual annotations can extract correct facts:

To evaluate knowledge graphs, human input has always been crucial to output a performance metrics although it is difficult to maintain objectivity. In the case of one of the original system *TEXTRUNNER* Banko et al. [2007], evaluators directly assessed the accuracy based on human input of a subset of facts which provides them with a final metric.

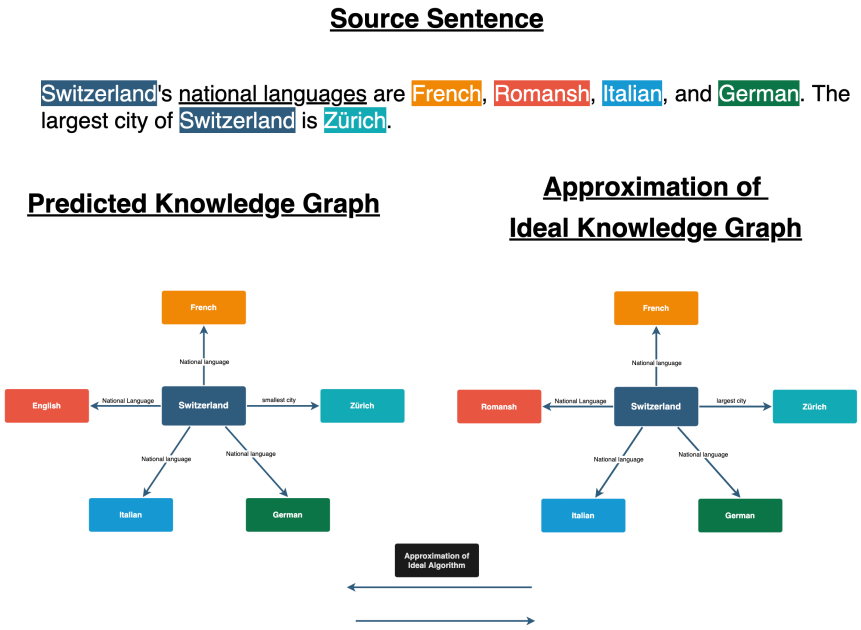

Figure 3.3. Diagram depicting an evaluation that depends on approximated ideal knowledge graph manually labeled by human annotators

Then in the case of supervised open information Stanovsky and Dagan [2016], a fully automatic evaluator has been created given a set of ideal triplets for each sentence, often referred to as gold triplets *OIE2016*. In this case, the set of ideal annotations is provided with manual annotation

In a similar way, the authors of *BenchIE* Gashteovski et al. [2021] create a template based knowledge graph annotation which are clustered by semantic meaning. In this advancement, although manual labor is provided by humans, the templates adapt to accept valid facts deviating from the gold triplets.

Due to many factors, annotations are vulnerable to subjectivity when choosing to include a certain word or fact. Since objectivity is crucial in science, authors have attempted to increase it by providing guidelines to annotators.

3.4 Data Structure

In the face of the scientific formal definition of the knowledge graph, the implementations diverged in multiple forms. Moreover, in some works, the predicted knowledge graph and the annotation have different formats. These data structures and formats are essential because they are dependencies for the evaluation algorithms which help produce metrics on the quality of the generation and explain how good an approach is for the construction of knowledge graphs.

3.4.1 Thesis Standard

To connect the various knowledge graph implementation files, a series of script called converters are being used. This proves to be useful as it allows to transform the dataset between the different implementations. The main data-structure with respect of this thesis is the thesis standard implementation. The utility is two fold, it provides a unifying class to simplify conversions and enables for the usage in this project such as visualisations.

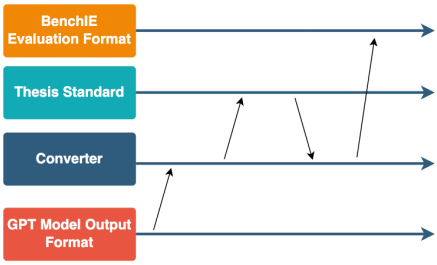

Figure 3.4. The converter scripts help the knowledge graph to be transformed into various formats from outputs to evaluators. The thesis standard minimises the number of converters need by being a 'format hub'

In this case, we symbolise the knowledge graph as a dictionary of two arrays. One of them containing the sentences and the other one the list of triplets. The association between the sentences and list of triplets is based on the index as can be seen below:

```
{
    'text': ["Switzerland's national languages are French, Romansh, Italian, and German. The
    largest city of Switzerland is Zürich."]
    'labels': [[(Switzerland, national languages, French),
                (Switzerland, largest city, Zürich)...], ...]
}
```

3.4.2 Supervised OIE

In this original data-structure for supervised learning *OIE2016* Stanovsky and Dagan [2016], the authors created variable names to denominate tails, heads and relations due to the predeceasing dataset structures with the following labels:

- head: *A0* for 'argument 0'
- relation: *P* for 'predicate'
- tails: *A1... An* for 'argument 1... argument n'

Each extraction is associated with a table in which there is a unique head and relation but may contain multiple tails. Furthermore, this data-structure occupies more space than necessary and limits words to be only part of one tail per head-relation pair and can not contain certain edge cases. To indicate the beginning and continuous inclusion of an argument in terms of words, the labels B and I are post-pended to the label.

Word_id	word	pred	pred_id	head_pred_id	sent_id	run_id	label
0	The	might barred	[9]	9	1	0	O
…	…	…	…	…	…	…	…

0	The	could filed	[13]	13	1	1	O
…	…	…	…	…	…	…	…

Figure 3.5. The tabular format of knowledge graph showing tables with associated labels for each extraction

3.4.3 Re-OIE2016

As a second iteration to the supervised open information extraction, the authors re-labeled the dataset to transform it into *Re-OIE2016* which is formatted as a JSON object Pezoa et al. [2016]. The keys are the sentences and the values are arrays of JSON objects representing extractions.


```
"A Democrat , he became the youngest mayor in Pittsburgh 's history...": [
    {
      "arg0": "he",
      "arg0_index": [3, 3],
      "pred": "became",
      "pred_index": [4, 4],
      "arg1": "the youngest mayor in Pittsburgh 's history",
      "arg1_index": [5, 11],
      "arg2": "at the age of 26",
      "arg2_index": [15, 19],
      "arg3": "",
      "arg3_index": [],
      "loc": "",
      "loc_index": [],
      "temp": "in September 2006",
      "temp_index": [12, 14],
      "context": "",
      "context_index": []
    }
```

3.4.4 BenchIE

In the case of *BenchIE*, the annotation and prediction formats are quite different. As the authors 'exhaustively list all acceptable surface forms of the same fact' Gashteovski et al. [2021] by creating clusters for each underlying fact, each cluster represents a different underlying fact. In the following annotation format, the words in square brackets represent optional terms:

```
sent_id:1 He served as the first Prime Minister of Australia and became a founding...
1--> Cluster 1:
He --> served as --> [the] [first] Prime Minister [of Australia]
He --> served --> as [the] [first] Prime Minister [of Australia]
```

In the predicted format, it is a tabular separated values file with the first column relating to the sentence id, the second entry as the head, the third as the relation and finally the fourth as the tail as such:

```
1 He         served       as the first Prime Minister of Australia
```

3.5 Machine Learning Tasks

3.5.1 Classification

Target variables can either be considered as quantitative or qualitative. A particular case of qualitative data is classification. Typically, for problems where each observation can be associated with one class only, the data is converted into a one-hot encoded vector. The magnitude of this vector is the number of total classes with all entries equaling to 0 apart from the assigned actual class which is equal to 1.

3.5.2 Translation

As natural language processing touches many fields, there are many uncertainties. For instance, the representation of knowledge facts which can have many representations in textual form. This is why the problem of knowledge graph construction should be viewed as a translation problem. Knowledge graphs can be inconsistent as they remain under the influence of the

natural language and it would be evident to see that they can form their own form of language as well. If this language remains too unspecific, then it will be difficult to build consistent and accountable data. However, if the language has a narrow domain on what can be represented then the expressive capabilities of the knowledge are restricted.

3.6 Transformers

3.6.1 Block

The transformer block is a new simple network sub-component which is based solely on attention mechanisms that do not require recurrence or convolutions. Experiments show that this model outperforms previous models in terms of quality, while being more parallelizable and requiring less time to train. To achieve these capabilities, the encoder and decoder each have six layers. The first sub-layer in each layer is a multi-head self-attention mechanism, and the second is a simple position-wise fully connected feed-forward network. The decoder additionally has a third sub-layer which performs multi-head attention over the output of the encoder stack. Vaswani et al. [2017]

3.6.2 Attention Function

Attention is a mechanism used by transformers to focus on specific parts of the input when producing the output. This is done by weighting the importance of each part of the input according to an attention function, which is computed using the query vector and the key vectors. The output is then computed as a weighted sum of the values, with each value being weighted according to its compatibility with the query. Specifically, we can differentiate between two categories of attention.

Scaled Dot-Product Attention

Similar to the function of the dot-product attention introduced in Luong et al. [2015], the scaled variation is presented as such:

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

where Q is the set of queries, K is the keys and V is the values in the shape of a matrix. Thus d_k is the dimension of the keys. Moreover, contrasted to the previous iteration of this function, the "scaled dot-product attention" computes the attention weights for a set of queries, keys and values. It is similar to the dot-product attention function, except that it scales the dot products by $\frac{1}{\sqrt{dk}}$. This helps to counteract the effect of large dot products pushing the softmax into regions where it has small gradients which could help reduce errors in the back-propagation. Vaswani et al. [2017]

Multi-Head Attention

Rather than limiting the function to a single attention head, the authors found it beneficial to linearly project the queries, keys and values h times with various learned linear projections. It allows to perform the attention function in parallel to produce values in the output. These are concatenated and once again projected, resulting in the final valuesVaswani et al. [2017].

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concatenation}\,(\text{head}_1, \ldots \text{head}_\text{h})W^O$$

where head i_i = Attention Function $\left(QW_i^Q, KW_i^K, VW_i^V\right)$ in which the projection matrices parameters are: $W_i^Q \in \mathbb{R}^{d_{\text{madel}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{mode}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{matel}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{madel}}}$.

3.6.3 Language Models

Large language models have shown that large gains can be made on various tasks dealing with natural languages by first generatively pre-training on a diverse corpus of text without annotations and then using discriminative fine tuning on a narrow task. To achieve effective transfer while requiring minimal changes to the model architecture, the proposed approach employs task-aware input transformations during fine-tuning. This is shown to outperform discriminatively trained models that use architectures tailored to each task, significantly outperforming the state of the art Radford et al. [2019].

Generational Pre-Training

To pre-train a standard language modeling objective to maximize the following likelihood given an unsupervised corpus of tokens (sub-components of words): ($\mathscr{U} = \{u_1, \ldots, u_n\}$):

$$L_1(\mathscr{U}) = \sum_i \log P\left(u_i \mid u_{i-k}, \ldots, u_{i-1}; \Theta\right)$$

in which P is models the conditional probability employing transformer based neural network with Θ, its parameters and k being the context window associated with a given input. Radford et al. [2018]

3.6.4 Fine-Tuning Models

When training a machine learning model for natural language processing (NLP), the model must generally handle text in such a way that downstream learning is possible. Thus, following the pre-training, the language model can now be effectively trained on a downstream discriminative task. Employing a supervised dataset where the input sequence is $[x^1, ..., x^m]$ and the matching output is y. The inputs are first passed through the transformer blocks and then finally added to the linear output layer W_y to predict the next token:

$$P\left(y \mid x^1, \ldots, x^m\right) = \text{softmax}\left(h_l^m W_y\right).$$

Following the objective function to be maximised:

$$L_2(\mathscr{C}) = \sum_{(x,y)} \log P\left(y \mid x^1, \ldots, x^m\right)$$

Variations of this fine-tuning approach can also be adapted to further increase performance.

Zero, One, Few Shots Approaches

Fine-tuning, few-shot learning, one-shot learning, and zero-shot learning are the four main methods for prompting or training a large language model on task-specific data. The most common approach is fine-tuning, which involves updating the weights of a pre-trained model by training on a dataset specific to the desired task. This can lead to strong performance on many benchmarks, but it has some drawbacks, such as the possibility of poor generalization out-of-distribution or overfitting to spurious features of the training data. Few shot learning is when the model is given a few demonstrations of the task as conditioning at inference time, but no weight updates are permitted. Contrasted with one shot learning is defined as to when only one demonstration is allowed in the context window. Zero shot learning on the other hand can not contain demonstrations and the model is only given a natural language guidelines. Brown et al. [2020]

Text-to-Text Paradigm

In this paradigm, the problems are guided by the central concept of approaching any text processing problem as a "text-to-text" problem, in which text is entered and new text is output Raffel et al. [2019]. The core benefit lies in that text-to-text architecture allows us to use the same parameters, decoding mechanism, training approach, and loss function to any task we consider.

Researchers make use of this adaptability by evaluating performance on a variety of NLP tasks, including question answering, sentiment categorization, document summarization, and now in our work knowledge graph generation.

Sizes and Performance of Models

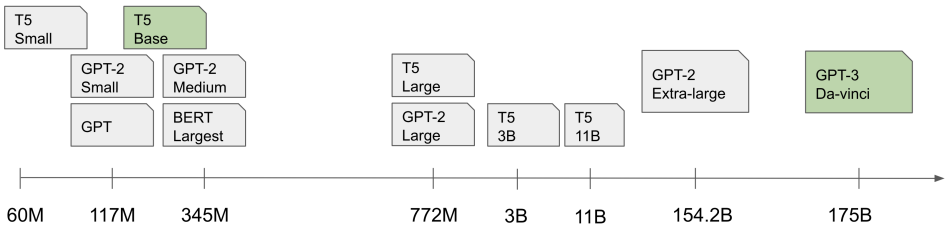

Figure 3.6. Number of parameters in each language model. Scale is not linear for visualisation purposes.

There are three main families of language models which are differentiated by their architectures, training methods and authors. As described in the section of related works.

Chapter 4

Material and Methods

In our study, we aim to use three categories of methods that enable the creation of knowledge graphs; Non-learning approaches (Algorithmic), supervised and unsupervised methods on the BenchIE dataset. The experimental models can be found in the directory, the results in the evaluators directory on the thesis repository.

4.1 Study Plan

We will evaluate performance on the task of knowledge graph creation using various machine learning methods: an algorithmic baseline, supervised, unsupervised (few-shot). In order to avoid additional computation and at the same time increase performance, pre-trained language models are employed. Experimental models will include the large language model translation using *gpt3* Brown et al. [2020] , and text-to-text transformer *T5* Raffel et al. [2019]. For validation purposes, we will use of the a learning baseline system *OpenIE 6* Kolluru et al. [2020a], an algorithmic approach *MinIE* Gashteovski et al. [2017], and a highly performant system *ClauseIE* Del Corro and Gemulla [2013].

4.1.1 Hardware Requirements

To increase the execution speed of the experiments, the neural network trainings and evaluation sessions are implemented using graphics processing units (GPU). In our study, we used NVIDIA T4 TENSOR CORE GPU with 16 GB of memory from Google Research col.

4.2 Experimental Models

Sub-component of Models: Language and Learning Models

Sub-component of Models: To prompt the models with the correct textual syntax and then to retrieve the answer from the text, a series of parsing function needed to be implemented.

Pre-Processor:

The concept of few-shot demonstrates that large language models can use the contextual information to do predictions rather than requiring the update of weights to specific datasets. In our case, we will be giving examples of sentences followed by facts for the language model to fill its context window. Then, we query the fact based on the last sentence inserted in the context window. For example, the query to the model should follow the format of *n* examples of sentence-fact pairs:

```
In the sentence: Bern is the capital of Switzerland
The facts are: (Bern, Capital, Switzerland)
In the sentence: Alphabet is a company headquartered in Mountain View
The facts are: (Alphabet, is a, company), (Alphabet, headquartered, Mountain View)
...
In the sentence:  <the sentence to extract facts from>
```

In general, the better the problem is described in human-like terms, the easier it will be for the large language model to evaluate the performance. Previous to this format, we tried the prefix "Q:", "sentence:" and "A:", "Facts:" for the sentence and triplet list respectively. After running the experiment with syntactically variations, we reported the results for the one that had the best syntax tuning which is the phrasing described above. Moreover, to avoid producing an empty string, a method that was effective was to exclude the last fact prefix prompt "The facts are:". Although it is unclear why the number of valid pairs increased with the exclusion of the final prompt, we hypothesise that the beam search algorithm is affected by the last phrase. As a result, the pre-processing requires four additional text variables to predict beyond the traditional systems. For a practical approach, we used “In the sentence:”, “The facts are”, “),(” and “„” as sentence prefix, triplet list prefix, fact separation token and argument separation token respectively. When the context window is built, it is important to ensure that the model does not exceed the maximum length which can be further asserted with a custom function.

Post-Processors:

For the scope of this thesis, a custom parser is implemented to produce a list of triplets associated with a given sentence by performing a string split among the fact separation and argument separation tokens. As an additional feature, the parser records the number of valid and invalid pairs to generate metrics. In a similar fashion the second step ensures that the words of the fact are present in the sentence. Finally, it outputs the resulting knowledge graph into the thesis standard format.

4.2.1 GPT-3 with Syntax Querying: Few-shot

Learning Parameters

Learning Parameters Unlike traditional models, this one does not require fine-tuning on the specific dataset as it aims to be a universal text predictor. The series of GPT models have been trained to predict the next logical continuation of the text. GPT-3 is a necessary model for comparison as it represents state of the art in natural language processing with its ability to create few-shot behaviours

Predicting Triplets

To run the model, we used OpenAI application point interface while ensuring that the connection does not fail between the local host and their servers. Moreover, to evaluate the performance, it is contextualised once with data from *BenchIE* benchmark and once with *Re-OIE2016* data.

Implementation

The hyper-parameters used to run this model were temperature of 0.95, maximum length of 359, top-p of 1, frequency penalty 0, and presence penalty of 0, number of runs of 3, on the $text-davinci-002$ engine on the API with a stop break of two new line characters (). From a formal perspective, when displaying the number of runs we create the union of multiple of inferences.

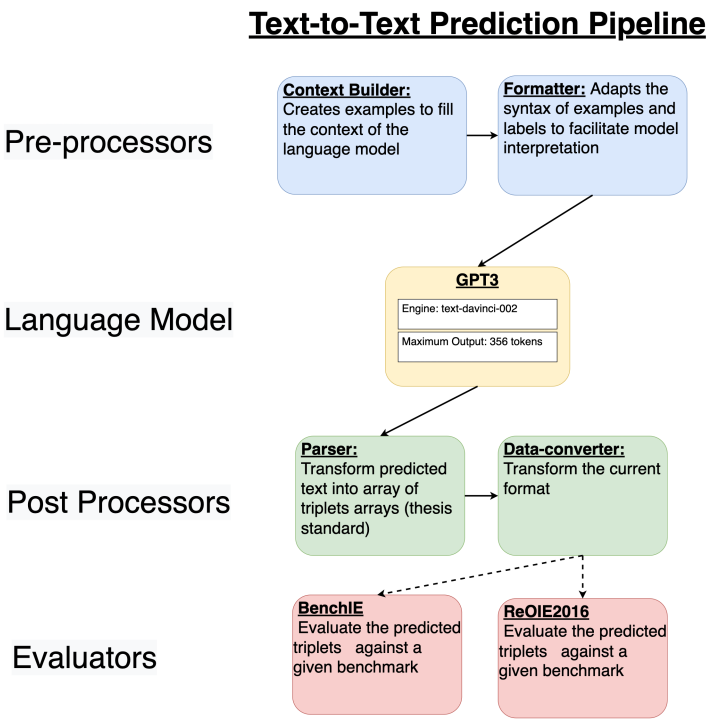

Figure 4.1. Large language model pipeline for text to text knowledge graph generation, GPT-3 demonstrating few-shot behaviour

4.2.2 Text to Text Language Model Translation T5: Fine-Tuning

Learning Parameters

A practical approach to solving the task of knowledge graph generation would be to view it as text-to-text model. By using *T5-Base* to fine-tune on the new text-to-text knowledge graph construction dataset Raffel et al. [2019], we are able to view it as a translation problem. The benefit of this approach lies in the fact that the generation process most likely generalises better as there is the possibility for variable length output.

Predicting Triplets

When predicting the tokens there is a unique tokens beyond the natural language vocabulary. The separation tokens of the triplet list is represented using the $< SEPARATION - TOKEN >$ token which in our case we used $',,'$.

Implementation

The *T5-Base* is fine-tuned with the previously mentioned custom dataset for 40 epochs using a python library Roy. Moreover, the prediction returned a variable number of outputs (3 or 4 depending on the dataset). The beam search used the following parameters number of beam of 5, top p of 1, top k of 50 and a repetition penalty of 1.5.

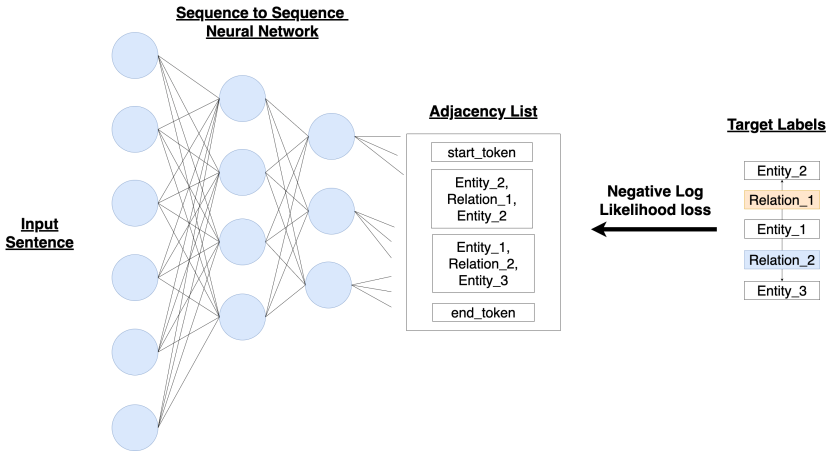

Figure 4.2. Adjacency tensor and the respective dimensions

4.3 Evaluation Methodology

4.3.1 Parsing the outputs

A custom parser is implemented to retrieve the triplet list from the unstructured prediction text converting it back into an array of triplets with text arguments. If there are issues with the syntax of the output, then the parser considers the prediction as invalid.

4.3.2 Evaluation Methodology

Metrics

We evaluated the performance of our model using key metrics which have been modified to the task of text-to-text knowledge graph construction. In the stage of the post-processing of the model's output, we look at how well the model can generate structured predictions which is defined as the *syntax accuracy*. Then in the task of open information extraction since the vocabulary of the prediction must be equal to $V_{prediction} = V_{sentence}$, then an additional accuracy is implemented on the triplet level called the *in sentence accuracy*. Finally, to compare across different models, we use *precision*, *recall*, and *f1* as defined in the *BenchIE* evaluator section in the Background Chapter 2. Furthermore, some traditional natural language processing metrics were excluded for instance the character level precision is not truly relevant as there are many permutations of the correct answers.

4.3.3 BenchIE Benchmark

The dataset is composed of 300 sentences, with 1350 clusters and 9049 possible template permutations that are acceptable into the evaluation algorithm Gashteovski et al. [2021]. Moreover, the outputs are stored as JSON files in the repository. In our evaluations, we will be using the BenchIE format. The benchmark model outputs has a the structure of one extraction per line:

```
SENTENCEID HEAD RELATION TAIL
```

When we train the model using the inputs, we perform a 85% train - 15% test split on the data. Therefore on the BenchIE benchmark, we split the dataset into 260 sentences for training and 40 sentences for testing. The training set is used to train the model and the test set is used to evaluate the performance of the model on the benchmark whilst ensuring no overlap.

Chapter 5

Results

In this chapter , we review the results from generating knowledge graphs using large language models of GPT3 Brown et al. [2020] and T5 Raffel et al. [2019] employing different training methods such as few-shot learning as well as fine-tuning. In details, the following chapter is divided into five sub-divisions. The first two sections include specific results and optimisations and configurations for each of the two tested language models.

Then, we introduce the visualiser, a dashboard to enable qualitative analysis of sentence knowledge graph data. Finally, we create two qualitative studies to understand the performance of the results beyond the numeric limitations of evaluation algorithms. Thus providing deeper insights into the task of knowledge graph generation using advanced machine learning techniques.

Finally, the last section contextualises and clusters the results of the thesis with other works in the field such as *ClausIE* Del Corro and Gemulla [2013].

5.1 GPT-3: Evaluation

As stated in the previous chapter about materials and methods, in our experiments GPT-3 is executed with two different configurations. The first one GPT-3(c=Re) has its context filled with the *RE-OIE2016* dataset whilst the second one has it with a subset of *BenchIE GPT-3(c=Be)*. However, both models are evaluated on the same benchmark although the one with the BenchIE excludes the phrases used in the context to have a more accurate measure for generalisability Gashteovski et al. [2021].

5.1.1 Configuration 1: Input Context with Re-OIE2016

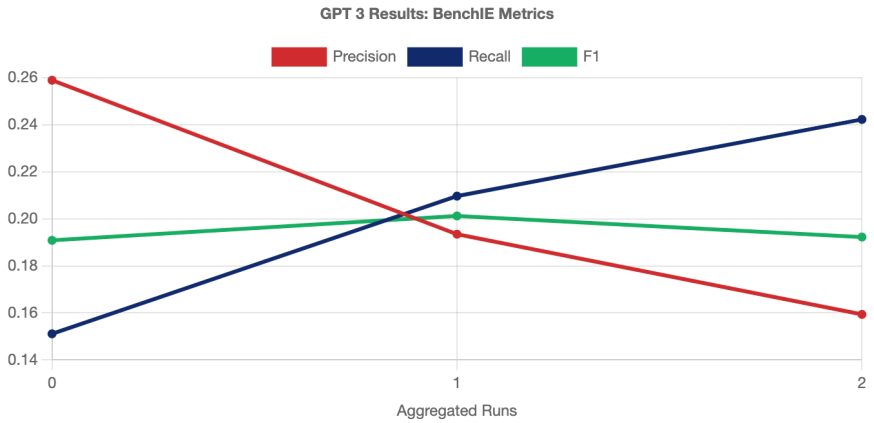

Figure 5.1. Chart depicting GPT-3's performance on the BenchIE benchmark. The key metrics include precision, recall and f1 across the dimensions of aggregated runs.

The model achieves peaks in precision at 0.26, in recall at 0.24 and in f1 at 0.21. The model's performance in terms of recall tends to increase with the dimensions of aggregated runs, whereas the precision decreases as runs accumulate. This relation could perhaps arise due to the fact that there is also an increase in the number of guesses reduces the precision due to the denominator increasing radically.

In the chart above, it becomes evident that the model understands two important properties of knowledge graph constructions. The first one being the suggested data schema of the text for the output. As shown in the chart, the model consistently predicts the structure of the sentence so that our greedy parser is able to extract triplets from 93% of the text predictions. Moreover, from this subset, the model successfully infers the correct vocabulary on 98 % of the text predictions which demonstrates correct understanding of the task of Open Information Extraction.

5.1.2 Configuration 2: Input Context with BenchIE

The model achieves peaks in precision at 0.15, in recall at 0.20 and in f1 at 0.164. The model's performance in terms of recall tends to increase with the dimensions of aggregated runs, whereas the precision decreases as runs accumulate. For a counter-intuitive reasons, there seems to be quite some strong variation for when the context prompt is built with BenchIE vs.

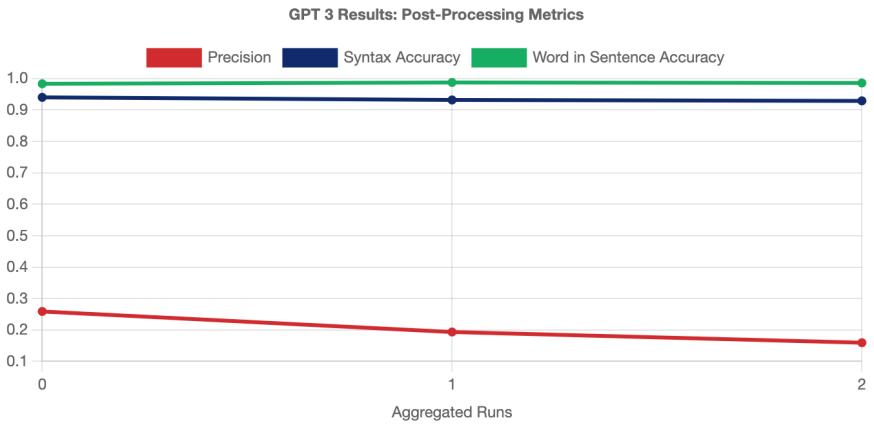

Figure 5.2. Chart depicting GPT-3's ability to generate structured text of knowledge graph triplets. The key metrics include syntax accuracy and word in sentence accuracy.

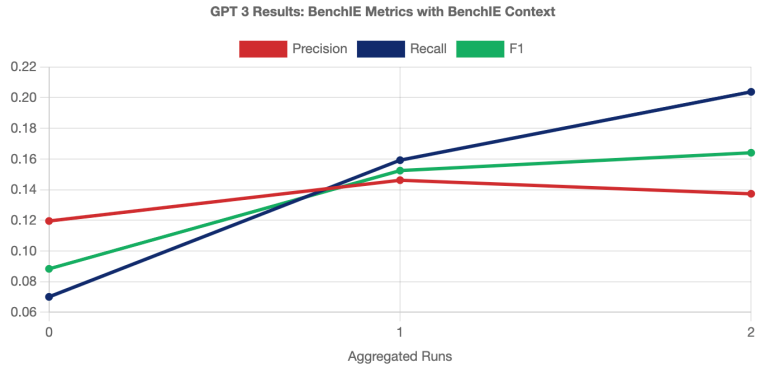

Figure 5.3. Chart depicting GPT-3's performance on the BenchIE benchmark. The key metrics include precision, recall and f1 across the dimensions of aggregated runs.

ReOIE2016 dataset. This pattern seems to be the opposite of what should be expected as the same dataset should have better results as shown with T5 fine-tuning in the following section. Perhaps, this may be due to GPT-3's strong variability in the response.

In the chart, it becomes evident that the model understands two important properties of knowledge graph constructions. The first one being the suggested data schema of the text for the output. As shown in the chart, the model consistently predicts the structure of the sentence so that our greedy parser is able to extract triplets from 97% of the text predictions. Moreover, from this subset, the model successfully infers the correct vocabulary on 98 % of the text predictions which demonstrates correct understanding of the task of Open Information Extraction. Although this is more or less consistent with the previous run, there seems to be medium variance in terms of correct syntax using this specialised dataset.

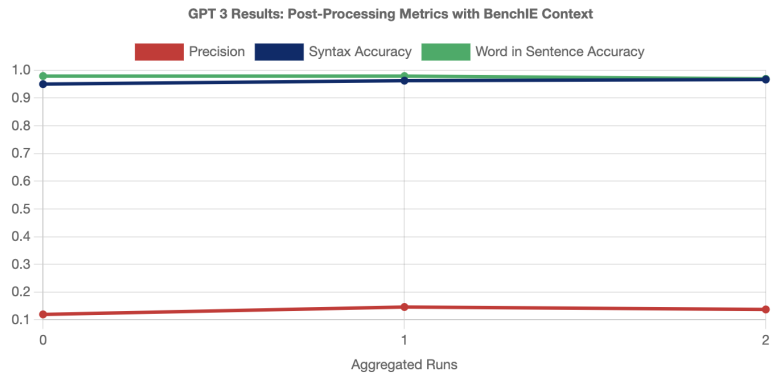

Figure 5.4. Chart depicting GPT-3's ability to generate structured text of knowledge graph triplets. The key metrics include syntax accuracy and word in sentence accuracy.

5.2 T5 Evaluation

5.2.1 Configuration 1: Fine-tuned with Re-OIE2016

The T5 model is able to extract relevant information from the sentences, and produces accurate knowledge graph triplets based on the context. The results further confirm the hypothesis that large language models are capable of generating knowledge graph triplets without the need of a fixed schema. Moreover, the syntactical accuracy reaches .993 and the word in sentence accuracy achieves .965 which adds further evidence that open information extraction can be viewed as a sequence to sequence problem.

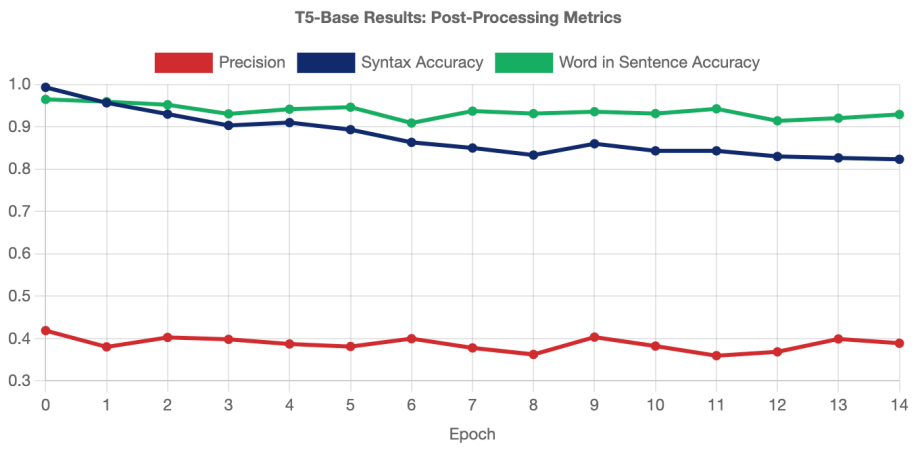

Figure 5.5. Chart depicting T5's ability to generate structured text of knowledge graph triplets. The key metrics include syntax accuracy and word in sentence accuracy. The independent variable is the number of epochs the model is fine-tuned for.

It is important to note that in this experiment, the fine tuned model only contains 1 run and it's context is not primed. The results do not show a clear pattern in terms of precision.

However, the recall seems to be progressively increasing up until epoch 13 which boosts the f1 score from .17 to .25.

To further clarify the relation between number of fine-tuning epochs and number of runs included in the model a 3D grid-search is established to measure the precision, recall and f1 scores respectively. By using the dataset of ReOIE2016 to train and BenchIE to evaluate, the precision, recall and f1 peak at respectively the 3rd epoch with 1 run, 14th epoch with 3 runs and 10th epoch with 2 runs. It appears that precision decreases with the number of epochs whereas recall increases until the 14th epoch. On the contrary, the recall has positive correlation and the precision negative correlation with the number of runs included.

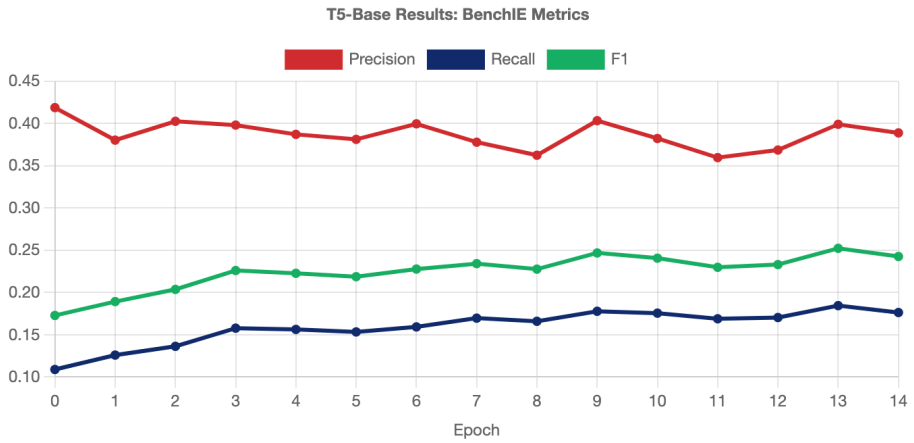

Figure 5.6. Chart depicting the key performance metrics of the T5 base model on the BenchIE benchmark. The independent variable is the number of epochs the model is fine-tuned for.

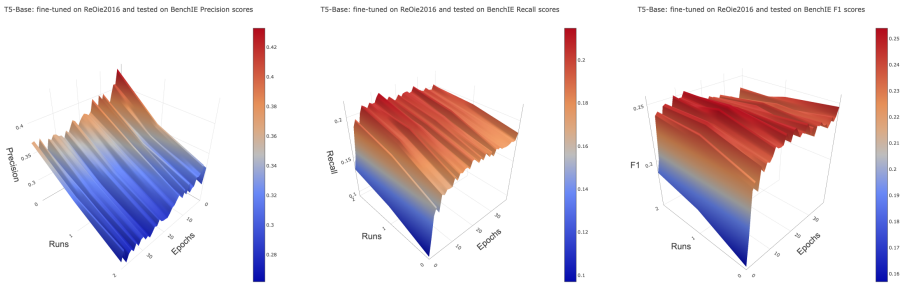

Figure 5.7. T-5 Base Model Fine-Tuned on Re-OIE2016 and evaluated on BenchIE benchmark. The charts show the performance in terms of precision, recall and f1 respectively. The independent variables are numbers of runs on the x axis and number of epochs on the y axis.

5.2.2 Configuration 2: Fine-tuned with BenchIE

As it is clear that the fine-tuned T5 model is capable of predicting structure text with the previous configuration, the task is to optimise for the correct training hyper-parameters. By using the BenchIE benchmark to train and predict, the metrics increase dramatically. The precision, recall and f1 peak at respectively the 12th epoch with 1 run; 30th epoch with 4 runs; at both 17th epoch with 01 runs and 11th epoch with 2 runs . Once again, the recall has positive correlation and the precision negative correlation with the number of runs included. Both the precision and recall in this case benefit from being fine-tuned with the f1 being the two.

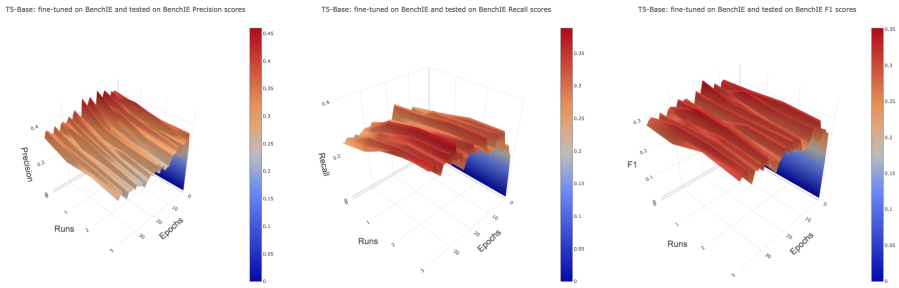

Figure 5.8. T-5 Base Model Fine-Tuned on BenchIE and evaluated on BenchIE benchmark. The charts show the performance in terms of precision, recall and f1 respectively. The independent variables are numbers of runs on the x axis and number of epochs on the y axis.

5.3 Interactive Visualisation Dashboard

With complex tasks in natural language processing, the qualitative analysis sometimes reveals insights beyond the capabilities of quantitative evaluation. A specialised dashboard was implemented to explore specific predicted sentence identifiers for a given experimental model as well as an overview into the current dataset structure. This allows for a qualitative evaluation on a finer-grained level, providing interesting insights into the behaviour of the current system. The dashboard is built using vis.js javascript library and can be hosted on a server as an html file with CDNs which interprets the results from a Javascript Object Notation (json) file. The visualiser can be found in the thesis GitHub repository under the visualiser directory [1].

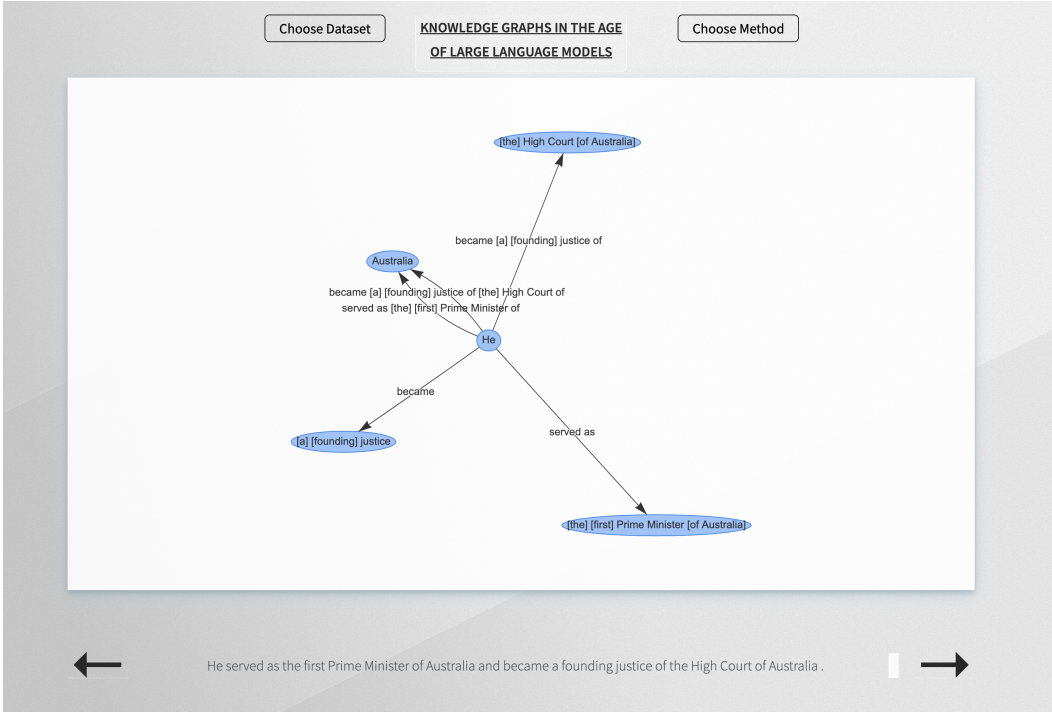

Figure 5.9. Visualiser demonstrating a sentence and its respective knowledge graph originating from labelled datasets or generated by a model

[1] https://github.com/MikeDoes/thesis/tree/main/visualiser

5.3.1 Qualitative Analysis 1: BenchIE prediction using Re-OIE2016 as training set

BenchIE Golden Standard

To enhance our understanding of how the models predict, we are going to explore a specific case in a qualitative manner. For instance, in the benchmark, the following sentence appears:

```
For example, a passenger can fly from Chardon, Neb. to Denver for as
little as $89 to $109, according to prices quoted by the company.
```

Since the *BenchIE* benchmark focuses on being fact-based as opposed to grammatical, the are eight fact synsets (triplets) consisting seven unique relations and nine unique entities Gashteovski et al. [2021]. The central head entity in the sentence is "*[a] passenger*" connecting to seven tails. We notice that the determinant of the passenger [a] is optional in the template. This is of great importance as it shows the advances in the flexibility of the framework which allows it to comprehend multiple variation of the same fact as in this case the determinant is entropical.

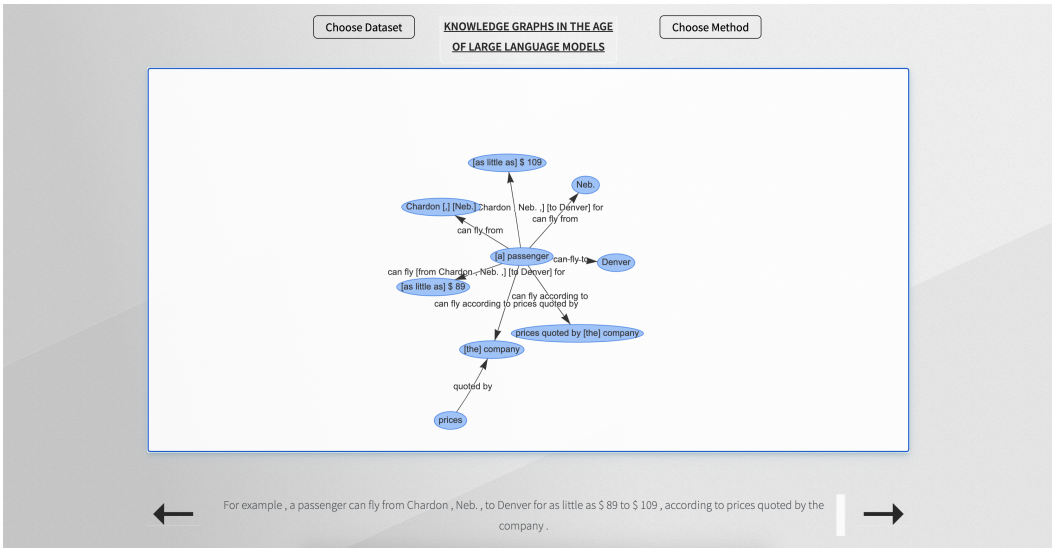

Figure 5.10. Visualiser demonstrating a sentence and its respective knowledge graph in the dataset of BenchIE

However, when we look at the numerical facts extracted from the sentence, we noticed that there are two separate facts for the price which is presented as a range "$89 to $109". In this approach, the interpretation can be ambiguous. Primarily, the reader can be lead to believe that the price of the ticket ranges from minimum $89 to maximum $109 which is little in the cost. On the other side, the reader could be led to believe that the minimum price could be either $89 or $109 depending on an external factor provided by the company. The issue with this interpretation is that the facts can either be separated as a single or as a dual fact. By creating a dataset with only the dual variation of the fact could diminish the evaluation of the models.

As a second note regarding the facts relating to the company entity. We notice that the fact of prices being quoted by the company appears in three separate triplet. Specifically, these synsets are:

```
prices --> quoted by --> [the] company
[a] passenger --> can fly according to --> prices quoted by [the] company
[a] passenger --> can fly according to prices quoted by --> [the] company
```

It becomes evident, that the facts and entities are repetitive and thus skewing the quantitative metrics which need to become more consistent. Moreover, although guidelines are provided, the variations available in the knowledge graph representation seem to still vary as there are multiple ways to phrase facts. Although using multiple exhaustive variations of the similar fact synsets is the best approach that researchers have come across in terms of evaluating knowledge graphs, there is still risk that some data may have ambiguous interpretations.

T5 Prediction with BenchIE fine-tuning

In the following dashboard, we notice T5 doing a prediction which contains five entity nodes connected with all the same relation. It is worthy to note that these entities are much more concise as demonstrated by the length of words. Moreover, it is clear that the neural network demonstrates some understanding in the differentiation of the various entities as each one represents something meaningful in the sentence. To further improve the results, it would be a good direction to increase the frequency or repetition penalty as it might mitigate the risk of having the same relation repeated.

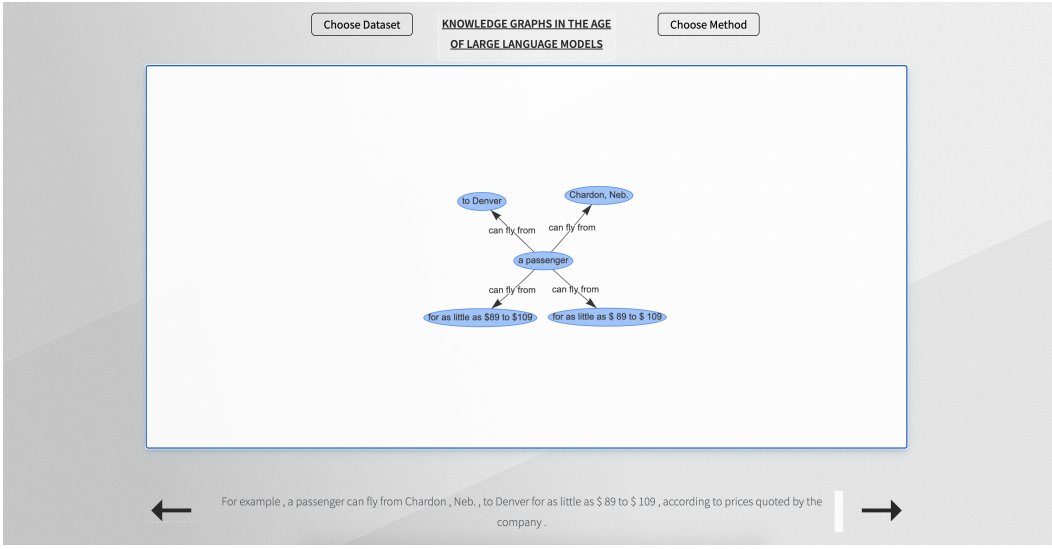

Figure 5.11. Visualiser demonstrating a sentence and the knowledge graph prediction by the fine-tunned T5 model on BenchIE training dataset

Going back to our previous qualitative analysis, the fine-tuned T5 model Raffel et al. [2019] interprets the "as little as $89 to $109" as one entity as opposed to the gold standard dataset which views it as two separate price entities. Perhaps, in future iterations of the experiment, it would be possible to limit the vocabulary to help straighten these ambiguities and improve multi-relational understanding in concise and clear manner for downstream applications of knowledge graphs.

GPT-3 Prediction with BenchIE context

GPT-3 employed an approach without fine-tuning and learned the knowledge graph text-to-text construction pattern simply with a few example in its context window. In this case, we notice that the graph contains six entity nodes connect via four relations meaning that the results are built of two separate graph components.

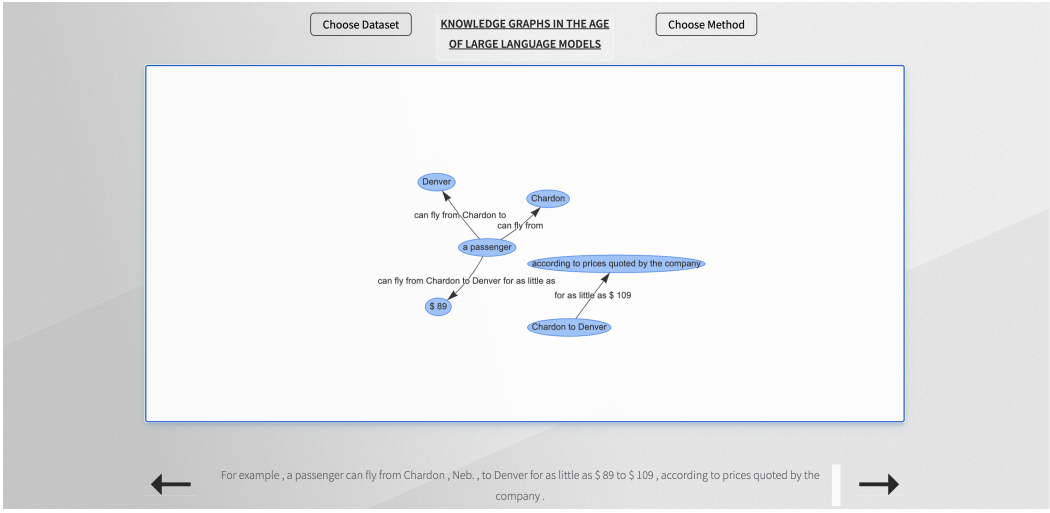

Figure 5.12. Visualiser demonstrating a sentence and the knowledge graph prediction by the fine-tunned GPT3 model

Although this can occur naturally in the prediction, the specific instance is due to the union of two forward runs. It can be seen that the first component seems to have straight forward and minimalist entities with more verbose relations. On the other hand, the secondary component appears to have more verbose entities and relations. Depending on the style of how the golden knowledge graph is constructed, there can be changes in how it will be evaluated for metrics such as accuracy, f1 and recall which shows the difficulty of evaluating the quality of a knowledge graph.

5.3.2 Qualitative Analysis 2: BenchIE test prediction using BenchIE as training set

To have a better grasp of how the models creates prediction, we will investigate a specific scenario in a qualitative approach. In the benchmark, for example, the following statement appears:

```
Mr. Mulford said reports of tension between the Treasury and Fed have been exaggerated ,
insisting that they involved `` nuances . ''
```

BenchIE Golden Standard

In the annotations, the knowledge graph is displayed using two graph components. The first one containing two entities connected via a single relation and the second component with three entities connected with two relations. However, it is clear that there is a slight mismatch as there are two of the nodes representing the same entity but this is due to the fact that there are many templates and that the visualiser needs to take not the first fact from the BenchIE cluster, but rather the ones that minimise the number of entities in the chart.

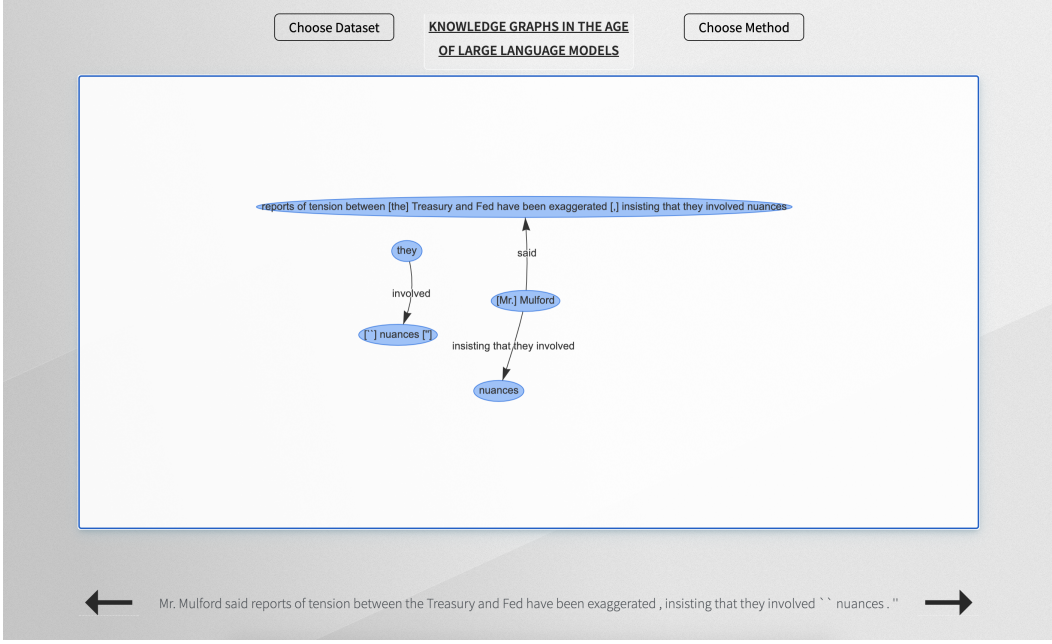

Figure 5.13. Visualiser demonstrating a sentence and its respective knowledge graph in the dataset of BenchIE

GPT-3 Prediction with BenchIE context

In the aforementioned run, GPT-3 generates two sets of facts similar to the annotation. The first set of facts being about what 'Mr. Mulford said' and the second about the 'nuances' which is very similar to what the benchmark has as the golden triplet. Regarding the first knowledge graph component, the part of nuances is not included in the prediction although it is still accepted in thanks to templating aspect of *BenchIE*.

For the secondary graph component, stating the involvement of 'nuances' we can see that there is multiple ways within the sentence to refer to the same entity. According to the benchmark however, only one of them is accepted for estimating the precision metric.

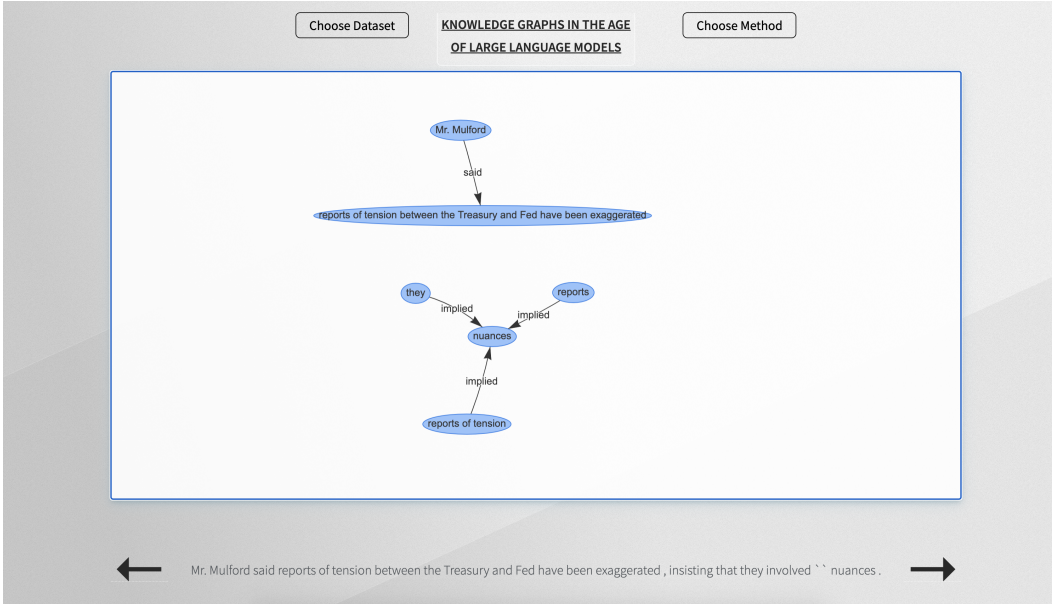

Figure 5.14. Visualiser demonstrating a sentence and the knowledge graph prediction by the fine-tunned GPT3 model with BenchIE as a context filler

GPT-3 Prediction with Re-OIE2016 context

In this prediction, only a single graph component is predicted. Three of the four relations focus on the word 'said' and the fourth one on insisting. Two of them focus on what was said and the other two on the nuances regarding the tensions. In this interpretation of the facts, we see that it is much more difficult to understand the situation using the knowledge graph thus the task should perhaps pre-filter on given sentences that are facts rather than quotations. For instance, the question of objectivity arises again. In other words, by including the information that the reports were exaggerated might violate the freedom from bias as these statements could be viewed as objective which makes it confusing for the language model. Moreover, it appears that the predicted triplets are much more verbose with the context being filled using *RE-OIE2016* as contrasted with the subset of *BenchIE*. This can be proved quantitatively as the average word lengths in a given entity are 2.83 and 5.6 with *BenchIE* and *RE-OIE2016* respectively.

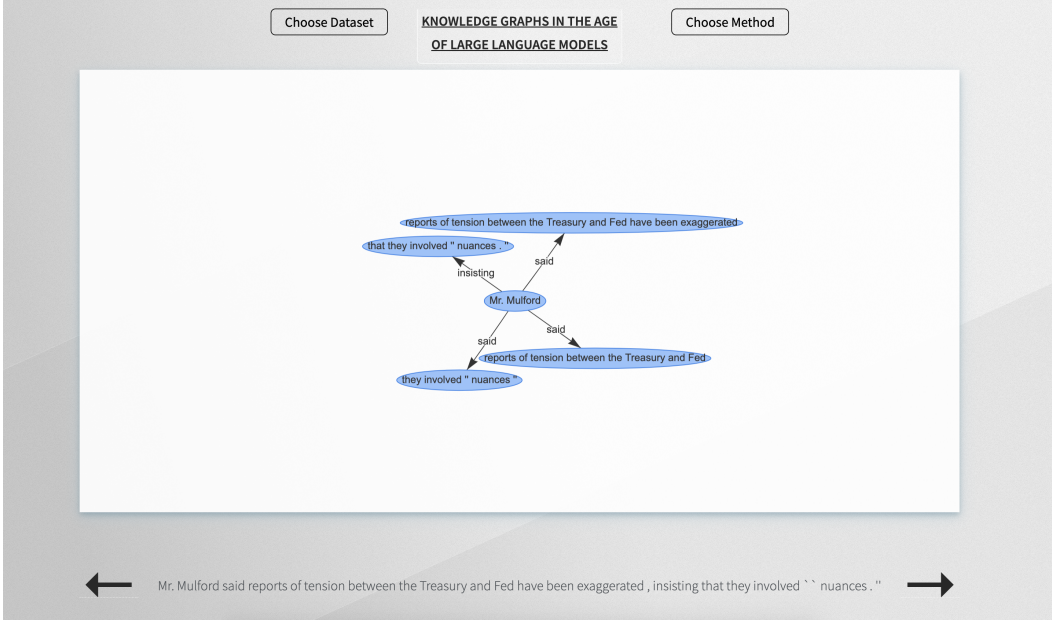

Figure 5.15. Visualiser demonstrating a sentence and the knowledge graph prediction by the fine-tunned GPT3 model with reoie2016 as a context filler

T5 Prediction with BenchIE fine-tuning

Although the T5 predictions scored the best in terms of metrics, for this given sentence, they under-perform. Despite the main two entities of 'nuances' and 'Mr. Mulford' are extracted, the relations include to many words which renders them lengthy with some of them appearing to be as long as 14 words. Moreover, it appears that the model is confused as to which facts to extract and tries to come to terms by increasing the words in the entities and relations.

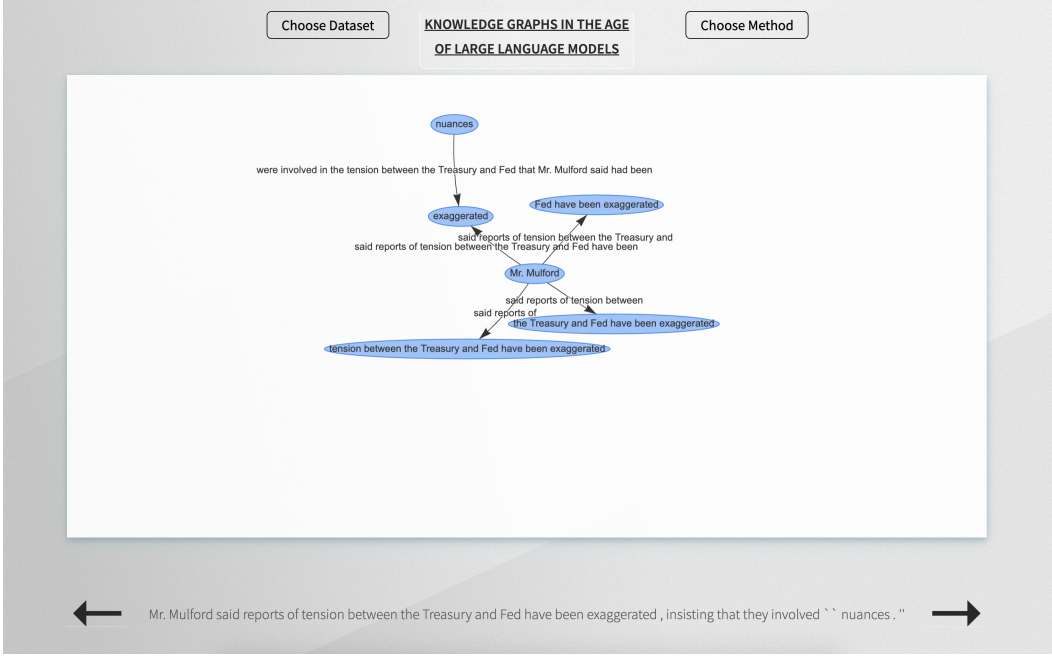

Figure 5.16. Visualiser demonstrating a sentence and the knowledge graph prediction by the fine-tunned T5 model with BenchIE training set

5.4 Summary

In the first set of results, the hypothesis of large language models being capable of predicting the correct syntax is shown to have strong supportive evidence. With the syntax and word accuracies of both language models always being above 94%. Moreover, despite to the lack of high quality data originating from the same source, the training set for the fine-tuning of the results matches the performance of some of the most recent OpenIE6 extractor in terms of f1 and some configuration outperform it in precision and recall. However, the well established algorithmic approach of ClauseIE still outperforms the large language model.

Model	Precision	Recall	F1	Syntax Accuracy	Word Accuracy
EXPERIMENTAL					
GPT3	*0.262*	*0.242*	*0.213*	**0.986**	*0.94*
T5−Base	*0.432*	*0.215*	*0.254*	*0.965*	**0.993**
CONTROL					
ClauseIE	**0.503**	**0.256**	**0.339**	-	-
MinIE	*0.429*	*0.277*	*0.337*	-	-
OpenIE6	*0.311*	*0.214*	*0.254*	-	-

Table 5.1. Performance of the best models runs for the task of knowledge graph generation for the entirety of the BenchIE dataset. The context of GPT3 and the fine-tuning dataset of T5 filled with ReOIE2016 triplets.

On the second set of experiments, the fine-tuning and context include subsets of the same dataset whilst making sure that there is no overlap between training and testing dataset. With the training and testing dataset originating from the same place, the results for the task of knowledge graph construction using large language models produces highly promising results. For instance, the T5-Base model performs the best out of all selected models in terms of recall but fails to produce the best recall and thus f1 scores as the algorithmic approach still displays the best overall performance.

Model	Precision	Recall	F1
EXPERIMENTAL			
GPT3(c=Re)	*0.167*	*0.185*	*0.175*
T5(f=Re)	*0.296*	*0.152*	*0.202*
GPT3(c=Be)	*0.146*	*0.204*	*0.137*
T5(f=Be)	*0.460*	**0.389**	*0.351*
CONTROL			
ClauseIE	**0.579**	*0.350*	**0.437**
MinIE	*0.404*	*0.293*	*0.339*
OpenIE6	*0.319*	*0.229*	*0.267*

Table 5.2. Performance of the best models runs for the task of knowledge graph generation for the last 40 sentences BenchIE dataset. The context (c=) of GPT3 and the fine-tuning (f=) dataset of T5 filled with ReOIE2016 (Re) or non-testing BenchIE (Be) dataset.

Finally, we extend the knowledge graph construction task to also be analysed in a qualitative manner using the dashboard. By displaying both the sentence and the knowledge graph at the same time. Using this approach, we determined various peculiarities in both the model predictions and datasets.

Chapter 6

Discussion

To conclude the thesis, we would like to review the main contributions, limitations and possible extension for further research work.

6.1 Contribution

In this work, we aimed to progress the research of the creation of knowledge graphs using language models in two practical implementations and four different configurations.

The first aspect revolves around creating the preliminary tools to be able to visualise and convert various datasets for knowledge graphs. At this end, we created a standard data-structure which has converter function from and to the standard and towards the specific formats enabling transformations across formats. This data-structure format hub minimises the number of transformations required for a knowledge graph to be transformed to a given format employed in modern literature. Moreover, to gain a deeper qualitative understanding of the data, a web-based knowledge graph application was created to visualise triplet lists with their corresponding sentences accessible via modern browsers wether it be on mobile or desktop devices. This visualisers can help understand the quality of original datasets, model predictions, pre and post-processing steps.

In the second part of the contribution, we aimed to use a large language model to generate knowledge graphs. In this manner, we implemented two custom pipelines to best leverage the potential of language. The first type of model focuses on GPT-3's ability to predict using few-shot behaviour Brown et al. [2020] by providing it a few example then letting it fill the prediction. This approach is almost fully unsupervised as the required training data is substantially low (below 10 examples)

The second type of model focuses on the fine-tuning abilities of the T5-Base model which requires much more high quality supervised data Raffel et al. [2019].

Although both showed high accuracies in terms of following the task syntax, the fine-tuned model shows promising competitive results in generating knowledge graphs for the most semantic open information extraction benchmark, *BenchIE* Gashteovski et al. [2021].

6.2 Limitations

Despite the best of our efforts, it is evident that as with all scientific endeavours, there are limitation in our study. For instance, due to the lack of hardware and time capacity, we used the base only size of the model. The two larger sizes of the models for the fine-tuning of T5 were not implemented which has strong potential to achieve competitive results including state of the art on the benchmark of *BenchIE*.

6.3 Further Work

In future research, it would be great to develop more advanced parsers, analytics, shifting vocabularies among the language models which generate knowledge graphs.

Specifically, advanced parsers would enable to develop higher accuracies in the benchmarks. Currently the pre-processing parser creates a input text given a template. Despite informal internal studies, it would be to do an analysis on what kind of sentence prefix, triplet list prefix, fact separation token and argument separation token maximise the clarity of the model.

As knowledge graph generation is a composite task, many more metrics could be extracted beyond syntax metrics, accuracy, f1 and recall. To further increase analytics, it would be important to see wether the model has learned to generate triplets in a logical order.

Finally, another benefit of the approach introduced in thesis is that it would be possible to adapt the vocabulary towards a specific downstream application rather than the source text. Moreover, the model would be able to extract meaningful triplets from the text which are not part of the training set enabling it to merge the sentence knowledge graph with a larger knowledge graph structure. This could have much larger implications in the field of search engines.

For instance, it could create a knowledge graph vocabulary custom to each user enabling the learning and communication experience to be of much higher quality. Namely, as Google search engine application employed knowledge graphs 400 billion times just in 2016 Goo.

6.4 Conclusion

In this work, we have proposed a novel model for Open Information Extraction from unstructured text. Our model leverages the advantages of a text-to-text model, namely the ability to use large language models with or without fine-tuning, while also being capable of learning the structure of the sentence. We have evaluated our model on a standard OIE dataset, BenchIE, demonstrating competitive results and a strong potential for better generalization. We believe that the proposed model can serve as a valuable tool in building knowledge graphs from text.

Bibliography

Michele Banko, Michael Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. pages 2670–2676, 01 2007.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. CaRB: A crowdsourced benchmark for open IE. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6262–6267, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1651. URL https://aclanthology.org/D19-1651.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL https://arxiv.org/abs/2005.14165.

Janara Christensen, Stephen Soderland, and Oren Etzioni. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, pages 113–120, 2011.

Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *CoRR*, abs/2011.01103, 2020. URL https://arxiv.org/abs/2011.01103.

Danilo Dessì, Francesco Osborne, Diego Reforgiato Recupero, Davide Buscaldi, and Enrico Motta. Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116:253–264, 2021.

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

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110, 2004.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134, 06 2005. doi: 10.1016/j.artint.2005.03.001.

Luciano Floridi and Barry Smith. The blackwell guide to the philosophy of computing and information, chapter 11: Ontology. 2004.

Niklas Friedrich, Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Mathias Niepert, and Goran Glavaš. Annie: An annotation platform for constructing complete open information extraction benchmark. *arXiv preprint arXiv:2109.07464*, 2021.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. Minie: minimizing facts in open information extraction. Association for Computational Linguistics, 2017.

Kiril Gashteovski, Mingying Yu, Bhushan Kotnis, Carolin Lawrence, Goran Glavas, and Mathias Niepert. Benchie: Open information extraction evaluation based on facts, not tokens. *arXiv preprint arXiv:2109.06850*, 2021.

Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, 1992.

Tom Hope, Aida Amini, David Wadden, Madeleine van Zuylen, Sravanthi Parasa, Eric Horvitz, Daniel Weld, Roy Schwartz, and Hannaneh Hajishirzi. Extracting a knowledge base of mechanisms from covid-19 papers. *arXiv preprint arXiv:2010.03824*, 2020.

Keshav Kolluru, Vaibhav Adlakha, Samarth Aggarwal, Soumen Chakrabarti, et al. Openie6: Iterative grid labeling and coordination analysis for open information extraction. *arXiv preprint arXiv:2010.03147*, 2020a.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathore, Soumen Chakrabarti, et al. Imojie: Iterative memory-based joint open information extraction. *arXiv preprint arXiv:2005.08178*, 2020b.

Elizabeth D Liddy. Natural language processing. 2001.

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

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

William Léchelle, Fabrizio Gotti, and Philippe Langlais. Wire57 : A fine-grained benchmark for open information extraction. 09 2018.

Harinder Pal et al. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, 2016.

Yannis Papanikolaou and Andrea Pierleoni. DARE: data augmented relation extraction with GPT-2. *CoRR*, abs/2004.13845, 2020. URL https://arxiv.org/abs/2004.13845.

Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273. International World Wide Web Conferences Steering Committee, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Youngbin Ro, Yukyung Lee, and Pilsung Kang. MultiΘ͡2 oie: Multilingual open information extraction based on multi-head attention with bert. *arXiv preprint arXiv:2009.08128*, 2020.

Shivanand Roy. Simple t5. URL https://pypi.org/project/simplet5/.

Swarnadeep Saha, Harinder Pal, et al. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323, 2017.

Swarnadeep Saha et al. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299, 2018.

Jacob Solawetz and Stefan Larson. Lsoie: a large-scale dataset for supervised open information extraction. *arXiv preprint arXiv:2101.11177*, 2021.

Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1252. URL https://aclanthology.org/D16-1252.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, page (to appear), New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. doi: 10.48550/ARXIV.1706.03762. URL https://arxiv.org/abs/1706.03762.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. Zero-shot information extraction as a unified text-to-triple translation. *CoRR*, abs/2109.11171, 2021a. URL https://arxiv.org/abs/2109.11171.

Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs. *CoRR*, abs/2010.11967, 2021b. URL https://arxiv.org/abs/2010.11967.

Junlang Zhan and Hai Zhao. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9523–9530, 2020.