

# **Advancing Knowledge-Enhanced Conversational Systems Leveraging Language Models**

Dissertation  
zur  
Erlangung des Doktorgrades (Dr. rer. nat.)  
der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der  
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von  
**Md Rashad Al Hasan Rony**  
aus  
Kushtia, Khulna, Bangladesh

Bonn 2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen  
Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Jens Lehmann  
2. Gutachter: Prof. Dr. Stefan Wrobel  
Tag der Promotion: 05.09.2023  
Erscheinungsjahr: 2023

# Abstract

---

Large language models empowering recent conversational systems such as Alexa and Siri require external knowledge to generate informative and accurate dialogues. The knowledge may be provided in structured or unstructured forms, such as knowledge graphs, documents, and databases. Typically, language models face several issues when attempting to incorporate knowledge for conversational question answering: 1) they are unable to capture the relationship between facts in a structured knowledge, 2) they lack the capability of handling the dynamic knowledge in a multi-domain conversational setting, 3) because of the scarcity of unsupervised approaches for question answer over knowledge graphs (KGQA), systems often require a large amount of training data, and 4) because of the complexities and dependencies involved in the KGQA process it is difficult to generate a formal query for question answering. All of these issues result in uninformative and incorrect answers. Furthermore, an evaluation metric that can capture various aspects of the system response, such as semantic, syntactic, and grammatical acceptability, is necessary to ensure the quality of such conversational question answering systems.

Addressing the shortcomings in this thesis, we propose techniques for incorporating structured and unstructured knowledge into pre-trained language models to improve conversational question answering systems. First, we propose a novel task-oriented dialogue system that introduces a structure-aware knowledge embedding and knowledge graph-weighted attention masking strategies to facilitate a language model in selecting relevant facts from a KG for informative dialogue generation. Experiment results on the benchmark datasets demonstrate significant improvement over previous baselines. Next, we introduce an unsupervised KGQA system, leveraging several pre-trained language models to improve the essential components (i.e., entity and relation linking) of KGQA. The system further introduces a novel tree-based algorithm for extracting the answer entities from a KG. The proposed techniques relax the need for training data to improve KGQA performance. Then, we introduce a generative system that combines the benefits of end-to-end and modular systems and leverages a GPT-2 language model to learn graph-specific information (i.e., entities and relations) in its parameters to generate SPARQL query for extracting answer entities from a KG. The proposed system encodes linguistic features of a question to understand complex question patterns for generating accurate SPARQL queries. Afterward, we developed a system demonstrator for question answering over unstructured documents about climate change. Pre-trained language models are leveraged to index unstructured text documents into a dense space for document retrieval and question answering. Finally, we propose an automatic evaluation metric, incorporating several core aspects of natural language understanding (language competence, syntactic and semantic variation). A comprehensive evaluation exhibits the effectiveness of our proposed metric over the state-of-the-art approaches. Overall, our contributions exhibit that the effective incorporation of external knowledge into a language model significantly improves the performance of conversational question answering. We made all the resources and code used in the proposed systems publicly available.



# Acknowledgements

---

Words cannot express my gratitude to Prof. Dr. Jens Lehmann for providing me with the opportunity to conduct research under his supervision. I would like to express my deepest appreciation to my supervisor Prof. Dr. Jens Lehmann and reviewer Prof. Dr. Stefan Wrobel, for their time and effort in providing valuable feedback and reviews. I am thankful to Prof. Dr. Jens Lehmann and Prof. Dr. Ricardo Usbeck for their continuous support throughout the time-span of this thesis which made the thesis journey smooth and comments which improved the writing. I would like to extend my thank to the Ph.D. commission members for putting their time and effort in assessing and completing the formal procedures involved in this process.

I am grateful to the Fraunhofer EIS department, where I had the opportunity to work on research projects as a master's student. This opportunity inspired me further to explore and conduct research in the direction of conversational AI. Especially, I would like to thank Dr. Debanjan Chaudhuri for providing me the opportunity to work with him on research papers during my master's study. That helped me build a strong foundation and made me aware of the research scopes of conversational AI. I also want to thank Dr. Debanjan Chaudhuri for providing constant guidance during the first year of my Ph.D. journey. Furthermore, I am thankful to my colleagues at Smart Data Analytics (SDA) and Fraunhofer Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme (IAIS) for their support. I would like to extend my thank to Ying Zuo, Roman Teucher, Uttam Kumar, and Liubov Kovriguina for collaborating on two papers contributing to my thesis.

Finally, I would love to express my profound gratitude to my parents (Md Hasanur Rahman and Rogina Akter) and wife, Samrin Priya. From the beginning of my work, I had their endless support with me. They believed in me and encouraged me throughout this challenging journey. It would not be possible for me to reach this far without their constant support and prayer. I dedicate this thesis to my parents, wife, and our son Zaviyar Rashad.



# Contents

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation and Challenges . . . . .	4
1.2	Research Questions . . . . .	5
1.3	Thesis Overview . . . . .	7
1.3.1	Contributions . . . . .	7
1.3.2	Publications . . . . .	9
1.3.3	Author Contributions . . . . .	10
1.3.4	Thesis Structure . . . . .	10
<b>2</b>	<b>Background Knowledge</b>	<b>11</b>
2.1	Language Models . . . . .	11
2.1.1	Static Word Embedding Models . . . . .	11
2.1.2	Recurrent Models . . . . .	12
2.1.3	Attention-based Models . . . . .	13
2.2	Knowledge Graph . . . . .	17
2.3	Conversational Systems . . . . .	18
2.3.1	Dialogue Systems . . . . .	18
2.3.2	Question Answering Over Knowledge Graphs . . . . .	20
2.3.3	Machine Reading Comprehension . . . . .	22
2.4	Evaluation Metrics . . . . .	25
<b>3</b>	<b>Related Work</b>	<b>31</b>
3.1	Dialogue Systems . . . . .	31
3.1.1	Task-oriented Dialogue Systems . . . . .	31
3.1.2	Non-task-oriented Dialogue Systems . . . . .	32
3.2	Question Answering Over Knowledge Graphs . . . . .	33
3.2.1	Entity Linking . . . . .	33
3.2.2	Relation Linking . . . . .	34
3.2.3	Answer Extraction . . . . .	34
3.3	Machine Reading Comprehension . . . . .	36
3.3.1	Document Retriever . . . . .	36
3.3.2	Document Reader . . . . .	37
3.3.3	MRC Dataset . . . . .	37
3.4	Evaluation of Generative Systems . . . . .	38

<b>4</b>	<b>Generative Dialogue Systems With Structured Knowledge</b>	<b>39</b>
4.1	Introduction . . . . .	39
4.2	Problem Definition . . . . .	42
4.3	Approach: DialoKG . . . . .	42
4.3.1	Knowledge and Dialogue Embedding . . . . .	42
4.3.2	Knowledge Attention Mask Construction . . . . .	43
4.3.3	Decoder . . . . .	45
4.4	Experimental Setup . . . . .	45
4.4.1	Data . . . . .	45
4.4.2	Hyper-parameter Settings . . . . .	46
4.4.3	Evaluation Metrics . . . . .	47
4.4.4	Baselines . . . . .	47
4.5	Results . . . . .	48
4.5.1	Quantitative Results . . . . .	48
4.5.2	Qualitative Results . . . . .	48
4.6	Analysis . . . . .	49
4.6.1	Ablation Study . . . . .	50
4.6.2	Effectiveness of Knowledge Embedding . . . . .	50
4.6.3	Impact of Knowledge Attention Mask . . . . .	51
4.6.4	Case Study . . . . .	51
4.6.5	Influence of Dialogue History . . . . .	52
4.7	Summary . . . . .	52
<b>5</b>	<b>Unsupervised Question Answering Over Knowledge Graphs</b>	<b>55</b>
5.1	Introduction . . . . .	55
5.2	Problem Definition . . . . .	57
5.3	Approach: Tree-KGQA . . . . .	58
5.3.1	Entity Linking . . . . .	58
5.3.2	Zero-shot Relation Linking . . . . .	60
5.3.3	Answer Entity Extraction . . . . .	61
5.4	Experiments and Results . . . . .	64
5.4.1	Data . . . . .	64
5.4.2	Experimental Setup . . . . .	64
5.4.3	Baselines . . . . .	65
5.4.4	Results . . . . .	65
5.5	Analysis . . . . .	67
5.5.1	Ablation Study . . . . .	67
5.5.2	Case Study . . . . .	68
5.5.3	Error Analysis and Limitations . . . . .	69
5.5.4	Discussion . . . . .	70
5.6	Summary . . . . .	71
<b>6</b>	<b>SPARQL Query Generation: A Generative Approach</b>	<b>73</b>
6.1	Introduction . . . . .	74



6.2	Approach: SGPT . . . . .	76
6.2.1	Problem Definition . . . . .	76
6.2.2	Encoding . . . . .	76
6.2.3	Decoding . . . . .	78
6.3	Experiments and Results . . . . .	80
6.3.1	Data . . . . .	80
6.3.2	Training Settings . . . . .	81
6.3.3	Evaluation Metrics . . . . .	83
6.3.4	Baselines . . . . .	83
6.3.5	Quantitative Results . . . . .	84
6.3.6	Qualitative Results . . . . .	85
6.4	Analysis . . . . .	85
6.4.1	Ablation Study . . . . .	85
6.4.2	Case Study . . . . .	86
6.4.3	Effectiveness of Entity Masking Strategy . . . . .	86
6.4.4	Effective Entity and Relation Generation . . . . .	87
6.4.5	Error Analysis and Limitations . . . . .	87
6.5	Summary . . . . .	88
<b>7</b>	<b>Question Answering Over Unstructured Knowledge</b>	<b>89</b>
7.1	Introduction . . . . .	89
7.2	Climate Bot System . . . . .	90
7.2.1	Retriever. . . . .	90
7.2.2	Reader. . . . .	91
7.2.3	User Interface (UI). . . . .	91
7.3	CCMRC: Climate Change Dataset . . . . .	93
7.3.1	Data Sources and Acquisition . . . . .	93
7.3.2	Data Annotation . . . . .	93
7.3.3	Dataset Statistics . . . . .	94
7.4	Evaluation . . . . .	94
7.5	Summary . . . . .	95
<b>8</b>	<b>Dialogue System Evaluation</b>	<b>97</b>
8.1	Introduction . . . . .	97
8.2	Approach: RoMe . . . . .	99
8.2.1	Earth Mover’s Distance Based Semantic Similarity . . . . .	99
8.2.2	Semantically Enhanced TED . . . . .	101
8.2.3	Grammatical Acceptability Classification . . . . .	102
8.2.4	Final Scorer Network . . . . .	103
8.3	Experiments and Results . . . . .	103
8.3.1	Data . . . . .	103
8.3.2	Hyper-parameter Settings . . . . .	104
8.3.3	Baselines . . . . .	105
8.3.4	Results . . . . .	105
8.3.5	Ablation Study . . . . .	107

8.3.6	Qualitative Analysis . . . . .	108
8.4	Robustness Analysis . . . . .	108
8.5	Summary . . . . .	109
<b>9</b>	<b>Conclusion and Future Directions</b>	<b>111</b>
9.1	Conclusion . . . . .	111
9.2	Future Directions . . . . .	112
<b>10</b>	<b>List of Publications</b>	<b>115</b>
	<b>Bibliography</b>	<b>117</b>
	<b>List of Figures</b>	<b>149</b>
	<b>List of Tables</b>	<b>153</b>

## Introduction

---

Conversational systems have recently gained increased attention due to the unforeseen advancements of deep learning techniques. A conversational system typically engages in interaction with other human or computer participant(s) in the form of speech, text, or sign<sup>1</sup>. Depending on the application, the interactions usually span across multiple turns (i.e., dialogue systems) or single turn question answering (i.e., machine reading comprehension). Almost all the smart devices that we use in our daily life (i.e., laptops and smartphones) are generally equipped with a conversational system, also known as voice assistant service (i.e., Cortana<sup>2</sup>, Siri<sup>3</sup>, and Alexa<sup>4</sup>). Furthermore, recent conversational systems are widely adopted in a wide range of real-life applications for performing various tasks, such as booking a hotel and providing navigation information for cars. They are also capable of performing chitchat in an engaging and natural way.

In the early history of conversational artificial intelligence, ELIZA [1] and PARRY [2] were the most influential dialogue systems. Both of them were rule-based conversational systems. ELIZA was introduced in 1966 with an objective of simulating a Rogerian psychologist, which attempts to reach a conclusion or goal based on the patients' statement. It contains a fixed set of patterns by which it tries to reach a conversational goal. Five years later, a similar type of system called PARRY [2] was introduced. PARRY was the first conversational system that passed the *Turing test* [3]. It was introduced for performing a study on schizophrenia. Unlike ELIZA, it contains variables that affect the mental state of the system and may generate aggressive output based on variables' defined value.

With the advancement of machine learning and deep learning techniques, modern conversational systems have significantly improved. In contrast to the rule-based approaches, modern conversational systems are heavily driven by a large amount of data. Specifically, large-scale data is leveraged by deep learning models to develop the feature representation, which is capable of capturing diverse dialogue patterns. Natural language understanding and generation are the two major aspects one should consider for developing an effective conversational system. Recently, large language models have revolutionized the field of natural language processing. GPT-3 [4], DialoGPT [5], PaLM [6], OPT [7], and ChatGPT<sup>5</sup> are examples of recent large language models.

---

<sup>1</sup><http://www.gregoryaist.com/jods/index.html>

<sup>2</sup><https://www.microsoft.com/en-us/cortana>

<sup>3</sup><https://www.apple.com/de/siri/>

<sup>4</sup><https://developer.amazon.com/en-US/alexa>

<sup>5</sup><https://openai.com/blog/chatgpt/>

Modern conversational systems typically employ several natural language understanding (NLU) modules (i.e., intent detection and dialogue state tracking) in their pipeline. Pre-trained language models are typically employed for developing these modules. Recent conversational systems utilize pre-trained models primarily in two ways: 1) adopting an off-the-shelf pre-trained model and fine-tuning it for the downstream task [8, 9], and 2) pre-training the language model on large conversation corpora [5, 10]. Both strategies were shown to be successful for generating conversations in their respective use cases. A different line of research emphasizes developing end-to-end dialogue systems where a single model tackles all the sub-tasks (i.e., knowledge grounding and dialogue generation [11, 12, 13]) involved in the dialogue generation process.

Pre-trained language models are often provided with external data or knowledge to solve downstream tasks such as question answering, machine reading comprehension, and dialogue generation [14, 15, 11]. External knowledge (i.e., Wikidata [16], DBpedia [17]) empowers these conversational systems to generate informative and accurate dialogues. Depending on the data source, external knowledge can be structured and unstructured. For instance, knowledge graphs and databases are sources of structured knowledge [18, 17, 16], whereas text documents are unstructured data [19]. Incorporating structured and unstructured knowledge effectively into the conversational system is challenging. The primary challenges are as follows: 1) capturing the underlying semantics of the structured data (connections and relations between facts) is difficult since they are different than natural sentences [8], 2) knowledge-based dialogue generation requires understanding the question and structured knowledge for reasoning over structured information and dialogue generation at the same time [13, 20], 3) searching over a large-scale data source containing millions of facts to find relevant information for answering a question is challenging [14, 21], and 4) acquiring train data is one of the bottlenecks in the development of conversational question answering systems [22].

To grasp a better understanding of the research objectives of this thesis, it is necessary to understand how various knowledge-based conversational question-answering systems function. On a high-level, dialogue systems are divided into task-oriented and non-task oriented dialogue (also known as chitchat) systems. Task-oriented knowledge-based dialogue systems are often provided with a set of facts, usually in the form of a knowledge graph or relational database, to perform certain tasks. Typically, the knowledge is linearized into a sequence and fed to the learning system as input, along with the user utterance and dialogue history. Based on the feature representation of the input sequence, the system then generates the output (in generative dialogue systems [23, 24, 25]) or retrieves the correct response from a set of candidate responses (in retrieval-based dialogue systems [26, 27, 28]). Various techniques such as Memory network [29, 13, 20] and Copy network [30, 31] are often employed to embed the knowledge and prepare the feature representation for the learning paradigm. However, external knowledge integration into a training process is challenging. Considering external knowledge, developing fluent and factually correct dialogue systems remains one of the primary challenges researchers are attempting to address.

As large knowledge bases have gained increased popularity in recent years, question-answering over knowledge graphs (KGQA) has become a prominent type of conversational system. A KGQA pipeline includes three major components: 1) entity linker, 2) relation linker, and 3) answer extractor or SPARQL endpoint that extracts the answer from the KG. The entity and relation linker first identifies the entity and relation that appear in the question and then maps it to the corresponding facts in the knowledge graph. Finally, a SPARQL<sup>6</sup> query is typically constructed from the mapped facts and

---

<sup>6</sup>W3C Specification for SPARQL <https://www.w3.org/TR/sparql11-overview/>

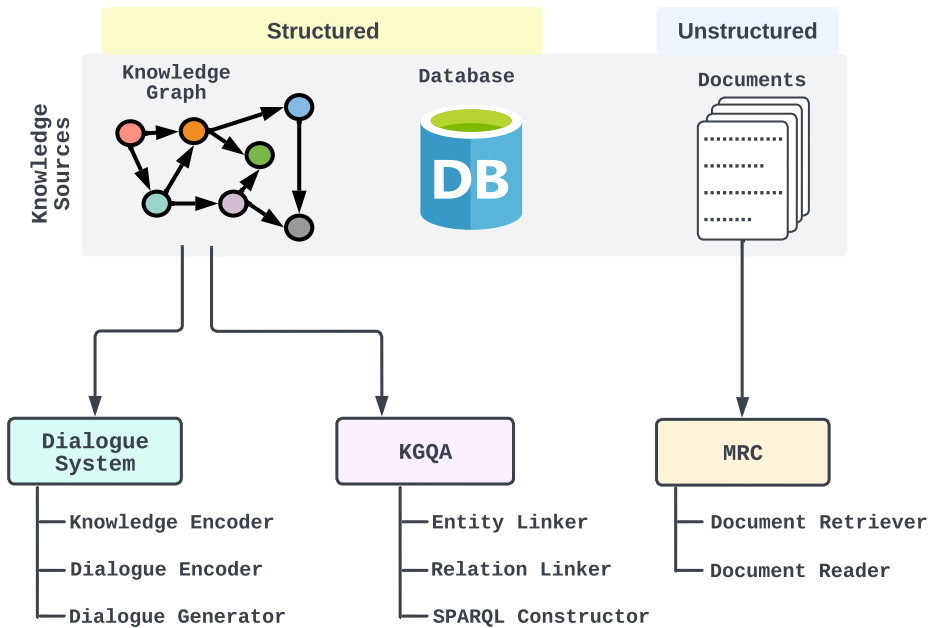


Figure 1.1: Conversational and question answering systems explored in this thesis.

executed to retrieve the answer from the target knowledge graph. SPARQL is a formal query language used to describe queries across various data sources, regardless of whether the data is stored natively or accessed via middleware as a Resource Description Framework (RDF). Knowledge graphs are the source of structured data, whereas text documents are unstructured. Machine reading comprehension (MRC) systems can perform question answering over unstructured text data. MRC systems have two essential components: 1) *Retriever* model and 2) *Reader* model. Given a question, the *Retriever* model retrieves a set of documents that are relevant for answering. Then, the answer to the question is extracted from the top-most retrieved document using a *Reader* model. Within the scope of this work, we explore KG-based task-oriented dialogue systems, question answering over knowledge graphs, and machine reading comprehension systems.

High-quality datasets and robust evaluation metrics are crucial for the development of a conversational system. Datasets are manually annotated in domain-specific conversational applications. Manual annotation is typically required to maintain the highest quality of the dataset. However, human annotation is time-consuming and resource intensive. On the other hand, an appropriate performance assessment is required to ensure the quality of a conversational system. Evaluation of conversational systems is difficult as it requires an understanding of the dialogue context and provided external knowledge. Word-overlap [32, 33] and contextualized embedding-based [34] matching are widely adopted techniques for evaluating conversational systems. Recent evaluation metrics employ pre-trained language models to obtain contextualized word embedding for dialogue evaluation [35, 36, 34]. Contextualized word embedding provides a rich representation of the word within a given context or sentence, thus effectively capturing a word’s semantic meaning. However, evaluating a system generated sentence against a reference sentence is difficult because of various surface forms and ordering of words within a sentence.

The remainder of this chapter describes the research objective of this thesis, as well as the challenges and research questions addressed in this thesis.

**Research Objectives:** Within the scope of this thesis, we investigate question answering, machine reading comprehension, and dialogue systems. The primary objective of this work is to incorporate external knowledge into pre-trained language models to advance the performance of conversational systems. Specifically, this thesis aims to investigate techniques to employ structured and unstructured external knowledge into various learning paradigm and leverages pre-trained language models for developing improved and factually correct conversational systems. Knowledge graphs (as structured knowledge) and text documents (as unstructured knowledge) are utilized as the source of external knowledge. Finally, to aid in the development of conversational systems, this thesis aims to develop a robust evaluation metric for measuring the performance of generative dialogue systems.

## 1.1 Motivation and Challenges

Definition: Primary Research Problem (RP)

Can incorporating structured and unstructured knowledge into pre-trained language models improve conversational question-answering systems?

### RP1 - Integration of Structured Knowledge into a Conversational System

Task-oriented dialogue generation is challenging since the underlying knowledge is often dynamic and effectively incorporating knowledge into the learning process is hard. This is due to the two different learning objectives that the existing systems follow. Typically, one part of existing systems focuses on capturing structured knowledge, whereas the other has a language modeling objective. It is particularly challenging to generate both human-like and informative responses in this setting.

### RP2 - Lack of Unsupervised KGQA Techniques

Most knowledge graph-based question answering systems rely on training data to reach their optimal performance. However, acquiring training data for supervised systems is both time-consuming and resource-intensive. Furthermore, training data is mostly tailored for specific knowledge graphs and often requires human annotation. The human annotation process involves identifying knowledge graph facts (i.e., entity and relation) appearing in the question and SPARQL query construction. Furthermore, KGQA is a multi-step process involving entity linking, relation linking, and answer extraction from a large knowledge graph. Overall, the complexities and inter-dependencies of components involved in the KGQA process make it difficult to develop an unsupervised KGQA system.

### RP3 - Extendability of a SPARQL Constructor:

SPARQL is a formal query language that is typically constructed and executed in KGQA to extract answer entities from a knowledge graph. SPARQL query generation from natural language questions is complex because it requires an understanding of the question and underlying knowledge graph (KG) patterns. Most SPARQL query generation approaches are template-based, tailored to a specific knowledge graph, hence, require pipelines with multiple steps, including entity and relation linking.

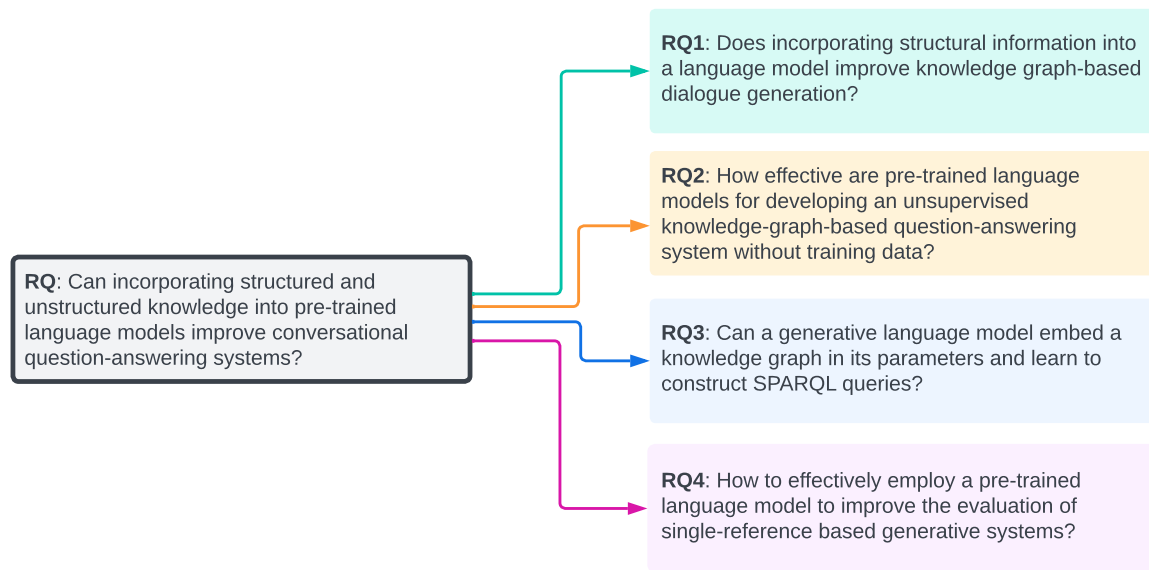


Figure 1.2: A breakdown of research questions.

Template-based approaches are also difficult to adapt and scale for new and large KGs. Generally, it requires manual efforts from domain experts to construct query templates for training template-based classifiers.

#### RP4 - Evaluation of Conversational Systems

Evaluating Natural Language Generation (NLG) systems is a challenging task. Firstly, the metric should ensure that the generated hypothesis reflects the reference’s semantics. Secondly, it should consider the grammatical quality of the generated sentence. Thirdly, it should be robust enough to handle various surface forms of the generated sentence. Thus, an effective evaluation metric has to be multifaceted.

## 1.2 Research Questions

Addressing the research problems discussed above, we formulate a set of research questions depicted in Figure 1.2. We further elaborate on the research questions below.

### Research Question RQ1

Does incorporating structural information into a language model improve knowledge graph-based dialogue generation?

Task-oriented dialogue systems are typically equipped with an external knowledge base (i.e., relational database or knowledge graph) as a source of structured information. The system must understand

the underlying connections between facts in structural knowledge to generate natural and informative dialogues. Therefore, it is essential to effectively incorporate structured knowledge into the learning mechanism. Recently, pre-trained language models have been increasingly used in conversational systems to generate conversations. Efficient and effective techniques are required to integrate structural knowledge into a language model for developing an improved conversational system.

### Research Question RQ2

How effective are pre-trained language models for developing an unsupervised knowledge-graph-based question-answering system without training data?

Knowledge graph is a source of an enormous amount of structured data that typically include millions of real-world facts. Because of the scale of KGs, question answering over the knowledge graph requires a large amount of training data. Entity and relation linking are the first two critical steps of KGQA. In the final step, a formal query (i.e., SPARQL) is typically executed to extract answers from the knowledge graph. Recently, pre-trained LMs have become popular as they are trained on large corpora and contain rich information in their parameters. Carefully designed mechanisms are required to employ and take advantage of the pre-trained language models in each step of the KGQA process.

### Research Question RQ3

Can a generative language model embed a knowledge graph in its parameters and learn to construct SPARQL queries?

Nowadays, Transformer-based [37] language models are widely adopted to capture text patterns and visual information. These models can embed a large amount of information in their parameters. Therefore, efficient and effective techniques are required to embed a knowledge graph into the language model. Furthermore, training such a model for generating structured sequences such as SPARQL queries requires careful designing of the system. This involves understanding the question and generating facts from the embedded knowledge of the language model.

### Research Question RQ4

How to effectively employ a pre-trained language model to improve the evaluation of single-reference based generative systems?

Evaluating natural sentences produced by generative systems such as dialogue and machine translation systems is challenging. Moreover, developing a uniform evaluation strategy for measuring the performance of such systems is difficult since it requires an understanding of the application, such as dialogue generation and data-to-text generation. In recent years, pre-trained language models have



been found effective in providing a rich and contextual embedding representation of textual data. However, effective techniques are required to adapt pre-trained embedding for the evaluation task. This involves capturing the generated sentences' semantic and syntactic meaning and grammatical acceptability.

## 1.3 Thesis Overview

This section provides a high-level overview of the thesis. First, addressing the research questions, we list down and briefly describe the contributions of this thesis. Then, we outline the accepted publications corresponding to the thesis's research contributions. Furthermore, the contributions of each author to the accepted papers are briefly discussed. Finally, we provide a structural overview of the thesis's remaining sections.

### 1.3.1 Contributions

#### Contributions for RQ1

A novel task-oriented dialogue system that incorporates external knowledge into a language model for task-oriented dialogue generation.

Addressing the Research Question RQ1, we propose a novel task-oriented dialogue system, dubbed *DialoKG* that employs structural information into a language model (LM) for generating informative dialogues. For this purpose, we exploit GPT-2 [38] - a language model developed based on a stack of Transformer decoders [37]. Specifically, we introduce a novel structure-aware multiple embedding layer-based knowledge embedding technique that constructively embeds the underlying relationship between the knowledge triples. *DialoKG* interprets the knowledge as a knowledge graph; therefore, separate embedding layers for word token, entity, triple and token type enable the system to capture the graph features (e.g., subject, relation and object). This enables the system to generate correct and human-like dialogues and prevents generating erroneous responses such as "*4 miles is located at 792 Bedoin Street Starbucks away*". Furthermore, the ability to correctly capture the relationship in the knowledge graph eliminates the need for template-based or sketch-based response generation.

#### Contributions for RQ2

An unsupervised knowledge graph-based question-answering system that does not require any training data and leverages only pre-trained language models for performing KGQA sub-tasks.

We present *Tree-KGQA*, a simple yet effective unsupervised KGQA method, leveraging pre-trained language models. The primary motivation of *Tree-KGQA* is to address the Research Question RQ2 and develop a dataset-independent KGQA system, which can answer natural questions from various

datasets without additional training or fine-tuning. *Tree-KGQA* adopts several powerful off-the-shelf language models, pre-trained on named entity recognition (NER) and natural language inference tasks for the KGQA sub-tasks [39, 40].

Specifically, we split the KGQA task into three sub-tasks: entity linking, relation linking, and answer entity extraction. **Firstly**, given a question, we employ a BERT-based [39] pre-trained NER model to detect the surface forms of the entities in the question. Additionally, we pre-process and index the contextualized representation of the entities into a dense space for effective and fast candidate entity generation during the inference. The index is utilized to generate a set of candidate entities, which are then disambiguated to obtain the final predicted entity. **Secondly**, by combining the 1-hop connected relations of the entities linked in the previous step, a set of candidate relations for relation linking is created. A pre-trained BART model [40] is then applied to the candidate relations to obtain the most probable relation in a zero-shot manner. **Finally**, we construct a set of  $k$ -level trees from the  $k$ -hop sub-graphs of the linked entities. Then, *tree-walking* and *tree-disambiguation* techniques are employed to extract answer entities from the constructed trees.

### Contributions for RQ3

A modular and expandable generative system for constructing SPARQL query from a natural question.

Addressing the Research Question RQ3, we propose a new system, dubbed *SGPT*, for SPARQL query generation. *SGPT* encodes the linguistic features of a natural language question (NLQ) and corresponding sub-graph information (i.e., entities, if provided), and leverages a generative language model (LM) to generate SPARQL queries. We hypothesize that a deeper understanding of the NLQ is crucial for generating a correct query, since a slight deviation in the syntactic structure of the question may result in a different SPARQL query.

Specifically, besides the standard word and positional embedding layers, we design special embedding layers that embed an arbitrary number of linguistic features of an NLQ, such as parts-of-speech (POS) tags and dependency tree features (i.e., dependency relations and information about tree node’s children). A stack of Transformer [37]-encoders is employed to encode the linguistic features. The proposed embedding techniques facilitate *SGPT* to inject additional knowledge (i.e., entities) as well as allow the integration of *SGPT* into pipeline-based systems in a modular fashion. Furthermore, we employ the Transformer [37]-decoder based language model GPT-2 [38], to generate SPARQL queries. Our training methodology enables *SGPT* to embed an arbitrary KG directly into the model parameters. Moreover, the system does not require any query template or KG as input at inference time.

### Contributions for RQ4

A metric for measuring the performance of generative systems that takes advantage of the pre-trained weights of language models.

We propose *RoMe*, an automatic and robust metric for evaluating NLG systems, addressing the Research Question RQ4. *RoMe* employs a neural classifier that considers the generated sentence’s grammatical,

syntactic, and semantic qualities as features to estimate the quality of the sentence against a reference sentence. **Firstly**, it calculates the earth mover’s distance (EMD) [41] to determine how much the hypothesis differs from the reference. During the computation of EMD, *RoMe* incorporates hard word alignment and soft-penalization constants to handle various surface forms of words in a sentence, such as repeated words and the passive form of a sentence. Using a semantically enhanced tree edit distance, the difference in syntactic structures between the reference and hypothesis sentences is quantified. **Thirdly**, the metric incorporates a binary classifier to evaluate the grammatical acceptability of the generated hypotheses. **Finally**, the scores obtained from the preceding steps are combined to form a representation vector, which is subsequently fed into a self-supervised network. The network produces a final score, referred to as *RoMe*’s output, representing the overall quality of the generated sentence.

### 1.3.2 Publications

The research papers accepted for publication in conferences and journals contributing to this thesis are listed below:

#### Conference Papers (peer reviewed):

- **Md Rashad Al Hasan Rony**, Ricardo Usbeck, and Jens Lehmann. 2022. *DialoKG: Knowledge-Structure Aware Task-Oriented Dialogue Generation*. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 2557–2571, Seattle, United States. Association for Computational Linguistics.
- **Md Rashad Al Hasan Rony**, Liubov Kovriguina, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann. 2022. *RoMe: A Robust Metric for Evaluating Natural Language Generation*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5645–5657, Dublin, Ireland. Association for Computational Linguistics.

#### System Demo Papers (peer reviewed):

- **Md Rashad Al Hasan Rony**, Ying Zuo, Liubov Kovriguina, Roman Teucher and Jens Lehmann, *Climate Bot: A Machine Reading Comprehension System for Climate Change Question Answering*. In Proceedings of IJCAI 2022, in AI for good track.

#### Journal Papers (peer reviewed):

- **Md Rashad Al Hasan Rony**, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann, *Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs*, in IEEE Access, vol. 10, pp. 50467-50478, 2022, doi: 10.1109/ACCESS.2022.3173355.
- **Md Rashad Al Hasan Rony**, Uttam Kumar, Roman Teucher, Liubov Kovriguina and Jens Lehmann, *SGPT: A Generative Approach for SPARQL Query Generation from Natural Language Questions*, in IEEE Access, vol. 10, pp. 70712-70723, 2022, doi: 10.1109/ACCESS.2022.3188714.

### 1.3.3 Author Contributions

In *DialoKG*, Md Rashad Al Hasan Rony developed the core concepts, implemented, conducted experiments, and wrote the paper. Prof. Dr. Jens Lehmann and Prof. Dr. Ricardo Usbeck reviewed the work and provided thoughtful feedback and comments on the writing. In *RoMe*, Md Rashad Al Hasan Rony contributed in developing the core concepts, implemented, conducted experiments, and wrote the paper. Liubov Kovriguina and Debanjan Chaudhuri partially contributed to the concept, specifically in the implementation and writing about the grammatical acceptability feature of the metric. In the system demo paper, *Climate Bot*, Md Rashad Al Hasan Rony and Liubov Kovriguina developed the core concepts for constructing the proposed dataset and have written the paper. Additionally, Md Rashad Al Hasan Rony implemented the system code, which includes a data parser, system pipeline, and user interface. Ying Zuo primarily contributed in the training and evaluation of the system. Roman Teucher was involved in the data annotation and evaluation. Prof. Dr. Jens Lehman provided valuable reviews and comments on the writing.

In *Tree-KGQA*, Md Rashad Al Hasan Rony developed the core concepts, implemented them, conducted all experiments and evaluations, and wrote the paper. Debanjan Chaudhuri took part in the conceptual discussions and partially contributed to the implementation. Prof. Dr. Ricardo Usbeck and Prof. Dr. Jens Lehmann reviewed the work and provided feedback and comments on the writing. In *SGPT*, Md Rashad Al Hasan Rony developed the complete concept, implemented and evaluated the proposed system, and wrote the paper. Uttam Kumar and Roman Teucher each evaluated a baseline system. Liubov Kovriguina reviewed the work and partially wrote the paper. Prof. Dr. Jens Lehmann reviewed the work and provided insightful comments on the writing.

### 1.3.4 Thesis Structure

This thesis is organized into ten chapters. The primary motivations and research problems addressed within the scope of this thesis are described in Chapter 1. Following that, we further discuss the challenges and research questions required to be addressed to tackle the primary research problem. Chapter 2 provides detailed background knowledge for understanding the core concepts used throughout this thesis. Language models, conversational systems such as question answering over knowledge graphs, dialogue systems, machine reading comprehension, and evaluation of generative systems are the core concepts of this thesis. Chapter 3 summarizes the state-of-the-art research that falls within the scope of this thesis. The following five chapters discuss the core contributions of this thesis that address the primary research problem. Chapter 4 discusses the techniques for knowledge integration into the language model for task-oriented dialogue generation. The chapter also demonstrates how structure-aware knowledge integration improves task understanding, resulting in high-quality and human-like dialogues. Chapter 5 summarizes techniques for performing question answering over knowledge graphs without training data. Chapter 6 introduces a generative approach for SPARQL query generation. It also discusses techniques to embed knowledge graph facts into language models' parameters. Chapter 7 discusses a machine reading comprehension system that can perform question answering over unstructured data from the climate change domain. In Chapter 8, we propose an evaluation metric to assess the performance of generative systems such as data-to-text, dialogue, and natural language generation. Chapter 9 includes the concluding remarks and future directions of this thesis. Finally, Chapter 10 outlines a list of accepted papers contributing to this thesis. All the Figures without any source mentioned in this thesis report are drawn by Md Rashad Al Hasan Rony.

# Background Knowledge

---

Conversational artificial intelligence is one of the challenging and widely studied fields of natural language processing. This chapter provides a theoretical understanding of the concepts used within the scope of this thesis. Specifically, this chapter provides a comprehensive overview of language models, knowledge graphs, conversational question answering systems, and evaluation metrics.

## 2.1 Language Models

Text representation models, also known as language models, can learn the representations for sub-words, words, sentences, or documents, in general, for any unit of text. Therefore they are widely adopted for tackling downstream NLP tasks such as sentiment analysis, question answering, natural language generation, machine translation, and text summarization. Earlier models mainly focused on words as input; however, several models emphasize learning characters as input (i.e., CharCNN [42], FastText [43], ELMo [44]). Most of the recent large language models (i.e., BERT [45], GPT-2 [38] and XLNet [46]) are developed based on concept of multi-head self-attention mechanism proposed in the Transformer model [37]. In this work, we employ and exploit several Transformer-based language models (i.e., GPT-2 [38], BERT [39], BART [40], SBERT [47], and ALBERT [48]) to tackle downstream tasks, such as task-oriented dialogue (Chapter 4), unsupervised question answering (Chapter 5), machine reading comprehension (Chapter 7), and evaluation metric (Chapter 8). Below we provide a brief description of various language models.

### 2.1.1 Static Word Embedding Models

Static word embedding models are simple neural networks typically containing one or a few hidden layers. The objective of most static word embedding models focuses on learning the vector representation of the text at the word level or n-gram word level. Typically, these models are trained to predict a vector representation of the word based on the input text. Word2Vec [49], GloVe [50], Doc2Vec [51], and FastText [43] are the most widely used static word embedding models. Below we provide a brief description of the models.

**Word2Vec.** A research team at Google proposed one of the prominent techniques to represent the word as a vector, called Word2Vec [49]. Word2Vec offers two separate unsupervised autoencoding

models to learn word representation: *Continuous Bag of Words (CBOW)* and *Skip-gram*.

**GloVe.** A matrix factorization-based learning technique, *GloVe*, was proposed by Pennington *et al.* [50]. In contrast to CBOW and skip-gram, a global co-occurrence of words is computed in GloVe. However, in GloVe, the word representations are learned in an unsupervised way.

**Doc2Vec.** The Doc2Vec [51] algorithm is an extension of Word2Vec, which can learn a fixed-length representation of a sentence, paragraph, or document. The proposed algorithm overcomes the two drawbacks that the CBOW algorithm poses: the order of words is not captured, and the semantics of the words are ignored.

**FastText.** The FastText algorithm [52] exploits the skip-gram architecture and takes the n-grams of a word as the input, allowing the algorithm to add sub-words into the vocabulary. For instance, in FastText the n-gram ( $n=3$ ) of the word "hello" is as follows:  $\langle \text{he}, \text{hel}, \text{ell}, \text{llo}, \text{lo} \rangle$ ,  $\langle \text{hello} \rangle$ . It is noteworthy that the original word is also considered along with the n-grams, where "<" and ">" are the special tokens. Unlike the previous algorithms, where a vector representation is learned for each word, FastText learns to generate a vector representation for each sub-word or n-gram of a word. Learning the sub-word information enables the algorithm to predict a vector representation for a previously unseen word.

Although static word embedding models are useful in handling simple tasks such as word similarity. They often fail in complex tasks where the overall contextual understanding of the task is required (i.e., question answering and dialogue generation).

## 2.1.2 Recurrent Models

**Recurrent Neural Network (RNN).** RNN can be considered as a folded neural network, which is constructed by appending a copy of the same neural network together. This property enables RNNs to learn sequences better. RNNs can take an arbitrary number of inputs, where the computation takes previous inputs into consideration, and the weights are shared across all the time steps. However, they are computationally slow and do not consider future input for the computation of the current state. Furthermore, RNNs often suffer from vanishing and exploding gradient problems. During the training, the model parameters get updated using the gradient values. If there are many layers in the network, the gradient becomes very small, which makes the network unstable during the training. Because of the chain rule, the gradient values are propagated through each layer down to the initial layer. If the gradient is very small because of the chain multiplications, it becomes zero before reaching the initial layer. This kind of gradient prevents the system from converging to the optimum, making the system unstable. Similarly, if the gradients are too large, they result in large weights. This causes the gradient to explode and leads the system toward divergence.

**Long Short-Term Memory (LSTM).** Addressing the shortcomings of RNNs, a long short-term memory network (LSTM) was designed by Hochreiter and Schmidhuber [53]. Similar to the RNN architecture, LSTM also contains folded neural networks. However, through gating mechanisms, LSTM controls the flow of the data. LSTM includes three gates: 1) *Input gate* (decides which

information to store in the cell state from the input), 2) *Forget gate* (decides which information to forget), and 3) *Output gate* (decides what to output).

**Gated Recurrent Units (GRU).** Gated recurrent unit (GRU) is another variation of RNN that is similar to LSTM and deals with the vanishing gradient problem. GRU combines the *input* and *forget gates* used in the LSTM and creates an *update gate*. Furthermore, a *reset gate* resets the information that is not required for the current state of the unit. GRU does not maintain any candidate cell state, unlike LSTM, and computes a candidate hidden state to generate the final hidden state. GRU consumes less memory and is easy to modify as it contains less number of gates than LSTM and does not require any memory units.

Besides static word embedding and recurrent models, convolution-based [54] models are also used to solve downstream tasks such as text classification [55, 56]. However, in recent years attention-based models have revolutionized the language modeling task and the field of natural language processing. Below we discuss about attention-based models in detail.

### 2.1.3 Attention-based Models

The attention model was first developed for solving machine translation tasks [57]. Ever since it has grown in popularity for a plethora of artificial intelligence (AI) applications. Nowadays, Attention-based models are widely used in natural language processing [39, 58, 7], computer vision [59], multi-modal tasks [60], recommendation [61, 62], and graph-based systems [63, 64]. Human biological processes are better suited to explaining the intuition behind the attention mechanism. For instance, our visual processing system has a tendency to selectively focus on some elements of an image while disregarding other extraneous information in order to aid perception [65]. In image captioning, some areas of the input image may be more useful for creating the following word in the caption. Therefore, attention-based models are intuitively a better option for solving these kinds of tasks. Figure 2.1 depicts diverse applications of attention-based pre-trained models.

In the past, encoder-decoder-based model was a popular choice for training a model for sequence-to-sequence tasks. However, the traditional encoder-decoder-based system suffers from two flaws:

1. The encoder encodes all input information into a single vector, resulting in information loss, and
2. The decoder has no connection to the input sequence, resulting in a lack of context for the generation process.

Addressing the issue, an attention-based encoder-decoder architecture was proposed [57], facilitating the decoder to input sequence for better text generation. The key idea of this approach is projecting attention weights over the input sequence to focus on the most relevant part of the input sequence for decoding the next word of the output sequence. The attention weights are obtained by training an additional feed-forward network. The attention-based sequence-to-sequence model can be formally

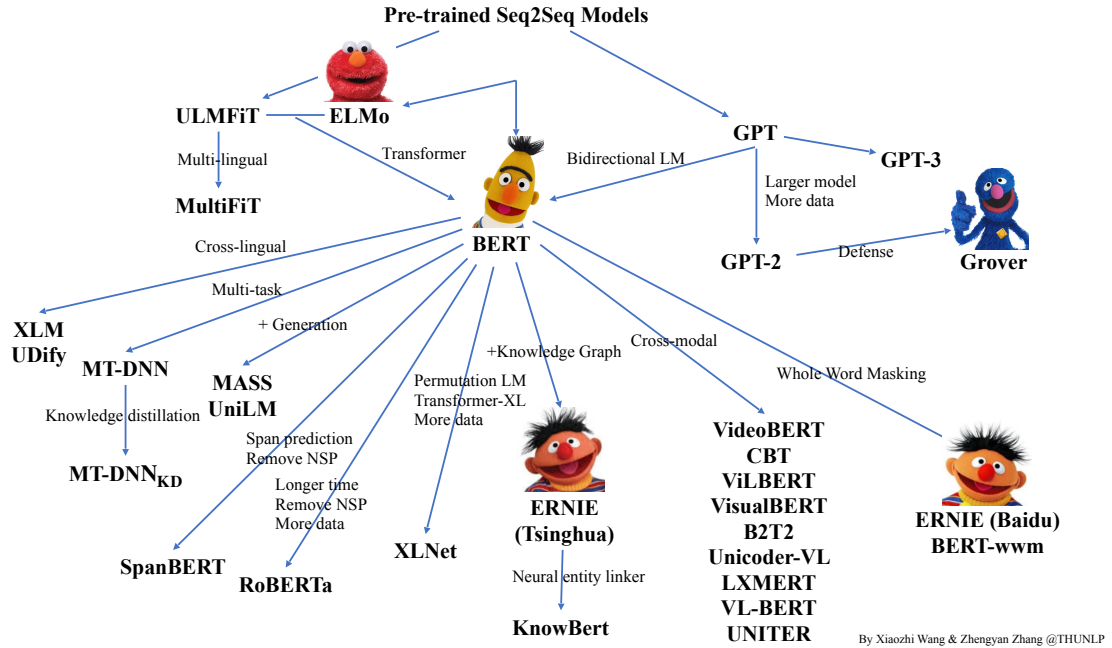


Figure 2.1: An illustration of pre-trained language model family. Model names are in **bold** text. Figure adapted from THUNLP.

defined as:

$$\begin{aligned}
 y_j &= f_n(y_{j-1}, dec_j, c_j), \\
 dec_j &= f(dec_{j-1}, y_{j-1}, c_j), \\
 c_j &= \sum_{i=1}^N \alpha_{ij} h_i, \\
 \alpha_j &= f_d(f_a(dec_{j-1}, h_i)), \\
 h_i &= f(x_i, h_{i-1}),
 \end{aligned} \tag{2.1}$$

where,  $x$ : the input sequence,  $y$ : output sequence,  $h$ : encoder hidden states,  $c$ : context vector,  $\alpha$ : attention weights over the input sequence,  $dec$ : decoder hidden states,  $f, f_n$ : non-linear functions,  $f_a$ : alignment function, and  $f_d$ : distribution function.

A self-attention-based *Transformer* architecture was first introduced by Vaswani *et al.* [37]. The proposed model eradicates the need for sequential processing and proposes a multi-headed self-attention mechanism to capture the global dependencies between input and output sequences. In contrast to the recurrent neural network (RNN), where the order of the words is learned sequentially, *Transformer* introduces a positional encoding technique that handles the word order efficiently, leveraging sinusoidal functions. The author demonstrated that sinusoidal functions could capture sequences larger than the ones encountered during the learning. Within the scope of this thesis, we leveraged various Transformer-based pre-trained language models. We fine-tune GPT-2 [38] in Chapter 4 for task-oriented dialogue generation, leverage pre-trained BERT [45] and BART [40] models



for entity and relation linking respectively in Chapter 5, fine-tune GPT-2 [38] for SPARQL query generation in Chapter 6, leverage SBERT [47] for document indexing and fine-tune ALBERT [48] for answer span extraction in Chapter 7, and utilize BERT [45] and ALBERT [48] for developing evaluation metric in Chapter 8. Figure 2.2 illustrates a high-level architecture of the Transformer network. The architecture follows an encoder-decoder design pattern.

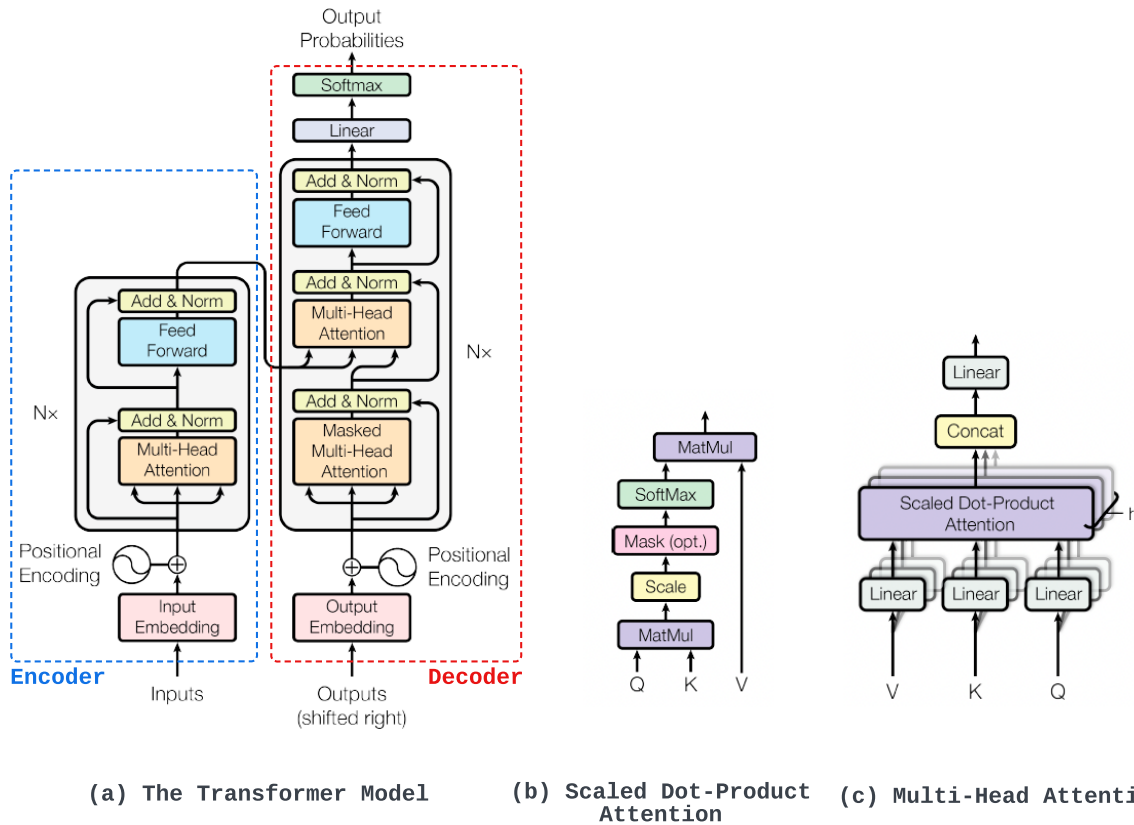


Figure 2.2: Transformer model architecture (Figure taken and adapted from Vaswani *et al.* [37]).

**Transformer Encoder:** The encoder block includes a stack of  $N$  number of identical layers. Each layer in the stack consists of two sub-layers: a multi-head self-attention layer, followed by a position-wise fully connected feed-forward layer. Furthermore, each layer includes a residual connection followed by layer normalization to eliminate the vanishing gradient problem.

**Transformer Decoder:** The decoder works autoregressively, meaning that the decoder predicts the next token based on the previous tokens in the sequence. Specifically, it predicts the next token based on the encoder output and self-attending to the previous output tokens. Similar to the encoder, the decoder also includes  $N$  identical layers. However, each layer employs an additional sub-layer to perform multi-head attention over the output of the encoder. Furthermore, the multi-head self-attention sub-layer in the decoder is modified into *masked* multi-head self-attention to prevent the decoder from attending to future tokens. Notably, the subsequent tokens of the output positions are masked, which

enables the learning of next-word prediction. Moreover, the output embedding is shifted one position to the right to facilitate the next token prediction.

It is worth noting that, each layer of encoder and decoder contains a position-wise feed forward layer, which is applied to the each position separately. We briefly discuss the core concepts of the Transformer model below:

- *Input and Output Embedding*: Trained embeddings are used to transform an input token into a continuous vector of dimension  $d_{model}$ . An additional positional embedding is added to the input and output embedding before feeding it to the encoder or decoder, which captures the word orders in a sequence. Furthermore, the trained embeddings are also used to compute next-token probabilities.
- *Positional Encoding*: Since Transformer does not include recurrence or convolution, it employs a positional embedding to handle word orders in a sequence. The authors employ sinusoidal functions of different frequencies as follows:

$$\begin{aligned} PE_{(idx,2i)} &= \sin(idx/10000^{2i/d_{model}}) \\ PE_{(idx,2i+1)} &= \cos(idx/10000^{2i/d_{model}}) \end{aligned} \quad (2.2)$$

where  $PE$ ,  $idx$ , and  $i$  are positional embedding, position, and dimension, respectively. The sinusoidal functions allow the model to tackle sequences larger than the ones encountered during the learning.

- *Scaled Dot-Product Attention*: The attention mechanism learns the mapping of a query and a set of key-value pairs to an output (Figure 2.2). A weighted sum of values is considered the output in this process. The queries and their corresponding keys are used to compute the weighted values. A multiplicative attention (Dot-product) in the Transformer model is computed as follows:

$$Attn(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (2.3)$$

where  $Attn(\cdot)$  is an attention function. Because of the multiplication operations, the value may grow large in magnitude, which may result in a gradient close to zero (vanishing gradient). A scaling factor of  $\frac{1}{\sqrt{d_k}}$  is employed to get rid of the vanishing gradient issue.

- *Multi-Head Attention*: A multi-head attention mechanism is employed in Transformer to attend tokens from different layers at different positions, which allows learning global dependencies (Figure 2.2). Let  $h$  be the number of attention layers or heads and  $d_k = d_v = d_{model}/h$ . Formally, the multi-head attention is computed as follows:

$$\begin{aligned} MultiHeadAttn(Q, K, V) &= [\text{head}_1; \dots; \text{head}_h]W^O \text{ where,} \\ \text{head}_i &= Attn(QW_i^Q, KW_i^K, VW_i^V). \end{aligned} \quad (2.4)$$

In Equation 2.4,  $W_i^Q$ ,  $W_i^K$ , and  $W_i^V$  are weight trainable weights with a dimension of  $\mathbb{R}^{d_{model} \times d_k}$ , where  $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ .

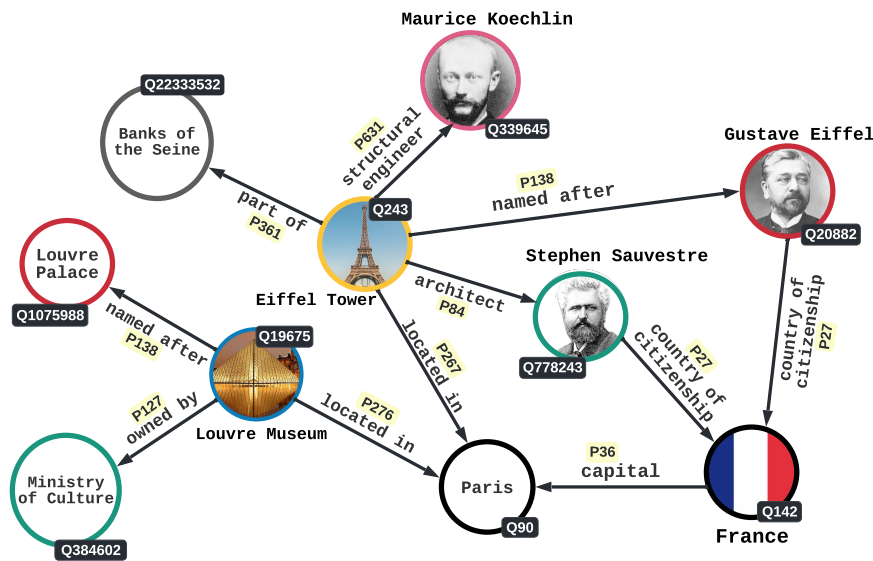


Figure 2.3: An illustration of a sub-graph of the Wikidata knowledge graph. The entity and relation IDs are shown in the text with black and yellow background, respectively.

Transformer-based pre-trained language models (PLMs) are extensively used in natural language processing applications such as machine translation [66, 67, 68], summarization [69], question answering [70, 71], sentiment analysis [72], and text classification [73, 74]. Transformer-based PLMs improved the performance of these downstream tasks over RNN and convolution-based methods. Within the scope of this thesis, we heavily utilized pre-trained language models for advancing conversational question answering systems.

## 2.2 Knowledge Graph

Knowledge Graphs are considered one of the largest sources of structured data [75]. A knowledge graph can be viewed as an abstraction of the real world as it stores real-world entities and their interrelations. Furthermore, knowledge graphs provide an easy way to store and access large-scale structured data. Because of their scale and flexibility, knowledge graphs are widely used in web searches, question answering systems, dialogue systems, and link prediction tasks. Nowadays, big technology companies such as Google, Microsoft, and Facebook use their own knowledge graph as a part of their infrastructures [76]. Freebase [18], DBpedia [17], Wikidata [16], and YAGO [77] are some of the widely used knowledge graphs in academia and industries. Figure 2.3 depicts a sub-graph of the Wikidata knowledge graph.

Although the term "Knowledge Graph" was introduced by Google in 2012 [78], the notion of a "Knowledge Graph" remains contentious [79, 80]. Färber et al [81] defined knowledge graph as an RDF<sup>1</sup> graph. An RDF graph includes a set of ordered RDF triples  $(s, p, o)$ , where  $s \in U \cup B$  is a subject,  $p \in U$  is a predicate or relation, and  $U \cup B \cup L$  is an object. These RDF terms are either an URI  $u \in U$ , a blank node  $b \in B$ , or a literal  $l \in L$  [81]. Paulheim [82] defined knowledge

<sup>1</sup><https://www.w3.org/RDF/>

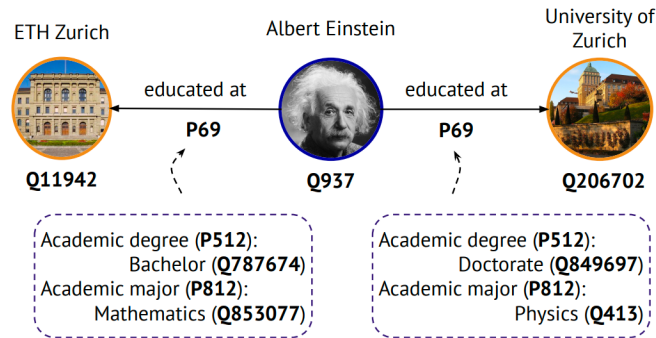


Figure 2.4: An example of hyper-relational model (Figure from Galkin *et al.* [83]).

graph as follows: "A knowledge graph (i) mainly describes real world entities and their interrelations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.". Although knowledge graphs contain a large number of structured facts, because of their scale, it is typically challenging to utilize them effectively in real-life applications. Within the scope of this thesis, we explored Wikidata [16] for developing an unsupervised KGQA system in Chapter 5 and modeled relational data as a knowledge graph for dialogue generation in Chapter 4.

Wikidata is a hyper-relational knowledge graph constructed collaboratively and operated by Wikimedia foundation<sup>2</sup> [82]. Lets define a fine set of entities  $\mathcal{E}$ , a finite set of relations  $\mathcal{R}$ , and the power set  $\mathcal{P} = 2^{(\mathcal{R} \times \mathcal{E})}$ . Then, a hyper-relational knowledge graph can be formally defined as  $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$ , where  $\mathcal{S} \subset (\mathcal{E} \times \mathcal{R} \times \mathcal{E} \times \mathcal{P})$  is a set of qualified statements [83, 84]. Figure 2.4 illustrates an example of a hyper-relational model. Wikidata is also available in multiple languages, which the Wikimedia foundation maintains. The January 2023 version of Wikipedia contains 101,486,080 data items.

## 2.3 Conversational Systems

### 2.3.1 Dialogue Systems

Dialogue systems can be primarily divided into two categories: Task-oriented and non-task-oriented dialogue systems. A brief description of these dialogue systems is provided below.

#### Task-oriented Dialogue Systems

Task-oriented dialogue (ToD) systems are designed to assist users in completing their tasks or achieving a certain objective. The functionalities of this kind of system include understanding the user utterance, tracking the current state of the conversation, and generating action or natural response. ToD has drawn a lot of interest from both academic and industry fields because of its wide range of use cases. Hotel and restaurant reservations, flight booking, and car navigation are the most popular applications of task-oriented dialogue systems. Below we summarize the core concepts involved in a task-oriented dialogue system:

<sup>2</sup><http://wikimediafoundation.org/>

- *Natural Language Understanding (NLU)*: For a given user utterance, NLU aims to capture a semantic abstraction of the user utterance. The semantic abstraction includes the intent and slot-value pairs. Consider the user utterance in the hotel reservation task, "I am looking for a cheap place to stay including free parking", the intent is "Query" and slot-value pairs are <Price-range, Cheap> and <Parking-cost, Free>. Hence, NLU can be decomposed into two primary sub-tasks: 1) intent detection as a classification task and 2) slot-value pair recognition as a sequence labeling task. A dialogue manager, which contains a dialogue state tracker and action generator, takes the NLU result as input and processes the final action for the natural language generation task.
- *Dialogue State Tracking (DST)*: Dialogue state tracker takes the dialogue context (i.e., dialogue history, current user utterance) as input and learns to estimate the user goal at each time step  $t$ . Early research focused on a finite set of dialogue states and modeled the interaction as a Markov Decision Process [85, 86]. Having a finite set of states make the system less robust to unseen situations. Addressing this issue, recent dialogue state trackers utilize the slot-value pairs to determine the user goal, given the current dialogue utterance [87, 88]. In this scenario, DST is modeled as a multi-task classification task.

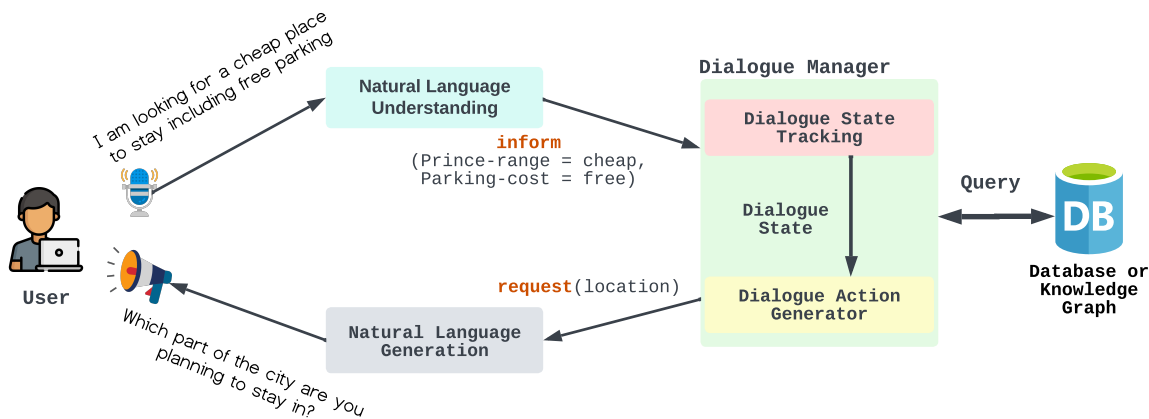


Figure 2.5: A high-level illustration of task-oriented dialogue system pipeline.

- *Dialogue Policy or Action generation*: Traditional dialogue systems are often equipped with dialogue policy learners that aim to capture the action based on the current dialogue state. In a typical dialogue policy learning setting, a supervised model learns to predict a dialogue action for each user utterance based on a fixed corpus. Then using a reinforcement learning model, the system is fine-tuned on the real users [89, 90]. Recent ToD systems learn the dialogue policy in their large-scale models' parameters [8] to get rid of complex training mechanisms.
- *Natural Language Generation (NLG)*: A language model is typically trained to generate a response in the form of natural language, taking the dialogue policy into consideration. However, recent end-to-end generative systems take the dialogue history  $\mathcal{H}$ , user utterance  $\mathcal{U}$  and language

models parameters  $\theta$  as input, learn to generate a response as follows:

$$p(S_t|\mathcal{H}, \mathcal{U}, \theta) = \prod_{i=1}^n p(s_i|s_1, \dots, s_{i-1}, \mathcal{H}, \mathcal{U}, \theta), \quad (2.5)$$

where  $S_t$  is the response at time step  $t$  and  $n$  is the response length. Figure 2.5 depicts a high-level overview of a task-oriented dialogue system pipeline.

Task-oriented dialogue systems are generally empowered by external knowledge to assist in completing the task. Typically the external knowledge comes in the form of database or knowledge graph triples. Integrating external knowledge into the dialogue generation mechanism is a complex process that requires an understanding of the knowledge and user utterance besides the language modeling objective. Taking the additional knowledge  $\mathcal{K}$  into account, the response generation objective of an end-to-end system can be redefined as follows:

$$p(S_t|\mathcal{H}, \mathcal{U}, \mathcal{K}, \theta) = \prod_{i=1}^n p(s_i|s_1, \dots, s_{i-1}, \mathcal{H}, \mathcal{U}, \mathcal{K}, \theta) \quad (2.6)$$

Various modular approaches such as Memory network [29] and Copy mechanism [30] are often employed in the learning paradigm to implicitly filter relevant knowledge for dialogue generation. Using a memory network, the system learns to predict a set of pointers that select relevant knowledge from the memory to respond to the current user utterance [91, 20]. In a different approach, task-oriented dialogue systems integrate copy mechanisms to copy facts from the provided knowledge for dialogue generation [11, 31].

## Non-task-Oriented Dialogue Systems

Open-domain dialogue systems are also known as non-task-oriented or chit-chat dialogue systems. Non-task-oriented dialogue (Non-ToD) systems are typically data-centric and not restricted to any task or domain. Existing non-ToD systems can be divided into three main categories: 1) *Retrieval systems*, 2) *Generative systems*, and 3) *Ensemble systems*. This type of system is capable of generating diverse and creative responses. Retrieval systems learn to retrieve the correct dialogue response from a set of pre-defined responses [26, 27, 28]. The output of these kinds of systems is limited by a finite set of responses. Generative systems, on the other hand, take the user utterance and dialogue history as input and learn to generate a response in a natural language [23, 24, 25]. In contrast to the former approaches, ensemble systems propose a hybrid method where both generative and retrieval systems are combined to form a flexible and extendable system [92, 93]. Very recently, OpenAI introduced ChatGPT<sup>3</sup>, which claims that the system is capable of answering follow-up questions, admitting its mistakes, and handling a diverse range of questions.

### 2.3.2 Question Answering Over Knowledge Graphs

Typically, question answering over knowledge graphs requires three key steps: 1) Entity linking, 2) Relation linking, and 3) Answer extraction. These steps are briefly discussed below.

<sup>3</sup><https://openai.com/blog/chatgpt/>

**Entity Linking.** Knowledge graphs such as Wikidata [16] and DBpedia [17] are a source of large-scale structured data. In KG-based applications such as question answering, relation extraction, and semantic search, entity linking is a widely used technique, primarily utilized to identify and connect the surface form of a text chunk to a knowledge graph entity. The linking facilitates the applications with additional facts (entities and relations) connected to the linked entity that might work as an additional signal to the main application. Lets consider the Wikidata KG and a question  $Q$ , Which company's CEO is Tim Cook?. The surface form of the entity is Tim Cook that connects to the KG entity <https://www.wikidata.org/wiki/Q265852>. There are three steps in entity linking:

1. *Entity Mention Detection:* Given a question  $Q$ , the first task of an entity linker is to identify the surface form of entity mentions  $m=\{m_1, \dots, m_n\}$ , in the question. Here,  $n$  is the number of entity appear in  $Q$ . A question may include multiple entities, depending on the complexity of the question. In the running example, Tim Cook is the surface form of the entity.
2. *Candidate Generation:* A knowledge graph includes millions of facts. It is a very common scenario that a knowledge graph contains entities with identical surface forms, representing two different identities in the real world. Therefore, it is crucial to find candidate entities that share the same surface form  $m_i$  to narrow the search space. For the running example, the candidate entity list includes *Tim Cook (Q7803347)* an Australian rules footballer, *Tim Cook (Q1404825)* an American ice hockey player, *Tim Cook (Q265852)* an American business executive.
3. *Entity Disambiguation:* Entity disambiguation is required only in the cases where multiple entities with the same surface form exist. The context of the question, additional entity description, and relation information are typically exploited to perform entity disambiguation. In the running example, the relation CEO could be utilized to perform entity disambiguation because a footballer or ice hockey player does not have the relation CEO directly connected to them. Thus, the correct entity is predicted and linked to the surface form of the text.

**Relation Linking.** Large KGs, such as Wikidata [16], contains over 100 million entities and 10 thousand relations. Despite the number of relations being less compared to entities, relation linking is the most challenging tasks in KGQA. Since knowledge graph relations appear in the question in a variety of surface forms, detecting their mention is a non-trivial task. The relation mention detection is not as straightforward as entity-mention detection. For instance, in the question What kind of disease does Montel Williams have?, the linked relation is <https://www.wikidata.org/wiki/Property:P1050> which is *Medical Condition (P1050)*. However, from the surface form of the question, it is not possible to explicitly extract the relation mention. Relation linking requires more context from the question, including entity information. Because of the complexity of relation linking as an individual task, recent approaches attempt to tackle the problem by jointly training with an entity linker [94, 14].

**Answer Extraction.** Existing answer extraction methods are primarily divided into 1) an information retrieval (IR) based approach and 2) a direct graph search-based approach. In the first approach, inverted indexes are built over the entire knowledge graph depending on the search mechanism (horizontal or vertical indexing). Then given a question, a candidate list is generated and later

re-ranked to get the top-k results using a graph structure. Typically, an in-memory representation of the graph is employed for faster traversal [95, 96]. The latter approach aims at executing SPARQL query over a knowledge graph to extract answer entities. This is done in two steps, constructing a SPARQL query from a natural question and then executing it over a target KG to obtain the answer entity or entities [97, 98]. Below we provide a brief description of the SPARQL query.

*SPARQL Query*: The term "SPARQL" stands for *SPARQL Protocol and RDF Query Language*<sup>4</sup>. According to the official definition, a SPARQL query can be formally considered as a tuple  $\langle GP, DS, SM, R \rangle$ , where  $GP$  is a graph pattern (query pattern),  $DS$  is an RDF dataset,  $SM$  is a set of solution modifiers (ORDER, PROJECTION, DISTINCT, OFFSET, LIMIT),  $R$  is a result form (SELECT, CONSTRUCT, DESCRIBE and ASK)<sup>5</sup>. Figure 2.6 illustrates the terms used in the formalization. Similar to the previous works, SGPT aims at generating the query body, which includes the result form, graph pattern, and solution modifiers.



Figure 2.6: An illustration of SPARQL query components.

In this thesis, we investigated two answer extraction methods. In Chapter 5 we introduce a tree disambiguation-based new answer extraction method. In Chapter 6, we propose a generative approach to generate SPARQL query from natural questions.

### 2.3.3 Machine Reading Comprehension

The most standard method for determining whether or not a person completely understands a piece is to have them answer questions about it. Machine reading comprehension, like the human language examination, is a natural way of measuring a machine's or system's ability to comprehend a language. A typical MRC assignment necessitates a computing system reading a set of text paragraphs and then answering questions about the text, which is a complex task. Recent research on machine reading comprehension can be divided into four primary categories [99]: 1) *Span prediction*, 3) *Cloze style*, 3)

<sup>4</sup><https://www.w3.org/TR/sparql11-query/>

<sup>5</sup><https://www.w3.org/2001/sw/DataAccess/rq23/defns>



*Multiple-choice*, and 4) *Free-form answer*. Below a brief description of each type of MRC system is provided:

- *Span prediction*: The system learns to predict the answer span  $(i_{start}, j_{end})$ , where  $0 \leq i_{start} \leq j_{end} \leq |p|$ , given a question  $Q$  and a context paragraph  $p$ . Here,  $|p|$  is the length of the paragraph  $p$ . This technique is also called *Extractive Question Answering*.
- *Cloze style*: In this category, the task is to predict a word from the context or vocabulary to complete or fill in the placeholder or blank position of a question. For instance, what is the "\_\_\_\_" of Germany?
- *Multiple-choice answer*: In this setting, the system predicts the correct answer  $a_i$  from a set of hypothesis answers,  $\mathcal{A} = \{a_1, \dots, a_k\}$ , given a question  $Q$ .
- *Free-form answer*: Unlike the previous categories, the predicted free-form answer is not restricted to the context or the span within the context. The generated answer can be a word or sequence of words from the vocabulary  $\mathcal{V}$ .

Within the scope of this thesis, we explored the first type of MRC that predicts the answer span within the text paragraph. Figure 2.7 depicts such an MRC system. This kind of MRC system consists of two main components: 1) *Document Retriever* and 2) *Document Reader*. Within the scope of this thesis, in Chapter 7, we develop an MRC system that predicts answer span.

## Document Retriever

Given a question  $Q$ , the task of the document retriever is to fetch  $n$  number of relevant text paragraphs from the document store that can potentially answer the question. The relevance of the documents is typically measured by checking how contextually similar the retrieved documents are to the input question. Here, the document store can be a data structure, database, or file that contains the complete set of texts on which the system should operate. A contextual document ranker retrieves the paragraphs from the document store and ranks them based on their relevance score. A neural document retriever is typically used as the contextual document ranker for this task. The top-ranked paragraph or document is then sent to a *Reader* module of the MRC system. We use the terms "document" and "paragraph" interchangeably to indicate retrieved text throughout the MRC-related discussions. Document retrieval research can be divided into the following three categories:

- *Sparse Retriever*: Sparse retrievers [100, 101] employ sparse representation of the text to reduce the search space and retrieve related documents based on the search query. Specifically, sparse retrievers utilize an exact term-matching heuristic to narrow down the initial set of retrieved documents, then re-rank them to get the final set of retrieved documents. Classical IR approaches such as TF-IDF [100] and BM25 [101] are sparse retrievers.
- *Dense Retriever*: Dense retrievers [102] take advantage of the contextualized embedding representation of the text to understand the usage or context of words in a passage better. Dense retrievers typically index the embedding of a passage or jointly the question and passage into a dense space. Then an embedding-based similarity matching is performed to retrieve the relevant passages close to the query in the dense embedding space. Similar to a sparse retriever, a re-ranker is often employed to find the final retrieved passage.

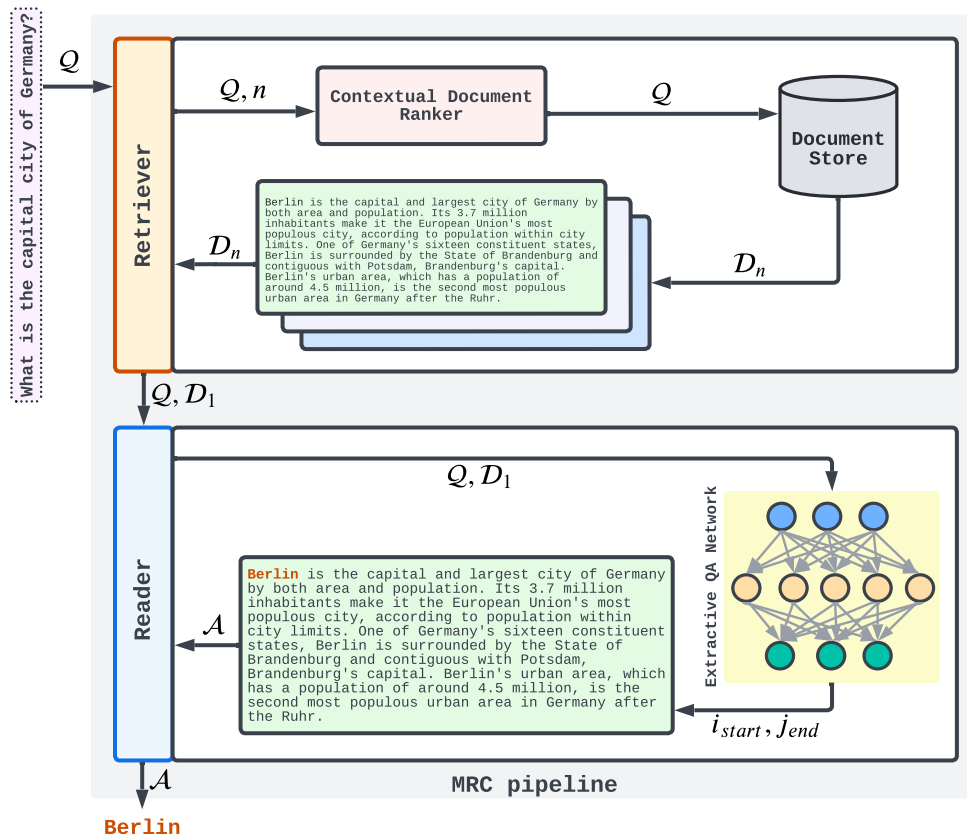


Figure 2.7: Architecture of a span prediction-based machine reading comprehension system.

- **Multi-step Retriever:** Multi-step retriever [103], also known as an iterative retriever, completes the retrieval task in multiple steps. The retrieval is primarily done in three steps. Firstly, a set of documents are retrieved based on the original question. Secondly, the original question is reformulated into a new query, usually in the form of dense embedding or natural language. Finally, the training or retrieval process is stopped based on a specified heuristic (i.e., number of iterations, fixed number retrieved documents). Multi-step types of approaches are effective for tasks where multi-hop reasoning is required.

Although dense retrievers obtain better performance, they are more resource intensive and have high latency compared to sparse retrievers [104]. The set of retrieved documents is then set to a *Document Reader* for the answer extraction task.

### Document Reader

The task of a document reader is to predict the span within the document that answers the given question. The reader module takes the question and top-ranked documents as input and predicts the start and end index of the answer span. Document readers can be divided into two primary categories:

- **Extractive Reader:** Extractive readers aim to predict the exact span of the answer. Typically, for

a given question, these systems learn to predict the start and end indices of the answer from the retrieved documents [15, 105, 106].

- *Generative Reader*: Generative readers learn to generate the answer in a sequence-to-sequence manner [107, 108]. However, generative readers suffer from syntax error and incoherence issues, unlike the extractive reader [109].

Within the scope of this thesis as conversational systems, we explored knowledge-based task-oriented dialogue systems [9], unsupervised question answering over knowledge graphs [110], and machine reading comprehension [111] systems.

## 2.4 Evaluation Metrics

Human evaluation is considered to be the most effective way of obtaining accurate quality measurements. However, obtaining a human judgment is both time-consuming and cost-intensive. Thus, automatic evaluation metrics have become a significant area of research for assessing system performance. Recent automatic evaluation metrics can be categorized into two primary types: word-overlap-based and embedding-based metrics. BLEU [32], METEOR [33], and ROUGE [112] are word-overlap-based metrics, whereas BERTScore [34] and MoverScore [113] are embedding-based metrics. We briefly discuss the widely used metrics below.

**BLEU.** BLEU [32] was originally designed for evaluating machine translation (MT) systems. It computes  $n$ -gram similarity score between words from reference  $x$  and translated  $y$  sentences. Specifically, it calculates the geometric mean of the  $n$ -gram precisions  $p_i$  and multiplies the result by a brevity penalty constant  $\gamma$  to obtain the final BLEU score. BLEU is computed as follows:

$$BLEU = \gamma \exp(\sum_{i=1}^n w_i \log p_i),$$

$$\gamma = \begin{cases} 1 & \text{if } |y| > |x| \\ \exp(1 - \frac{|x|}{|y|}) & \text{otherwise,} \end{cases} \quad (2.7)$$

where  $|x|$  and  $|y|$  denotes the number of words in  $x$  and  $y$ , respectively. A positive weight  $w_i$ , summing up to 1.0, is used as a multiplication factor for computing the BLUE score. Four-gram is widely adopted in BLUE to measure the similarity between sentences. However, as BLEU works at the word-level, it is not effective for capturing semantic similarity between sentences.

**ROUGE.** Recall-Oriented Understudy for Gisting Evaluation ROUGE [112] was developed for automatic evaluation of the summarization task. ROUGE is available in four variants: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N computes the  $n$ -gram recall between a set of reference summaries and a candidate summary. A Longest Common Sub-sequence (LCS) based F1 measurement is used in ROUGE-L to compute the similarity between two summaries, whereas ROUGE-W utilizes weighted LCSes for the measurement. A skip-bigram-based pair matching is used in ROUGE-S to handle arbitrary gaps within a sentence. Overall, ROUGE demonstrates a higher correlation to human evaluation compared to BLEU and METEOR when it comes to assessing summary quality.

**METEOR.** Addressing the shortcomings of BLEU, the metric METEOR [33] was proposed that can capture the order of words when computing the similarity score. Besides, it performs word-to-word matches, considering the stem or synonyms as the same word. For handling word orders, METEOR introduces a penalty term  $\beta$  in the final computation. METEOR is computed as follows:

$$\begin{aligned} \text{METEOR} &= F_{mean} * (1 - \beta), \\ F_{mean} &= \frac{10PR}{9P + R}, \\ \beta &= 0.5 * \left( \frac{\text{number of chunks}}{\text{number of matched uni-grams}} \right)^3, \end{aligned} \quad (2.8)$$

where,  $P$  and  $R$  refer to uni-gram precision and uni-gram recall, respectively. For multiple reference sentences, the score is computed across all the reference sentences with respect to the predicted sentence. Finally, the maximum score is set as the final evaluation score. Although METEOR improves the evaluation scores over BLUE, it is still unable to capture the semantic variations between sentences.

**Language Models as Evaluators.** Based on the hypothesis that the next utterance generation relates to the utterances in the dialogue history in open-domain dialogue systems, an evaluation metric is proposed [114] where language models are employed as evaluators. It considers two consecutive utterances and compute a coherence score between them as follows:

$$P(U) = \prod_{i=m+1}^{m+n} p(U_{m+n} | U_i, U_{n+1}, \dots, U_{m+n-1}), \quad (2.9)$$

where  $m$  and  $n$  denote the history and target utterance length, respectively. Here,  $U$  is the target utterance. Finally, a two level nested aggregation is performed to compute the final score. The first level aggregates word-level scores where the second level aggregates utterance-level scores. The final score is formally computed as follows:

$$LM_U = \sum_{u=1}^{|U|} \left( \frac{\sum_{w=1}^{|W|} p(w = w)}{|W|} \right), \text{ where } W \in U, \quad (2.10)$$

where  $|U|$  and  $|W|$  denote number of utterances in the dialogue and number of words in an utterance, respectively. This approach demonstrates higher correlation to human judgement in evaluating open-domain dialogue systems over other open-domain dialogue evaluators.

**Earth Mover's Distance.** The Earth Mover's Distance (EMD) estimates the amount of work required to transform a probability distribution into another [41]. Inspired by the EMD, in NLP, the transportation problem is adopted to measure the amount of work required to match the system-generated hypothesis sentence with the reference sentence [115, 113]. Let us define the reference as  $\mathcal{R} = \{r_1, r_2, \dots, r_p\}$  and the hypothesis as  $\mathcal{H} = \{h_1, h_2, \dots, h_q\}$ , where  $r_i$  and  $h_j$  indicates the  $i$ -th and  $j$ -th word of the reference and hypothesis, respectively. The weight of the word  $r_i$  and  $h_j$  are denoted as  $m_i$  and  $n_j$  respectively. Then, the total weight distribution of  $\mathcal{R}$  and  $\mathcal{H}$  is  $m_\Sigma = \sum_{i=1}^p m_i$  and  $n_\Sigma = \sum_{j=1}^q n_j$ , respectively. Here, the sentence-level and normalized TF-IDF score of a word are considered

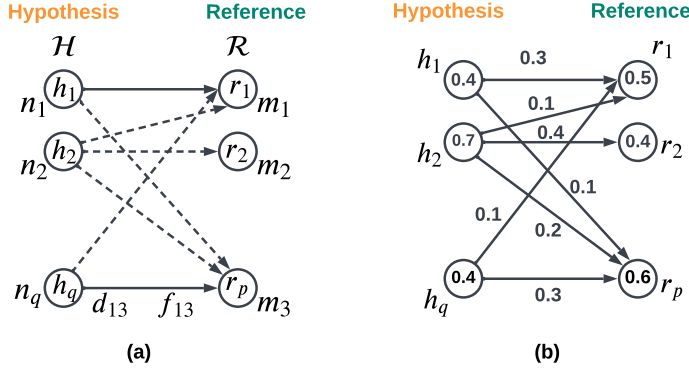


Figure 2.8: Figure (a) depicts a high level overview of the Earth Mover's Distance, where the weight-flow constraints are demonstrated in Figure (b).

as the word's weight. Formally, EMD can be defined as:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})} \sum_{i=1}^p \sum_{j=1}^q d_{ij} f_{ij}}{\min(m_{\Sigma}, n_{\Sigma})} \quad (2.11)$$

where  $d_{ij}$  is the distance between the words  $r_i$  and  $h_j$  in the space and  $\mathcal{F}(\mathcal{H}, \mathcal{R})$  is a set of possible flows between the two distributions that the system tries to optimize. In Equation 2.11,  $EMD(\mathcal{H}, \mathcal{R})$  denotes the amount of work required to match the hypothesis with the reference. The optimization is done following four constraints:

$$\begin{aligned} f_{ij} &\geq 0 & i = 1, 2, \dots, p \text{ and } j = 1, 2, \dots, q, \\ \sum_{j=1}^q f_{ij} &\leq m_i & i = 1, 2, \dots, p, \\ \sum_{i=1}^p f_{ij} &\leq n_j & j = 1, 2, \dots, q, \\ \sum_{i=1}^p \sum_{j=1}^q f_{ij} &= \min(m_{\Sigma}, n_{\Sigma}) \end{aligned} \quad (2.12)$$

The first constraint indicates that each flow must be non-negative. The second constraint limits the total weights flowing from  $r_i$  to less than or equal to  $m_i$ . Similarly, the third constraint restricts the total weights flowing from  $h_j$  to less than or equal to  $n_j$ . The final constraint indicates that the total flow of weights must be equal to the minimum weight distribution. Figure 2.8 depicts the EMD for a given hypothesis-reference pair. Word mover distance (WMD) [115] and MoverScore [113] incorporate concepts from EMD to assess the semantic similarity between reference and hypothesis sentences. In this thesis (Chapter 8), we exploit RoMe and make changes to the EMD algorithm to adapt the concept for computing the semantic similarity between a hypothesis-reference pair.

**Tree Edit Distance.** Trees are among the most studied data structures in computer science. The tree comparison method is used in a wide range of areas, including image analysis, compiler design, and computational biology. In 1989, Zhang and Shasha [116] proposed *Tree edit distance (TED)*,

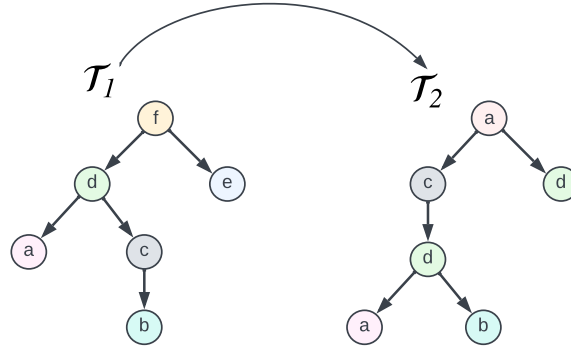


Figure 2.9: Tree transformation.

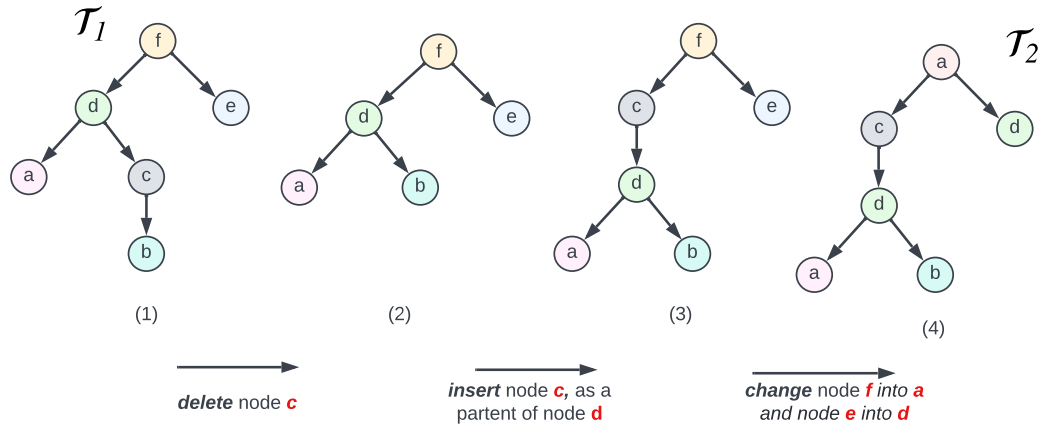


Figure 2.10: TED operations required for transforming tree  $\mathcal{T}_1$  into  $\mathcal{T}_2$ .

which computes the minimum number of operations required to transform one tree into another. During the transformation, the performed operations are: Change, Delete, and Insert. Figure 2.9 illustrates a tree before and after the transformation. The tree edit distance is formally described below in detail.

Let  $\mathcal{T}$  be a rooted and labeled tree, where  $m \in \mathcal{T}$  is a node of the tree  $\mathcal{T}$ . We denote  $\mathcal{T}(m)$  as a sub-tree of the Tree  $\mathcal{T}$  with a root  $m$ . Let  $n$  is a descendent of  $\mathcal{T}(m)$ , where  $q$  is a node to the left of  $n$ . Now, we define  $\mathcal{X}$  as the set of finite alphabets used as the labels of the tree nodes. Let  $\zeta$  be a cost function such that  $\zeta : (\mathcal{X} \times \mathcal{X}) \rightarrow (\theta, \theta)$ , which represents the cost of edit operation. Formally, for any  $a, b, c \in \mathcal{X}$  the following constraints are met:

$$\begin{aligned} \zeta(a, b) &\geq 0, \zeta(a, a) = 0, \\ \zeta(a, b) &= \zeta(b, a), \\ \zeta(a, c) &\leq \zeta(a, b) + \zeta(b, c). \end{aligned} \tag{2.13}$$

Let  $\mathcal{F}$  be a forest, where  $\mathcal{F} - m$  means the deletion of node  $v$  from the forest  $\mathcal{F}$  and  $\mathcal{F} - \mathcal{T}(m)$  means deletion of all the descendants of the sub-tree  $\mathcal{T}(m)$  including the node  $m$ , from the forest  $\mathcal{F}$ . Let  $\Omega$  be an empty tree and  $\theta \notin \mathcal{X}$  be a special blank label where  $\mathcal{T}_\theta = \mathcal{X} \cup \theta$ . Let  $\mathcal{T}_2$  and  $\mathcal{T}_2$  be labeled

trees. The tree edit distance can be formally defined as:

$$\phi(\mathcal{T}_1, \mathcal{T}_2) = \min_{\zeta(Q)} \{Q \text{ is a set of operations for transforming } \mathcal{T}_1 \text{ into } \mathcal{T}_2\}. \quad (2.14)$$

A recursion-based tree edit distance was proposed by Klein *et al.*[117]. Let  $\mathcal{F}_1$  and  $\mathcal{F}_2$  be forests. The TED computation can be formally defined as:

$$\begin{aligned} \phi(\Omega, \Omega) &= 0, \\ \phi(\mathcal{F}_1, \Omega) &= \phi(\mathcal{F}_1 - m, \Omega) + \zeta(m \rightarrow \theta), \\ \phi(\Omega, \mathcal{F}_2) &= \phi(\Omega, \mathcal{F}_2 - n) + \zeta(\theta \rightarrow n), \\ \phi(\mathcal{F}_1, \mathcal{F}_2) &= \min \begin{cases} \phi(\mathcal{F}_1 - m, \mathcal{F}_2) + \zeta(m \rightarrow \theta), \\ \phi(\mathcal{F}_1, \mathcal{F}_2 - n) + \zeta(\theta \rightarrow n), \\ \phi(\mathcal{F}_1(m), \mathcal{F}_2(n)) + \phi(\mathcal{F}_1 - \mathcal{T}_1(m), \mathcal{F}_2 - \mathcal{T}_2(n)) + \zeta(m \rightarrow n). \end{cases} \end{aligned} \quad (2.15)$$

The time complexity of the algorithm is  $O(|\mathcal{T}_1|^2 |\mathcal{T}_2| \log |\mathcal{T}_2|)$ . There have been other researches that focus on improving the optimization of the tree edit distance algorithm [118, 119].

In this thesis, we exploit the TED algorithm in Chapter 8 to capture the syntactic difference between two sentences. Specifically, we calculate the syntactic dissimilarities between two natural language sentences by computing the tree edit distance of their corresponding dependency trees. In computational linguistics, dependency and constituency trees are used to represent syntactic dependencies between words in a sentence. However, unlike the constituency tree, a dependency tree can represent non-adjacent and non-projective dependencies in a sentence, which frequently appear in spoken language and noisy text. That leads us to prefer dependency trees over constituency trees for evaluating NLG output.

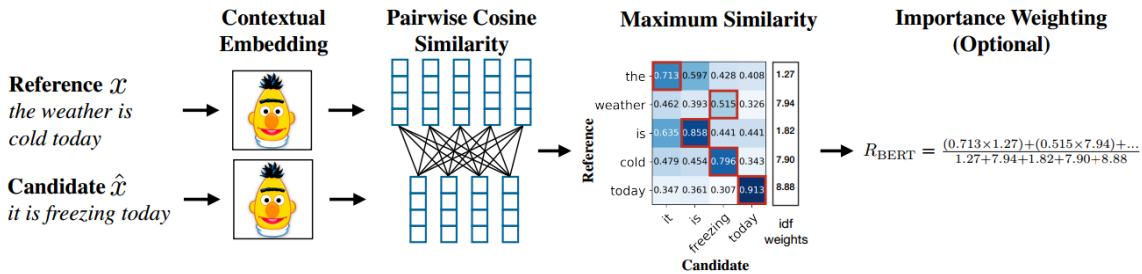


Figure 2.11: BERT-Score (image from Zhang *et al.* [34]).

**BERTScore.** BERTScore [34] is an embedding-based evaluation metric that performs contextualized embedding-based greedy matching to obtain a maximum similarity score. For a reference

sentence  $x$  and predicted sentence  $y$  the BERTScore,  $BERT_F$  is computed as follows:

$$\begin{aligned} BERT_F &= 2 \frac{BERT_P \cdot BERT_R}{BERT_P + BERT_R}, \\ BERT_P &= \frac{1}{|y|} \sum_{y_j \in y} \max_{x_i \in x} x_i^\top y_j, \\ BERT_R &= \frac{1}{|x|} \sum_{x_i \in x} \max_{y_j \in y} x_i^\top y_j, \end{aligned} \tag{2.16}$$

where,  $x_i$  and  $y_j$  denote the  $i$ -th and  $j$ -th token of the reference and predicted sentence, respectively. Figure 2.11 demonstrates a high-level overview of BERTScore, where  $BERT_R$  is denoted as  $R_{BERT}$ . An all-pair cosine similarity score is computed to obtain  $BERT_P$  and  $BERT_R$  scores. Since the value of cosine similarity ranges between a scale of  $[-1,1]$ , normalization is applied as follows to obtain the final evaluation score:

$$BERT_F = \frac{BERT_F - l_b}{1 - l_b} \tag{2.17}$$

where  $l_b$  is an empirical lower bound computed from the Common Crawl monolingual datasets<sup>6</sup>. For consistency, the same normalization technique is applied to obtain both  $BERT_P$  and  $BERT_R$ . Similar to BERTScore, popular metrics such as BLEURT [36] and WE\_WPI [35] also employ contextualized word embedding for computing the evaluation score.

Within the scope of this thesis, following previous works, we employ *BLEU*, *METEOR*, *BERTScore*, and *MoverScore* to evaluate dialogue systems. In this thesis, we investigate task-oriented dialogues; therefore, we do not use *Language Models as Evaluators* as the metric is suited for open-domain dialogue evaluation. Furthermore, we exploit Tree edit distance in our proposed metric when addressing research question RQ4.

---

<sup>6</sup><https://commoncrawl.org/>



---

## Related Work

---

### 3.1 Dialogue Systems

Most dialogue systems can be divided into task-oriented and non-task-oriented systems. The development of contemporary dialogue systems is briefly summarized below.

#### 3.1.1 Task-oriented Dialogue Systems

The majority of task-oriented conversation systems are pipeline-based or end-to-end systems. Pipeline-based systems are modular and aim to improve a specific module or component. Recent research focuses on enhancing task-oriented dialogue performance by providing a better dialogue state tracker [87, 120], user intent recognition module [121], system action generator [122], and improved dialogue generation [123, 11, 12]. However, due to the inter-dependencies between components, each component's error propagates across the whole system. The propagation of errors eventually affects the overall performance of the system.

End-to-end task-oriented-dialogue systems alleviate the human effort in designing components. These kinds of systems typically train several components jointly in a sequence-to-sequence manner. Recent research proposed techniques to jointly train dialogue state tracker and generation models [124]. A separate direction of research focuses on jointly training dialogue action and generation [125]. In a different work, Zhang *et al.* [126] proposed techniques for jointly training dialogue management and generation. In a separate direction of research, latent knowledge reasoning [127] and projecting the action into a latent space was introduced to optimize the dialogue generation [128]. Incorporating knowledge graph and dialogue history as a sequence and training a generative system for dialogue generation were also explored by [20, 12]. These approaches propose knowledge encoding techniques for effectively incorporating external knowledge into the generation process. Besides, API calls are also utilized to get rid of components such as dialogue state tracking [129]. In a multi-task setting, Hosseini *et al.* [10] proposed a sequence-to-sequence model that jointly trains by optimizing a multi-task loss for the dialogue generation process. This approach considers all the tasks as a single sequence prediction problem.

More recently, pre-trained language models were widely adopted for dialogue generation. Pre-trained language models contain rich embedding representations of the text, which are good at capturing the context. Causal language models such as GPT-2 [38] and CTRL [130] are typically employed for

learning the next-word distribution in the dialogue generation process. These models are pre-trained on large dialogue corpora such as Reddit dump [58], Twitter posts [131], and later utilized in different parts of the dialogue generation process. However, existing systems face two issues when they try to generate dialogues in a multi-domain setting. Firstly, they are unable to capture the underlying semantics of a knowledge graph, such as the relationship between entity and relation. This leads frequently to incorrect and inappropriate dialogue generation [31]. Secondly, they lack the ability to encode dynamic knowledge in a multi-domain setting, resulting in noisy dialogues [8]. Generally, integrating a knowledge base into the learning process and generating correct and coherent dialogues at the same time is a challenging task.

### 3.1.2 Non-task-oriented Dialogue Systems

Existing non-task-oriented (open domain) dialogue systems can be divided into three primary categories: *retrieval-based systems*, *generative systems*, and *ensemble systems*. Non-task-oriented dialogues are also known as *chit-chat systems*.

Retrieval-based open domain dialogue systems aim to retrieve the correct dialogue from a set of pre-defined responses. Depending on single-turn and multi-turn settings, the complexity of these systems vary [132, 26]. Early researches focus on non-neural network-based approaches (i.e., support vector machine) [133]. Later, feed-forward network, CNN, and RNN are employed to match candidate response and dialogue history [134, 27, 26]. User interaction matching-based networks were also proposed for retrieving the most relevant dialogue [135, 136]. In a different approach, ranking loss based retrieval optimization method was proposed to improve the models' performance [137]. More recently, pre-trained language-based models are employed for the retrieval task [45, 58]. This approach significantly improved the state-of-the-art performance over the former methods. The conversations in retrieval-based dialogue systems are often coherent and fluent since they are constructed from manual efforts. However, they lack the capabilities of performing conversations on out-of-domain topics.

The early dialogue systems in the history of conversational AI were mostly rule-based generative dialogue systems [1, 2]. In recent years, with the development of deep learning algorithms, generative dialogue system have gained a lot of attention in the NLP community. Majority of the recent open-domain dialogue systems are developed and trained to work in a sequence-to-sequence manner. Early approaches utilized LSTM models to generate dialogues [138, 139], typically provided with external knowledge [140, 141]. LSTM-based sequence-to-sequence models are further utilized in several research works to learn semantic dependency [23] and generate persona-based dialogues [135]. Later, GRU-based dialogue systems become popular that performs utterance aggregation to understand the context for dialogue generation [142]. In a different work [143], a knowledge injection method is proposed to learn facts for the dialogue generation. More recently, Transformer-based approaches achieved state-of-the art performance on non-task-oriented dialogue generation [8, 12]. Transformer-based language models are typically trained on a large corpora to capture a wide range of dialogue patterns [5, 24]. These pre-trained models are additionally utilized to identify emotion [144] or factual correctness [25] in the dialogue.

A combination of both retrieval-based and generative dialogue systems is proposed for developing an improved conversational system [145, 146]. In a different line of research, multi-modal dialogue systems have become popular, facilitating speech and image data for an accurate dialogue generation [147].

Within the scope of this thesis, in Chapter 4, we introduce a new task-oriented dialogue system. In contrast to the existing approaches, our proposed system employs knowledge embedding and attention masking techniques to embed structural knowledge into a language model for generating informative and engaging dialogues.

## 3.2 Question Answering Over Knowledge Graphs

Question answering over knowledge graphs involves three key steps: 1) Entity linking, 2) Relation linking, and 3) Answer extraction. We discuss recent works related to these three steps below.

### 3.2.1 Entity Linking

The entity linking task entails identifying the entity mentions in the question and linking them to the corresponding knowledge graph entities. Recent works on entity linking primarily focused on first detecting entity mentions in the question based on text similarity. Then, they link these mentions to the correct entity in the knowledge using entity labels as well as other features such as entity type information [148, 149]. Several other studies focused on training entity mention detection and entity disambiguation together to perform entity linking [150, 151]. However, in order to train these systems, it is necessary to have datasets with annotated entity mention boundaries. Recently, natural language processing has reached a new height of success with the emergence of Transformer-based [37] pre-trained language models [39, 40]. In the context of question answering, pre-trained language models have been widely studied for the entity linking task [152, 153].

A BERT-based [39] entity mention technique was proposed by Boros *et al.* [154] to identify entity mention and classify the entity type. The authors utilized link probability scores between anchors and Wikipedia pages to sort candidate entity mentions and followed [155] to perform entity disambiguation. In a similar work [156], BERT is also used to recognize entity mentions where person, location, and organization entities from German Wikipedia are transformed into English entities. Here, entities are indexed into a dense space to generate candidate entities. A *Random Forest* model is later employed to rank the final set of entities. However, this process obtains decreased performance due to entity loss caused by the translation process. Multiple research [157, 158] index Wikidata labels and perform ElasticSearch<sup>1</sup> to generate candidate entities for the entity linking task.

In another approach, an encoder-decoder attention model is leveraged in an end-to-end manner to handle long entity labels and implicit entities of Wikidata knowledge graph [159]. A separate work [160] proposed a joint entity mention and linking algorithm that leverages multiple context embeddings to compute candidate entity score. A deep convolutional neural network later utilizes the score to find the final entity. A pointer network-based [161] end-to-end model was to perform entity mention detection and linking [162].

Entity linking is a widely studied topic. More recent works on English entity linking approaches are explained in detail by [163]. Unlike the recent work, this thesis utilizes a BERT [45] model pre-trained for name entity recognition (NER) task to detect entity mentions. This thesis takes Wikidata as the target knowledge graph and indexes Wikidata entity labels into a dense space for candidate entity generation. Finally, a relation-linking guided entity disambiguation technique is proposed to predict

---

<sup>1</sup> <https://www.elastic.co/elasticsearch/>

the final linked entity (discussed in Chapter 5). Another research [164] proposes local compatibility and semantic similarity-based statistical entity linking mechanism.

### 3.2.2 Relation Linking

Relation linking is another challenging task in KGQA since it requires complex language inference capabilities. The relation linking task entails linking the surface form of a relation phrase to a predicate of a knowledge graph. Both supervised and distantly supervised approaches have been explored for the relation linking task [149, 165]. Several works focus on candidate generation techniques for relation linking [166, 167, 168]. They perform text-similarity over a dictionary such as PPDB [169] and PATTY [170], built from patterns mined from large text corpora. Connection density of a knowledge graph for relation linking was introduced by Dubey *et al.* [14]. A different research leverages English morphology and alignment model for relation linking [171]. Abstract meaning representation and Transformer-based models are also utilized to rank [172] and disambiguate [173] relations.

In a different research, systems use already linked entities from the preceding step to perform relation linking, utilizing the structural information of the knowledge graph [174]. Unseen relation linking has also been studied recently, where the model needs to predict relations which are not seen during the training step [175]. Similarly [176, 177] modeled joint learned knowledge graph embedding for entity linking, where the linked relation information is used additionally to perform disambiguation among the candidate entities. In a disparate research, a zero-shot methodology has also been used to investigate relation linking [178].

Relation linking is particularly challenging since relation mentions in a natural question are frequently implicit [171, 172]. A relation path ranking method is proposed to link the surface form of relation to relation path in knowledge graph [179]. The authors leverages gated mechanism to align word embedding and structural information space.

In contrast to previous works, in this thesis, we propose a zero-shot relation linking technique that does not require training data. We leverage a pre-trained natural language inference model to develop the zero-shot relation linker (discussed in Chapter 5).

### 3.2.3 Answer Extraction

The two most prevalent methodologies for the answer entity extraction sub-task are semantic parsing-based and retrieval-based methods. Semantic parsing-based methods transform the natural question into a logical form which is then utilized to fetch the answer entities from the target KG [180, 181]. On the contrary, retrieval-based methods use the entity and relation extracted from the natural question to obtain the answer entities from the KG [182, 183]. In another direction of research, a graph neural network-based method for KGQA has been proposed by Sorokin *et al.* [22], while other approaches fetch candidate SPARQL queries using the entities and predicted relations and re-rank them using neural network-based methods [97, 184]. More recently, a message-passing based system for the KGQA task has been developed, where a confidence score is propagated throughout the knowledge graph, computed by input question parsing and matching [185].

Unlike the previous research, this thesis proposes two techniques for extracting answer entities. An unsupervised method where the entity and relations are already linked (discussed in Chapter 5) and a generative system to construct SPARQL query directly from a natural question (discussed in Chapter 6). Existing SPARQL query generation methods can be divided into three primary categories:

1) Manual and semi-automatic, 2) Template-based, and 3) Generative approaches. Recent works related to these methods are summarized below.

*Manual and Sem-automatic Approaches:* The early research on SPARQL query generation primarily focused on hand-crafted query construction [186, 187, 188, 189, 190, 191]. In these approaches, SPARQL queries were manually designed to test the coverage and inference capabilities of ontology systems. A different research direction emphasised on query generation from datasets [192, 193]. Görlitz et al. [192] carefully explored an RDF dataset and defined a set of query characteristics for the query selection purpose. The authors employed a query generation heuristic to predict the final SPARQL representation, which checks all possible combinations of query patterns based on the defined query characteristics. In another paper, Qiao et al. [193] proposed a technique to construct a synthetic graph from a given RDF graph employing three separate algorithms to generate various types of SPARQL queries. However, the algorithm-generated queries are limited by six triples and can have at most two attributes. An ontology-based semi-automatic method was proposed by Dibowski et al. [194], where a user interface is provided to modify or select relevant concepts from the ontology for generating the SPARQL query. Nevertheless, manual efforts make it difficult to adapt these systems for large scale knowledge bases such as Wikidata [16] and DBpedia [195].

*Template-based Approaches:* Recently, to alleviate the manual efforts, a schema-based SPARQL query generation has received significant research attention [196, 197, 198]. These approaches aim to generate an intermediary schema representation (template) of the SPARQL query. The slots in the SPARQL schema are then filled up based on the defined heuristics to rank and obtain the final SPARQL query. In another schema-driven approach, Zenz et al. [199] proposed a method to bind domain-specific keywords to generate query template. The authors followed an incremental refinement strategy to obtain the final SPARQL query from a query template. In a different work, Unger et al. [196] introduced a method to generate SPARQL query templates, utilizing the semantic structure of the question. More recently, a classification based approach was proposed by Vollmers et al. [98], where semantically similar types of questions are classified to obtain a query template. Nevertheless, the query generation task remains limited due to the fixed number of schema. To extend the coverage of these systems for additional types of questions and queries, manual schema creation is required.

*Generative Approaches:* In a different direction of solutions, Soru *et al.* [200] developed a sequence-to-sequence system that utilizes bi-directional LSTM [201] for generating SPARQL templates. An interpreter reconstructs the final SPARQL query from the query template using rule-based heuristics. However, the method cannot handle out-of-vocabulary words in the test set and lacks understanding of the question, thus frequently generating incorrect graph patterns in the query. Recently, Zafar et al. [97] exploited syntactic features to train a SPARQL query ranking model leveraging Tree-LSTM [202]. The similarity score between syntactic features of a question and a query is used for ranking candidate queries. Since the syntactic features are not learned and are only used to compute tree-similarity, the system does not generalize well when encountering an unseen question. Furthermore, the system needs to find all query patterns from the extracted sub-graph to predict the final SPARQL query, otherwise it fails to generate the query.

Despite substantial research efforts, adapting these systems to arbitrary KGs and handling low-frequency question types is difficult. The main challenges can be summarized as follows: (1) SPARQL templates are usually created manually or semi-automatically by domain experts, which is both time consuming and cost intensive, (2) The query templates are tailored to a particular KG, which results in

potentially changing of the whole template set when the underlying graph is changed, (3) The extension of template sets to handle new question types is performed manually or semi-automatically, and (4) In pipeline-based approaches, the SPARQL generation module is dependent on the performance of the preceding modules (i.e., entity and relation linkers as well as ranking algorithms) and, thus, suffer from error propagation. In contrast to the approaches mentioned above, this thesis aims to encode the linguistic features of an NLQ and leverages a pre-trained language model to both learn the graph patterns and generate SPARQL queries.

### 3.3 Machine Reading Comprehension

Most machine reading comprehension systems have two major components: 1) *Document retriever* and 2) *Document reader*. Few works additionally utilize question reformulation techniques for generating search queries prior to the core MRC task [203, 204]. MRC datasets play a vital role in the development of these components. We briefly discuss the major research works on MRC below.

#### 3.3.1 Document Retriever

A document retriever can be considered as an information retrieval (IR) system that retrieves relevant documents with respect to the question. Recent document retrievers can be categorized into three major approaches: 1) *Sparse Retriever*, 2) *Dense Retriever*, and 3) *Multi-step Retriever*.

##### Sparse Retriever

TF-IDF [100] and BM25 [101] are the most widely used sparse retriever adopted by various IR systems. Sparse representations are used to measure term frequency in these classical IR approaches. DrQA [15] utilized bi-gram matching and TF-IDF score to retrieve a set of Wikipedia articles relevant for answering given a natural language question. BERTserini [106] utilized various modalities of a document (i.e., sentence-level, paragraph-level, and document-level) matching to index information during the pre-processing step. The best-performing retrieval approach was paragraph-level indexing. However, the words in the question may not appear in the relevant paragraphs for answering the question. Addressing this issue, dense retrievers are later proposed that consider contextualized word embedding for retrieval.

##### Dense Retriever

With the increasing attention on Transformer-based language models, dense representation of the text has boosted the performance of natural understating. In a widely used dense retrieval approach, dual-encoders are employed to retrieve relevant paragraphs. Both the question and paragraphs are encoded independently in these approaches [205, 206, 102, 207]. The encoding is primarily done using BERT [45]. Another approach jointly encodes the question with a document with a focus on inter-token interaction [208, 209, 210]. Interaction-based retrieval approach enables the system to capture rich interaction and understand the relevant documents better. However, such approaches are resource intensive and difficult to adapt for large-scale data.

A hybrid approach, where the dual-encoder and interaction based techniques are combined for improved performance and efficiency [211, 212, 213]. In general, dense retrievers are computationally

expensive, despite the effectiveness and performance gain in the retrieval task. To alleviate this issues, questions and documents are encoded separately and stored in the memory to perform offline retrieval. In this case, the retrieval trade-offs the performance to get rid of computational overheads.

#### **Multi-step Retriever**

Multi-step retrievers, firstly, retrieve relevant documents based on the original query utilizing sparse [70, 214] or dense retrievers [215, 216]. Next, the original question is reformulated into natural language query [70] or dense embedding [217]. The reformulated queries are then finally utilized to find a refined set of documents and stop the iterative process of retrieval [215, 216, 218, 219]. In multi-step retrieval approaches, systems are generally trained till a fixed number of iterations or a defined number of documents are retrieved [71, 70, 219, 214].

#### **3.3.2 Document Reader**

Document readers can be primarily divided into two categories: Extractive and Generative readers. Given a question, extractive readers focus on predicting the answer span from the retrieved paragraphs. Utilizing a paragraph selector component DS-QA [220] first predicts an answer span from the retrieved paragraph. The paragraph selector selects the paragraphs based on the probability of being the answer in those paragraphs. DPR [102] employs a BERT-based reader to compute the probability of being a token at the start or end position of the answer. In a different approach, the reader system forms an intermediary graph and then learns to extract the answer from the intermediary graph [221]. A different focus of work optimize the reader component to extract the answer span by a joint training mechanism [222].

The generative reader learns to generate the answer span from the provided question and retrieved paragraph. S-Net [223] first identifies the span where the answer might exist and then incorporate a sequence-to-sequence model to generate the final answer. In a different approach, RAG [224] employs a BART model to generate the final answer, taking the question and retrieved paragraph as input. However, generative readers usually suffer from spelling mistakes, change of meaning by adding additional text before and after the answer span [109, 225].

#### **3.3.3 MRC Dataset**

Dataset is one of the crucial factors for developing intelligent systems. Machine reading comprehension datasets are broadly classified into two types: open-domain and domain-based. Open domain datasets are typically constructed from Wikipedia [226, 19], books from various domains [227], news portals [228], social-media posts [229]. On the other hand, domain-specific datasets are constructed from reliable sources such as official reports [230], articles [111], official web-portals [111], and domain-specific books [231]. Besides, numerous MRC datasets are constructed, focusing on developing conversational [231, 232] and multi-modal systems [233] across multiple domains.

Following the previous works, this thesis adopts a dense retriever and utilizes a BERT-based reader to train a machine reading comprehension system on a climate domain dataset (discussed in Chapter 7).

### 3.4 Evaluation of Generative Systems

A potentially good evaluation metric is one that correlates highly with human judgment. Among the unsupervised approaches, BLEU [32], METEOR [33] and ROUGE [112] are the most popular evaluation metrics traditionally used for evaluating NLG systems. ROUGE [112] is used mostly for evaluating document summarization. There are four variants of ROUGE, amongst which in this paper, we use ROUGE-L as a baseline. Unlike BLEU, ROUGE-L does not need a predefined value for  $n$ -gram matching as it calculates word-overlap based on the longest common sequence (LCS). Although these metrics perform well in evaluating machine translation (MT) and summarization tasks, [234] shows that none of the word overlap based metrics is close to human level performance in dialogue system evaluation scenarios. In a different method, word embedding-based metrics are introduced for evaluating NLG systems [235, 236]. Several unsupervised automated metrics were proposed that leverage EMD; one of them is word mover's distance (WMD) [115]. Later, [236] proposed an evaluation metric, incorporating WMD and word-embedding, where they used word alignment between the reference and hypothesis to handle the word-order problem. Recently, [35] introduced an EMD-based metric WE\_WPI that utilizes the word-position information to tackle the differences in surface syntax in reference and hypothesis. In a disparate approach, Transformer-based [37] language models are modeled as evaluators. Two consecutive utterances are considered to compute a coherence score between them. Another work [237], treats the dialogue quality assessment task as an anomaly detection problem. The authors investigated four dialogue modeling approaches and found negative correlation with human judgement.

Several supervised metrics were also proposed for evaluating NLG. ADEM [238] uses a RNN-based network to predict the human evaluation scores. With the recent development of language model-based pre-trained models [239] proposed BERTScore, which uses a pre-trained BERT model for evaluating various NLG tasks such as machine translation and image captions. Recently, [113] proposed MoverScore, which utilizes contextualized embedding to compute the mover's score on word and sentence level. A notable difference between MoverScore and BERTScore is that the latter relies on hard alignment compared to soft alignments in the former.

In contrast to the previous methods, this thesis proposes a robust evaluation metric which focuses on handling the sentence's word repetition and passive form when computing the EMD score. Furthermore, the proposed metric trains a classifier by considering the sentence's semantic, syntactic, and grammatical acceptability features to generate the final evaluation score (discussed in Chapter 8).



---

# Generative Dialogue Systems With Structured Knowledge

---

This chapter addresses the first research question RQ1, "**Does incorporating structural information into a language model improve knowledge graph-based dialogue generation?**". In this chapter, we discuss about techniques to incorporate structured knowledge into a language model for the dialogue generation task. We study how the effective inclusion of structural knowledge influences the overall performance of a task-oriented conversation system through qualitative and quantitative evaluations. The content of this chapter is based on the following publication:

- **Md Rashad Al Hasan Rony**, Ricardo Usbeck, and Jens Lehmann. 2022. *DialoKG: Knowledge-Structure Aware Task-Oriented Dialogue Generation*. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 2557–2571, Seattle, United States. Association for Computational Linguistics.

This chapter consists of seven sections. The experiments and results are described in Section 4.5. A comprehensive analysis including case study is provided in Section 4.6. Finally, Section 4.7 summarizes the contributions of this chapter.

## 4.1 Introduction

In this chapter, we propose a novel task-oriented dialogue system, named DialoKG that employs structural information of the knowledge graph into a language model (LM) for generating informative dialogues (see Figure 4.1 and 4.2(a)). For this purpose, we exploit GPT-2 [38] - a language model developed based on a stack of Transformer decoders [37]. Specifically, we introduce a novel structure-aware multiple embedding layer-based knowledge embedding technique that constructively embeds the underlying relationship between the knowledge triples. DialoKG interprets the knowledge as a knowledge graph; therefore, separate embedding layers for word token, entity, triple and token type enable the system to capture the graph features (e.g., subject, relation and object). This enables the system to generate correct and human-like dialogues and prevents generating erroneous responses such as "*4 miles is located at 792 Bedoin Street Starbucks away*". Furthermore, the ability to correctly capture the relationship in the knowledge graph eliminates the need for template-based or sketch-based response generation.

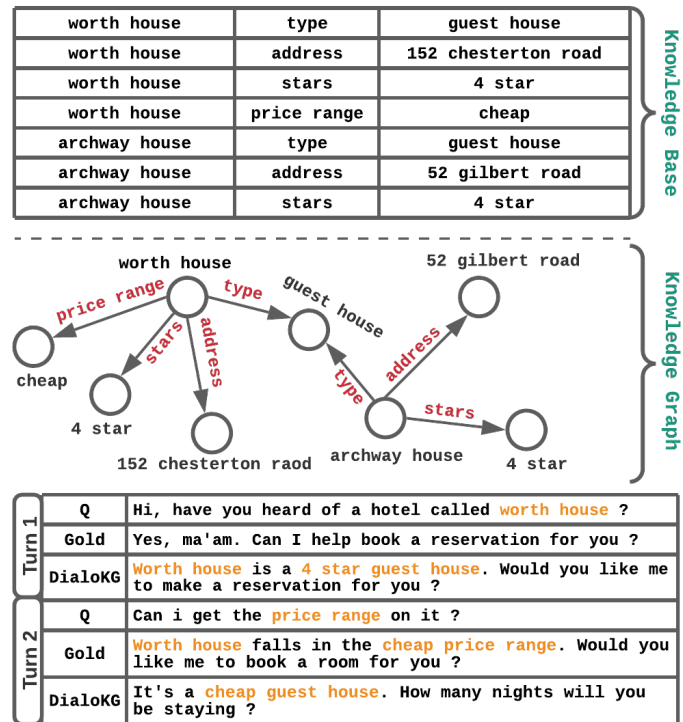
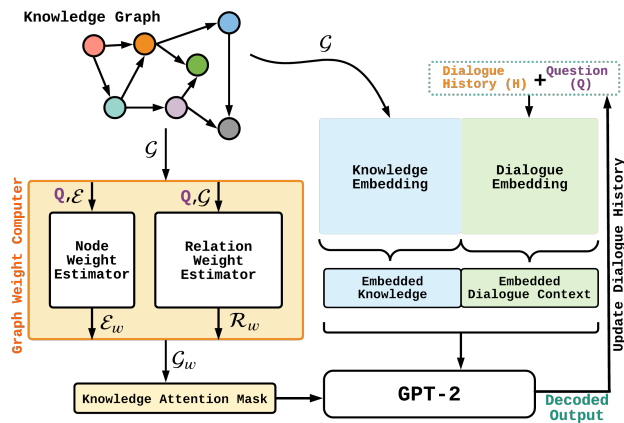


Figure 4.1: An illustration of knowledge-based multi-turn dialogue where DialoKG models the knowledge base as a Knowledge Graph. The user utterance is denoted by **Q**, the ground-truth response by **Gold**, and the words in orange are knowledge graph entries.

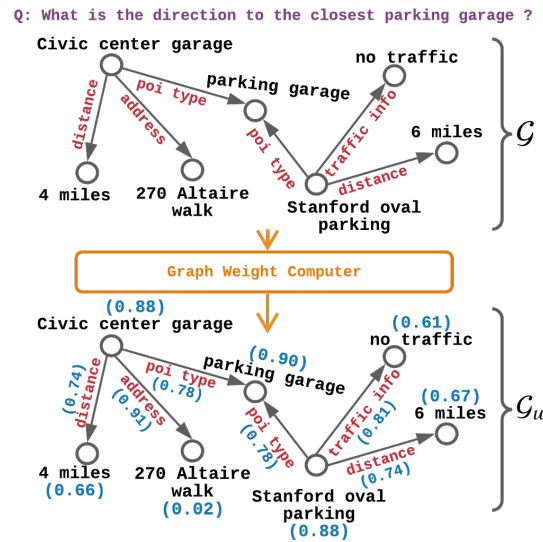
In order to guide the decoder on relevant parts of the knowledge graph, we propose a new knowledge attention masking method. For constructing the knowledge attention mask, in each dialogue turn, a weighted graph is computed in two steps: 1) Entity weights are computed using a pre-trained language model that estimates the importance of an entity for the given utterance, and 2) relation weights are computed based on the concept of graph convolution networks (GCN) [240]. Both steps take the user utterance into consideration, i.e., the obtained weighted graph is question specific. A set of triples is then selected based on the most relevant entities and relations of the weighted graph to construct a knowledge attention mask for the language model. This allows the masked language model to focus on relevant graph triples. We hypothesise that this leads to the generation of more accurate responses and enhance the model’s capabilities of understanding the domain and task.

To assess the performance of DialoKG, we conduct experiments on three public benchmarks: SMD [241], CamRest [242] and Multi-WOZ 2.1 [243]. We evaluate the system generated responses using both human and automatic metrics. Furthermore, we analyse impact of the individual components on the overall performance to verify the effectiveness. Our experimental results show that DialoKG outperforms state-of-the-art models in knowledge-grounded dialogue generation and can generate human-like responses. We made our code publicly available <sup>1</sup>.

<sup>1</sup><https://github.com/rashad101/DialoKG>



(a) System architecture.



(b) Weighted-graph computation.

Figure 4.2: A high-level overview of DialoKG is shown in Figure (a). Figure (b) depicts the input and output of the *Graph Weight Computer* module of DialoKG.

## Contributions

- A knowledge embedding technique, that embeds the structural information of a knowledge graph effectively.
- A knowledge graph-weighted attention masking method that guides the masked language model to attend to the relevant knowledge entries for generating correct and informative responses.
- A novel task-oriented dialogue system, effectively employing knowledge into a language model.

## 4.2 Problem Definition

DialoKG aims to generate informative responses given a dialogue history, a question and a knowledge graph. We define the dialogue history  $\mathcal{H}$  as a set of turns between two speakers, such that  $\mathcal{H} = \{U_1, S_1, \dots, U_t, S_t\}$ , where  $U_i$  and  $S_i$  are the sequences of words in turn  $i$ . We assume that the knowledge is stored in a multi-relational knowledge graph  $\mathcal{G}$ . Here,  $\mathcal{G}$  is a set of triples  $\mathcal{T}$  such that  $\mathcal{T} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ , where  $\mathcal{E}$  is the set of entities and  $\mathcal{R}$  the set of relations. A triple  $\mathcal{T} \in \mathcal{G}$  is denoted as  $(s, r, o)$  in which  $s \in \mathcal{E}$  and  $o \in \mathcal{E}$  denote the subject and object entities, respectively, and  $r \in \mathcal{R}$  is the relation between them. We use the terms "Knowledge Graph" and "Graph" interchangeably throughout this chapter. Furthermore, we denote the user utterance of the current dialogue turn as  $Q$ . A GPT-2 [38] language model is used in this chapter to generate responses. However, any Transformer decoder-based LM can be used. Formally, the probability distribution of generating a response by the language model is defined as:

$$p(S_t | \mathcal{H}, Q, \mathcal{G}) = \prod_{i=1}^n p(s_i | s_1, \dots, s_{i-1}, \mathcal{H}, Q, \mathcal{G}) \quad (4.1)$$

Here,  $S_t$  is the generated response in turn  $t$  and  $n$  is the maximum length of the generated response.

## 4.3 Approach: DialoKG

### 4.3.1 Knowledge and Dialogue Embedding

DialoKG takes a knowledge graph  $\mathcal{G}$ , dialogue history  $\mathcal{H}$ , and the current user utterance  $Q$  together as input and constructs a single input sequence as depicted in Figure 4.3. The first part of the sequence contains graph related information (i.e., subject, relation, and object) and the latter part dialogue specific information such as dialogue history ( $\mathcal{H}$ ) and the current user utterance ( $Q$ ).

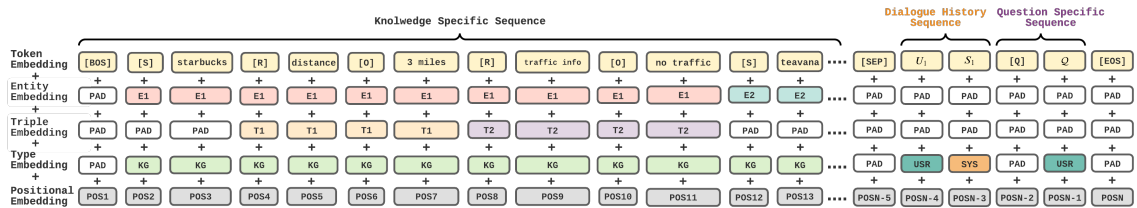


Figure 4.3: An illustration of knowledge and dialogue embedding techniques.

**Knowledge Specific Embedding.** To infuse structural information, DialoKG employs entity embedding, triple embedding and type embedding, besides the usual word token and positional embedding. Such an embedding technique allows the system to encode the knowledge graph structure. To do this, knowledge graph triples are linearized into a sequence as input, as depicted in Figure 4.3. To facilitate order invariance of the knowledge embedding, we shuffle the order of the graph triples in the input sequence during training. In the token embedding layer [S], [R] and [O] are special tokens to separate subject, relation and object of a triple from each other in the sequence. Entity and

triple embedding layers embed entity and triple-level information of the word token. For instance, ENT1 in the entity embedding layer indicates that the corresponding words in the token embedding layer are related to the first subject, which is *starbucks* in this case. Likewise, T1 and T2 in the triple embedding layer indicate that the corresponding words in the token embedding layer are related to the first and second triple, respectively. Finally, the type embedding indicates that the corresponding tokens are from the knowledge graph as opposed to the dialogue history.

**Dialogue Specific Embedding.** The dialogue specific part of the input sequence is separated from the knowledge specific part by a [SEP] token in the token embedding layer. Furthermore, the user utterance/question ( $Q$ ) of the current turn is separated by a [Q] token from the dialogue history. The type embedding layer stores information about whether the corresponding utterance is from the user or system. This way, the decoder can use information about typical dialogue turn patterns.

The positional embedding in both knowledge and dialogue embeddings encodes the position of each word token in the sequence. Finally, embeddings from all five layers are summed up as depicted in Figure 4.3. *Layer Normalization* [244] is then applied to obtain the final embedding representation of the complete input sequence. It normalizes the embedding representation of layers, which restricts the weights of the learning network from exploding.

We argue that the proposed design pattern of forming a single sequence and specifying each item in the input sequence further with additional embedding layers can improve the system’s understanding of the task and domain.

### 4.3.2 Knowledge Attention Mask Construction

To notify the decoder about the relevant KG triples for answering the current user question, a knowledge graph weighted-attention mask is constructed. Prior to the construction of the knowledge attention mask, a weighted-knowledge graph,  $\mathcal{G}_w$  is first computed by a *Graph Weight Computer* module, where the entity and relation weights are computed independently. Figure 4.2(b) illustrates the weighted graph computation. We discuss the components of the *Graph Weight Computer* module below.

**Entity Weight Estimator:** A pre-trained language model RoBERTa [245], is used to compute the entity weights, similar to [246]. Each entity  $E_i \in \mathcal{E}$  of graph  $\mathcal{G}$  is concatenated with the user utterance  $Q$  to obtain the probability score from the language model.

$$E_{iw} = LM_{head}(LM_{enc}([Q; E_i])) \quad (4.2)$$

In Equation 4.2,  $LM_{head} \circ LM_{enc}$  represents the probability of the entity  $E_i$  computed by the language model. We consider  $E_{iw}$  as the weight of the entity  $E_i$ , which represents the relevance of the entity for the given user utterance  $Q$ .

**Relation Weight Estimator:** We follow [240, 247] and leverage the concept of GCN to obtain the relation weight. In contrast to the previous works, our proposed relation weight estimator transforms the input graph into an undirected graph, where the relations are considered as nodes of a graph. This transformation technique allows the relation estimator to obtain a score for each relation. The graph transformation is demonstrated in Figure 4.4(a). The relation weight is computed as follows:

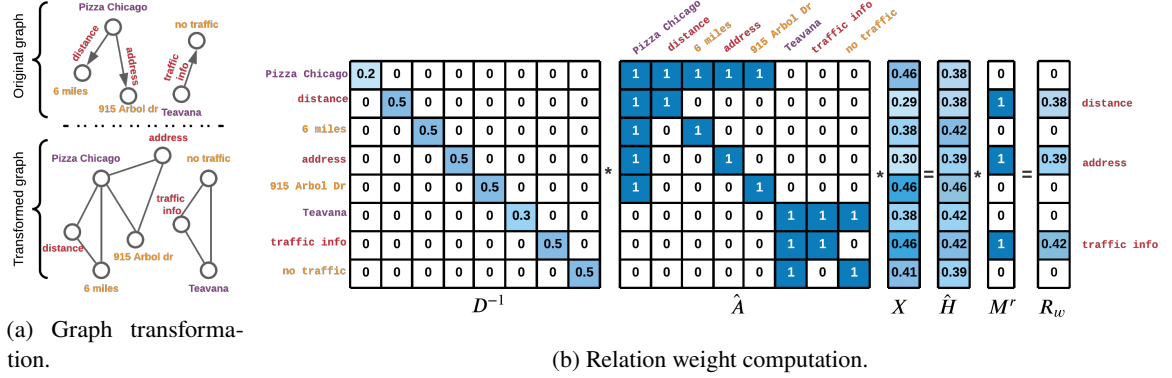


Figure 4.4: For the graph in Figure (a) and the question "Find me the quickest route to the restaurant?" the computation of the relation weight is shown in Figure (b), where  $\hat{A} = A + I$ .

$$\begin{aligned}
 R_w &= \tilde{H}M^r, \\
 \tilde{H} &= D^{-1}(A + I)X
 \end{aligned}
 \tag{4.3}$$

Here,  $D^{-1}(A + I)$  computes the row-normalized adjacency matrix, where  $D$  and  $A$  are respectively the degree matrix and adjacency matrix of the graph  $\mathcal{G}$  as depicted in Figure 4.4(b) and  $I$  is the identity matrix. Let  $d_g = |\mathcal{E}| + |\mathcal{R}|$  be the total number of entities and relations in the graph  $\mathcal{G}$ , then  $D, A, I \in \mathbb{R}^{d_g \times d_g}$ . A feature vector  $X \in \mathbb{R}^{d_g \times 1}$  is obtained by computing the cosine similarity between the embedding of knowledge graph entries (entities and relations) and the embedding of question. Furthermore, a relation mask  $M^r \in \mathbb{R}^{d_g \times 1}$  is constructed by setting a value of 1 and 0 to the positions that correspond to relations and entities, respectively, to attend to the values that correspond to the relations only. Finally, values that correspond to the entities in  $\hat{H}$  are masked out by multiplying with  $M^r$  to obtain final relation weights  $R_w \in \mathbb{R}^{d_g \times d_g}$ .

**Mask Construction:** We construct the mask in two steps. First, we construct a mask for the dialogue-specific sequence (dialogue history and question specific sequence) following the causal language model masking strategy. In a causal language model mask, already predicted tokens are only shown, where masking the future tokens in the output sequence. Then, we construct the knowledge attention mask based on the *Knowledge Specific Sequence*. Finally, we combine these two masks to construct the final mask.

We use the normalized score of  $R_w$  and  $E_w$  for constructing the knowledge attention mask. Based on the normalized entity and relation scores, first, we sort a variable number of knowledge triples that fits into the models' input size (depicted by *Knowledge Specific Sequence* in Figure 4.3), keeping the dialogue-specific sequence fixed. Furthermore, we filter out irrelevant knowledge triples and select triples based on top-k entities and relations from the *Knowledge Specific Sequence*. Here,  $k$  is a hyper-parameter which we chose from a range of  $[0, \max(|\mathcal{E}|, |\mathcal{R}|)]$ , based on the validation score.

Finally, based on the selected  $\hat{\mathcal{E}}$  and  $\hat{\mathcal{R}}$ , the knowledge attention mask is constructed as follows:

$$M_{i,j}^{kg} = \begin{cases} 0, & \text{if } ((s_i \vee o_i) \in \hat{\mathcal{E}}) \wedge (r_i \in \hat{\mathcal{R}}) \\ -\infty, & \text{otherwise} \end{cases}$$

Here  $r_i$ ,  $s_i$ , and  $o_i$  correspond to the relation, subject, and object entity of triple  $\mathcal{T}_i$ . Any position that corresponds to the value of  $-\infty$  results in 0 after computing the *softmax* during the attention computation (discussed in the next sub-section). The final mask  $M \in \mathbb{R}^{n \times n}$  is obtained by appending the dialogue-specific mask with the knowledge attention mask, where  $n$  is the sequence length. This type of masking strategy ensures that the provided knowledge during each token prediction for a particular output sequence always remains the same. Padding is added to adjust the dimension of the metrics.

### 4.3.3 Decoder

A Transformer [37] based GPT-2 [38] model is used for generating the response. The attention, computed in each of GPT-2’s heads is formalized as follows:

$$\begin{aligned} \text{Attn}(Q, K, V) &= \text{softmax}\left(\frac{1}{\sqrt{d_k}}(QK^T) + M\right)V, \\ H_i &= \text{Attn}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \quad (4.4)$$

where,  $\text{Attn}(\cdot)$  computes the masked attention,  $H_i$  is the  $i$ -th head,  $d_k = d_m/h$ . Here,  $d_m$  is the dimension of the model where  $h$  the number of heads.  $Q$ ,  $K$  and  $V$  are query, key and value where  $W_i^Q$ ,  $W_i^K$ ,  $W_i^V$  are trainable parameters. The objective of the model is to minimize the negative log-likelihood  $\mathcal{L}$  for next-token prediction. For a dialogue dataset  $D = \{D_1, D_2, \dots, D_j\}$ , we formally define  $\mathcal{L}$  as follows:

$$\mathcal{L}(D) = - \sum_j^{|D|} \sum_i^n \log p(s_i^j | s_1^j, \dots, s_{i-1}^j, \mathcal{H}^j, \mathcal{Q}^j, \mathcal{G}^j), \quad (4.5)$$

where  $n$  is the maximum response length and  $\mathcal{H}^j, \mathcal{Q}^j, \mathcal{G}^j \in D_j$ . Top-k sampling [248] decoding is used to generate the next word token at each time step, during the inference.

## 4.4 Experimental Setup

### 4.4.1 Data

We evaluate DialoKG on three publicly available knowledge-grounded and task-oriented dialogue datasets: Stanford Multi-Domain dataset (SMD) [241], CamRest [242] and Multi-WOZ 2.1 (MWOZ) [243]. SMD consists of three domains: weather, navigation, and calendar. MWOZ contains five domains: train, hotel, restaurant, taxi and attraction. We use the splits provided with the datasets for train, validation, and test. Each dialogue is provided with a knowledge graph. Table 4.1 shows the statistics of the benchmark datasets.

Dataset	#Dialogues	#Utterances	Avg. Length of Utt.	#Utt. with Entities	Avg. #Entities per Utt.
SMD [241]	3,031	15,928	9.22	4430	2.96
CamRest [242]	676	2,744	11.72	2366	2.43
MWOZ [243]	2,877	19,870	16.68	6241	2.06

Table 4.1: Dataset statistics.

#### 4.4.2 Hyper-parameter Settings

Throughout this chapter, we use the GPT-2 [38] model with 117M parameters. AdamW [249] with  $\epsilon = 1e-8$  and learning rate of  $6.25e-5$  is employed as optimizer. GELU [250] is used as activation function. The best hyper-parameters for each dataset were found using grid search and based on the results on the validation set. We run all experiments on a distributed training setting with 10 GPUs, each with 12 GB of memory.

We report the hyper-parameters used to train DialoKG in Table 4.2 for SMD, CamRest, and MWOZ. GPT-2 specific hyper-parameters are also reported in Table 4.2. All the hyper-parameters are found after a grid search and evaluation on the validation set. We sample learning rate from  $\{6.25e-01, 6.25e-04, 6.25e-05\}$  and maximum history token and knowledge token from  $\{128, 256, 384, 512\}$ .

	SMD	CamRest	MWOZ
Learning rate	6.25e-05	6.25e-04	6.25e-05
Adam epsilon	1e-08	1e-08	1e-08
Batch size	4	4	4
Gradient accumulation steps	4	4	4
Max history turn	4	4	1
Maximum history token	128	256	128
Maximum knowledge token	384	256	384
Top relations	7	7	6
Top entities	7	5	7
Epochs	40	25	30

Table 4.2: Training parameters.

For both training and evaluation, we use a batch size of 4. Hyper-parameters used during the inference are reported in Table 4.3. We used 12 NVIDIA TitanX GPUs, each with 12GB of memory to train models. It took 30, 18 and 45 minutes to train on SMD, CamRest and MWOZ data.

	SMD	CamRest	MWOZ
Temperature	0.68	0.85	0.18
Top-k	6	8	10
Top-p	0.9	0.9	0.9
Maximum response length	100	80	120
Top entities	7	7	6
Top relations	7	5	7

Table 4.3: Decoding parameters.



Model	SMD			CamRest			MWOZ		
	BLEU	MoverScore	Ent. F1	BLEU	MoverScore	Ent. F1	BLEU	MoverScore	Ent. F1
GLMP [20]	13.9	54.2	59.6	15.1	57.2	58.9	6.9	51.2	32.4
MLM [251]	17.0	64.0	54.6	15.5	57.0	62.1	-	-	-
Ent. Const. [252]	13.9	53.8	53.7	18.5	65.9	58.6	-	-	-
GPT2+KE [8]	17.4	<u>66.4</u>	59.8	18.0	65.8	54.9	<b>15.0</b>	<u>60.9</u>	<u>39.6</u>
TTOS [253]	17.4	59.8	55.4	20.5	67.0	61.5	-	-	-
DF-Net [254]	14.4	56.3	62.7	-	-	-	9.4	54.2	35.1
EER [255]	17.2	60.9	59.0	19.2	66.1	65.7	13.6	57.2	35.6
FG2Seq [256]	16.8	60.2	61.1	20.2	66.6	66.4	<u>14.6</u>	58.4	36.5
CDNet [257]	<u>17.8</u>	61.1	<u>62.9</u>	<u>21.8</u>	<u>67.8</u>	<u>68.6</u>	11.9	55.8	38.7
<b>DialoKG</b>	<b>20.0</b>	<b>70.6</b>	<b>65.9</b>	<b>23.4</b>	<b>70.4</b>	<b>75.6</b>	12.6	<b>62.6</b>	<b>43.5</b>

Table 4.4: Performance of DialoKG and baseline models on three benchmark datasets. Best scores in **bold** and second-best underlined.

### 4.4.3 Evaluation Metrics

**Automatic Metrics.** Following the baseline models, we use BLEU [32] and Entity F1 score [241] as automatic evaluation metrics. The Entity F1 score represents the model’s capability of generating knowledge grounded responses. It computes the F1 score between the set of entities present in the ground truth and system-generated responses. Several studies [258, 234] on evaluation metrics suggest that word-overlap based metrics such as BLEU are insufficient for evaluating natural language generation (NLG) systems. Hence, we use MoverScore [113] as addition metric to evaluate the semantic similarity between the system generated response and the ground truth. We compute both MoverScore and BLEU scores on the sentence level.

**Human Evaluation.** To assess the quality of the system-generated responses, we conduct a human evaluation based on the following criteria: 1) Naturalness: how human-like and fluent the generated responses are, and 2) Correctness: how correct the knowledge-grounded responses are. We asked three annotators (two from Computer Science (CS) and one from a non-CS background) who are not part of this research work to evaluate the quality of the system-generated responses. We randomly sampled 90 dialogues in total from the benchmark datasets and asked annotators to evaluate the system-generated responses given the ground truth response and the knowledge graph triples on a scale of [1,5] (higher the score, the better it is). The inter-annotator agreement score (Cohen’s kappa  $\kappa$ ) of the annotated data is 0.82. The human evaluation process is explained in detail in § 4.5.1.

### 4.4.4 Baselines

We compare DialoKG with the following state-of-the-art methods: **GLMP** [20], **MLM** [251], **Ent. Const.** [252], **DF-Net** [254], **CDNet** [257], **GPT2+KE** [8], **TTOS** [253] and **EER** [255]. Most of these approaches adopt memory networks to generate knowledge grounded dialogues, whereas **GPT2+KE** [8] directly embeds the knowledge graph into the model’s parameters and **TTOS** [253] proposed a reinforcement learning-based framework.

Model	Naturalness	Correctness
EER [255]	3.27	3.61
FG2Seq [256]	3.33	3.87
CDNet [257]	3.53	3.94
<b>DialoKG</b>	<b>4.33</b>	<b>4.01</b>

Table 4.5: Human evaluation results.

## 4.5 Results

### 4.5.1 Quantitative Results

We conduct both quantitative and qualitative analyses to assess system-generated responses. Table 4.4 summarizes the performance of DialoKG with respect to the baseline models. It is evident that DialoKG outperforms the baseline models significantly in Entity F1 score on CamRest, which contains mostly knowledge-grounded dialogues about restaurant reservations. A high Entity F1 score of 75.6 on CamRest shows DialoKG’s ability to generate knowledge-grounded with high accuracy. Although DialoKG achieves an improved Entity F1 score on the MWOZ dataset, it has a lower BLEU score since MWOZ often contains lengthy responses. However, the high MoverScore across all datasets demonstrates that DialoKG can generate highly semantically similar responses. We report the domain-wise results for SMD and MWOZ in Table 4.6 and Table 4.7 respectively. Baseline model’s results are reported from [257] and [8]. The MWOZ dialogue dataset contains conversations on the following domains as reported in the baseline works: attraction, restaurant, and hotel. The domain-wise results demonstrate that DialoKG achieves improved performance in almost all domains in a multi-domain setup. This demonstrates DialoKG’s capacity to handle a dynamic knowledge graph.

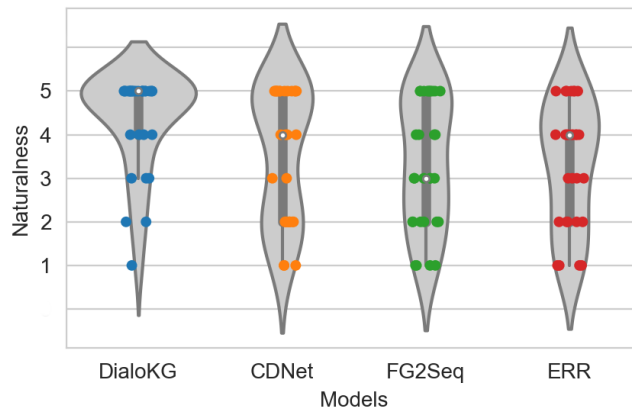


Figure 4.5: Distribution of human evaluation scores.

### 4.5.2 Qualitative Results

We obtain human evaluation scores (naturalness and correctness) for the closest three models. Results in Table 4.5 show that our proposed dialogue system can generate more human-like responses. An

Models	BLEU	MoverScore	Entity F1	Schedule	Navigate	Weather
GLMP [20]	13.9	54.2	59.6	72.5	54.6	56.5
MLM [251]	17.0	64.0	54.6	66.7	46.9	56.0
Ent. Const. [252]	13.9	53.8	53.7	55.6	54.5	52.2
GPT2+KE [8]	17.4	66.4	59.8	72.6	53.5	57.7
TTOS [253]	17.4	59.8	55.4	63.5	45.9	64.1
DF-Net [254]	14.4	56.3	62.7	73.1	57.9	57.6
EER [255]	17.2	60.9	59.0	71.8	52.5	57.8
FG2Seq [256]	16.8	60.2	61.1	73.3	56.1	57.4
CDNet [257]	17.8	61.1	62.9	75.4	56.7	61.3
<b>DialoKG (Ours)</b>	<b>20.0</b>	<b>70.6</b>	<b>65.9</b>	<b>77.9</b>	<b>58.4</b>	<b>72.7</b>

Table 4.6: Domain-wise results on SMD dataset.

Models	BLEU	MoverScore	Entity F1	Attraction	Restaurant	Hotel
GLMP [20]	6.9	51.2	32.4	24.4	38.4	28.1
MLM [251]	-	-	-	-	-	-
Ent. Const. [252]	-	-	-	-	-	-
GPT2+KE [8]	<b>15.0</b>	60.9	39.6	<b>43.3</b>	37.1	33.4
TTOS [253]	-	-	-	-	-	-
DF-Net [254]	9.4	54.2	35.1	28.1	40.9	30.6
EER [255]	13.6	57.2	35.6	43.0	34.3	35.7
FG2Seq [256]	14.6	58.4	36.5	37.2	38.9	34.4
CDNet [257]	11.9	55.8	38.7	38.9	41.7	36.3
<b>DialoKG (Ours)</b>	12.6	<b>62.6</b>	<b>43.5</b>	39.8	<b>46.7</b>	<b>37.9</b>

Table 4.7: Domain-wise results on MWOZ dataset.

improved score is also achieved in terms of correctness, reflecting DialoKG’s ability to generate highly accurate dialogues. Furthermore, Figure 4.5 shows the distribution of human evaluation scores. The figure allows a better direct comparison of the individual score levels.

Figure 4.6 shows the interface of the annotation tool used to obtain human annotation scores. The interface displays a set of knowledge triples, a user utterance, the ground truth response, and a system-generated response for each point. Given the information displayed on the annotation tool, we asked the annotators to rate the system-generated responses against the ground-truth on a scale of [1,5] (higher is better). We explained the participants about the purpose of this research. The first two participants are male (over 30 years old), and the third participant is female (more than 35 years old), both with several years of experience in the domain.

## 4.6 Analysis

This section investigates the contribution of each component to the overall performance of DialoKG. Furthermore, the effectiveness and impact of knowledge embedding and knowledge attention mask are also discussed. Moreover, through a case study, we explore the advantages and limitations of DialoKG.

**Knowledge Base**

Subject	Relation	Object
chevron	distance	5 miles
chevron	traffic info	moderate traffic
chevron	poi type	gas station
chevron	address	783 arcadia pl
town and country	distance	5 miles
town and country	address	383 university ave
jacks house	poi type	friends house
jacks house	address	864 almanor ln
the clement hotel	traffic info	no traffic

**User utterance:** What is the address of chevron ?

**Ground truth:** 783 arcadia pl is the address for chevron gas station .

**System generated response:** chevron is located at 783 arcadia pl .

Naturalness  1  2  3  4  5

Correctness  1  2  3  4  5

Save
Previous
Next

Figure 4.6: The interface of the annotation tool to obtain the human annotation scores.

### 4.6.1 Ablation Study

We conducted an ablation study to investigate the contribution of major components of DialoKG. The results on CamRest in Table 4.8 demonstrate that the *ses2seq* approach achieves the lowest scores, which represents the DialoKG model without the embedding layers: entity embedding, triple embedding, and type embedding. Inclusion of the entity and triple embedding layers significantly improved model’s performance in both BLEU and Entity F1 scores. The type embedding further improved DialoKG’s performance. The significant difference in results shows the effectiveness of the proposed embedding technique. Finally, we observed a remarkable improvement in DialoKG’s overall performance after the inclusion of knowledge attention mask. Question-aware weighted-graph computation used to construct knowledge attention mask, helped the model focus on the task at the inference time.

### 4.6.2 Effectiveness of Knowledge Embedding

The proposed graph embedding technique works best in combination with the knowledge attention mask. The graph embedding design allows DialoKG to handle disconnected graphs and triples. This makes DialoKG suitable for large-scale graphs, where a cosine-similarity based triple selection may

Approach	BLEU	$\Delta$	Ent. F1	$\Delta$
DialoKG (seq2seq)	14.5	-	59.4	-
+ Entity embedding	17.7	3.2 $\uparrow$	63.0	3.6 $\uparrow$
+ Triple embedding	19.2	1.5 $\uparrow$	67.8	4.8 $\uparrow$
+ Type embedding	20.1	0.9 $\uparrow$	68.4	0.6 $\uparrow$
+ Knowledge attention mask	23.4	3.3 $\uparrow$	75.6	7.2 $\uparrow$

Table 4.8: Ablation study.

Top- $k$ (entity)	Top- $k$ (relation)	BLEU	MoverScore	Entity F1
3	5	10.8	65.3	48.2
3	7	11.0	65.4	48.9
5	5	16.9	68.0	62.1
5	7	17.4	68.1	62.5
7	5	19.3	70.4	64.4
7	7	<b>20.0</b>	<b>70.6</b>	<b>65.9</b>
All	All	15.9	67.2	59.0

Table 4.9: Effect of triple selection on the performance.

be used to fit the graph triples inside the model’s input capacity. The entity and triple embedding layers allow the model to preserve the structural information of a particular triple even though triples from different parts of the input sequence are selected based on the top- $k$  entities and relations to construct the knowledge attention mask. Overall, the graph embedding technique improves the Entity F1 score by 5.4, 9.0, and 3.7 points on SMD, CamRest, and MWOZ, respectively. This indicates the effectiveness of the proposed embedding techniques for capturing graph triples.

### 4.6.3 Impact of Knowledge Attention Mask

To understand the effect of the knowledge-graph weighted attention mask, we experiment with the triple selection process described in DialoKG’s approach. Table 4.9 shows the performance of DialoKG with selected top- $k$  entities and relations on the SMD dataset. We observe that DialoKG achieves the best performance on SMD when the top 7 entities and relations are chosen to construct the knowledge mask. Consider the question "Do you have any local coffee shops?" the ground truth is "There is Coupa, it s just 6 miles away but there is heavy traffic on our way". The ground truth contains traffic information in addition to the distance and name of the coffee shop. Selecting a high number of entities and relations increases the chance of generating such additional information related to the subject of the question. However, choosing too many entities harms the model since it is more likely to add irrelevant noise (see Table 4.9). For MWOZ, six entities and seven relations, and for CamRest, seven entities and five relations result in the best performance.

### 4.6.4 Case Study

Figure 4.7 shows two cases from the MWOZ dataset given a subset of the knowledge graph. In Case 1, we observe that in answering the user question, DialoKG correctly picked *Rice House* that serves *cheap* and *Chinese* food. However, in this case, multiple correct answers exist, e.g. *Charlie Chan* also

Knowledge Triples			Case 1			
Charlie Chan	food	chinese	User	I'd like to find a restaurant that serves cheap chinese food	BLEU	MoverScore
Charlie Chan	address	Regent street city centre	Truth	Charlie Chan is located in the centre and it is cheap! Would you like me to book it for you?	-	-
Charlie Chan	pricerange	cheap	CDNet	Charlie Chan is a chinese restaurant in the center of town.	0.0	56.5
Charlie Chan	area	center	FG2Seq	Charlie Chan is a cheap restaurant.	0.0	58.3
Yu Garden	address	529 Newmarket road fen ditton	DialoKG	Rice House is a restaurant serving chinese food in the cheap price range! Would you like me to make a reservation?	14.5	67.3
Yu Garden	pricerange	expensive	Case 2			
Yu Garden	food	chinese	User	Thank you.	BLEU	MoverScore
Rice House	address	88 mill road city centre	Truth	You're welcome . Have a great day!	-	-
Rice House	food	chinese	CDNet	You're welcome.	34.6	69.6
Rice House	pricerange	cheap	FG2Seq	Have a great day!	50.8	70.5
			DialoKG	Thank you for using our service & have a good day!	0.0	66.4

Figure 4.7: Case study: comparison between ground truth and system-generated responses.

falls into the same category of restaurant. Despite generating the correct answer based on the given knowledge and the user question, DialoKG receives a low Entity F1 score since the generated response entity does not match the ground truth. In Case 2, where the baseline systems focus on imitating the ground truth, DialoKG generates a fluent and engaging response. Despite generating a meaningful and semantically similar sentence, it obtained a BLEU score of 0.0 because of the low overlap with the ground truth response. However, a high MoverScore in both cases indicates DialoKG’s ability to generate a semantically similar response. Overall, we observe that DialoKG can generate human-like, engaging, and informative responses in a multi-turn dialogue setting.

#### 4.6.5 Influence of Dialogue History

Dialogue history is particularly crucial since it gives the model the context for generating the response. In some cases where the entity information is missing in the current user utterance, the dialogue context provides the model with enough information to perform the inference and generate the correct response. For instance, for the question, *What is the food type they serve?*, the name of the restaurant is not given in the question, but the system can infer it from the dialogue history. However, from the experiments, we found that too much dialogue context may inject noisy and irrelevant information to answer the current question, in particular for knowledge-grounded responses in MWOZ. To quantify this, we selected different numbers of dialogue turns as history for the model’s input depending on the characteristics of the dataset and visualised the result in Figure 4.8.

### 4.7 Summary

We have presented DialoKG, a novel knowledge-grounded task-oriented dialogue system improving the state-of-the-art across multiple benchmark datasets. DialoKG focuses on capturing the underlying semantics of the knowledge graph and pays attention to the relevant graph triples to understand the task and generate correct and human-like responses. The key contributions of DialoKG include 1) **Knowledge embedding technique**, that embeds the structural information of a knowledge graph effectively, and 2) **Knowledge graph-weighted attention masking**, which guides the masked language model to attend to the relevant knowledge entries for generating correct and informative responses. Finally, we demonstrated DialoKG’s ability to generate accurate, diverse, and human-like dialogues through quantitative and qualitative analysis. We performed an ablation study and studied the effect of

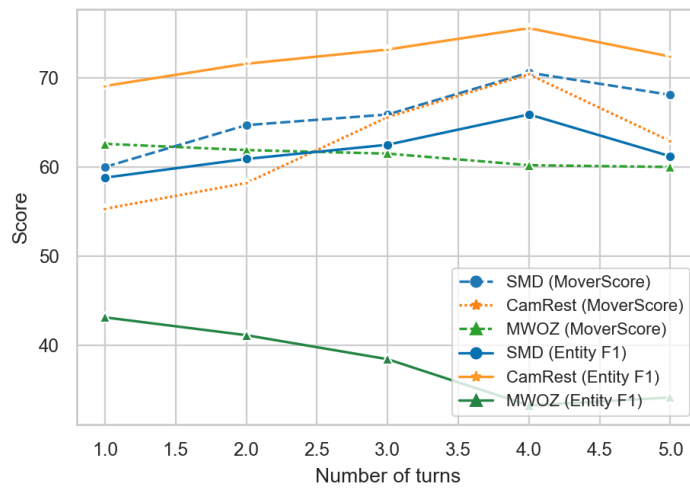


Figure 4.8: DialoKG’s performance on benchmark datasets for different number of dialogue contexts.

dialogue history, knowledge embedding, and knowledge attention masking. DialoKG answers the first research question, RQ1 (*Does incorporating structural information into a language model improve knowledge graph-based dialogue generation?*), affirmatively and demonstrates that the effective incorporation of structured knowledge into a language model significantly improves the performance of a task-oriented dialogue system.





---

# Unsupervised Question Answering Over Knowledge Graphs

---

In contrast to Chapter 4, where a small set of knowledge triples is provided in the form of a knowledge graph to a dialogue system, this chapter handles a large knowledge graph containing millions of facts for question answering. This chapter addresses the research question RQ2, "**How effective are pre-trained language models for developing an unsupervised knowledge graph-based question-answering system without training data?**". Specifically, this chapter leverages pre-trained language models in an unsupervised manner to develop a question answering over knowledge graphs without training data. A large knowledge graph Wikidata [16] containing millions of facts is utilized as a source of structured knowledge to conduct experiments to show the effectiveness of the proposed techniques. The content of this chapter is based on the following publication:

- **Md Rashad Al Hasan Rony**, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann, *Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs*, in IEEE Access, vol. 10, pp. 50467-50478, 2022, doi: 10.1109/ACCESS.2022.3173355.

This chapter is organized into six sections. Section 5.1 discusses the challenges and issues of the existing systems. Furthermore, addressing the issues, a high-level overview of the proposed system is also provided. Section 5.2 formally defines the problem statement. A detailed description of the proposed system is provided in Section 5.3. All the experiments, results, and their analysis are reported in Section 5.4. Finally, Section 5.6 concludes the chapter by summarizing the contributions.

## 5.1 Introduction

Since the advent of large-scale knowledge graphs (KG) such as DBpedia [195], Freebase [18], and Wikidata [16], KG-based systems have evolved significantly. Given a natural language question, the task of a KG-based question answering (KGQA) system is to retrieve the correct answer from the knowledge graph. Entity and relation linking are the primary sub-tasks of KGQA. These sub-tasks include determining the *surface form* (mentions in the question) of the entity and relation in the question and subsequently mapping them to the respective entity and relation in the knowledge graph. The linked entity and relation are then utilized to obtain the answer entity in the final step [184].

KGQA on both simple and complex questions is a well-researched topic [185, 259, 260]. For training, supervised systems depend heavily on knowledge graph-based question answering datasets. Reaching peak performance often requires a significant amount of training data [150, 151]. Since both data collection and training processes are time consuming and cost-intensive, this is a bottleneck in developing dataset-independent KGQA systems. Furthermore, supervised systems are often vulnerable to brittleness [261]. Since they aim to capture the underlying dynamics in the training data, they frequently fail to generalize well when tested on previously unseen data. The KGQA task is depicted in Figure 5.1, where the circular nodes indicate entities and the connecting directed lines represent the relationship between two KG entities.

To alleviate the time and effort necessary to develop a question answering (QA) system, researchers recently explored unsupervised and few-shot question answering techniques [262, 263]. Effective unsupervised **KGQA** is still a challenging research problem. Unsupervised KGQA is particularly hard because, **firstly**, large-scale knowledge graphs such as Wikidata [16] contain more than 80 million entities and a few thousand relations. Linking the entity and relation mentioned in the question to the corresponding large-scale KG entity and relation is thus a challenging task. **Secondly**, it is a standard practice to execute a query (e.g., using SPARQL) over the KG to extract answer entities [264, 184]. Query construction for this purpose adds an additional layer of difficulty.

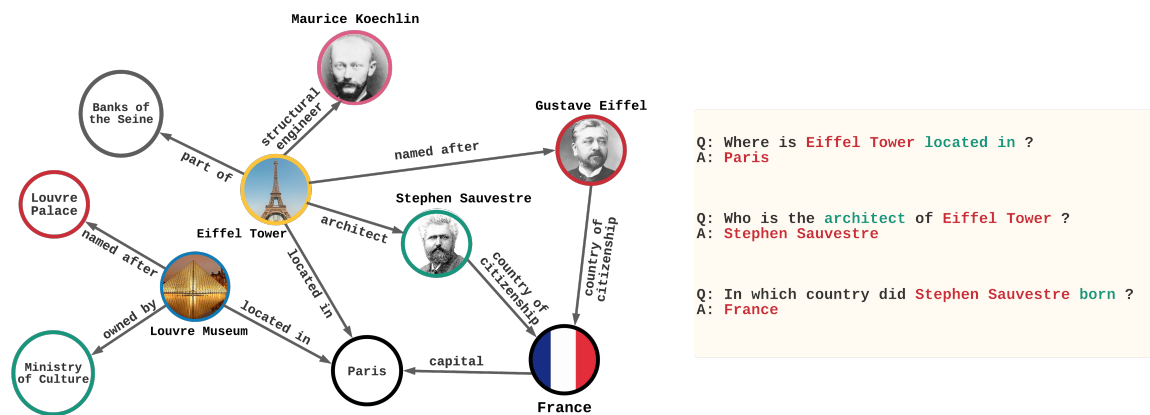


Figure 5.1: An illustration of question answering over a knowledge graph. Figure a) depicts a sub-graph of the Wikidata KG, where Figure b) demonstrates sample question-answer pairs based on the example sub-graph. In the sample question-answer pairs, the surface form of the entities and relations are in red and green, respectively.

Addressing the issues mentioned above, we propose a simple yet effective unsupervised KGQA method leveraging pre-trained language models. The primary motivation of this research is to develop a dataset-independent KGQA system, which can answer natural questions from various datasets without additional training or fine-tuning. We adopt powerful off-the-shelf language models pre-trained on named entity recognition (NER) and natural language inference tasks for the KGQA sub-tasks [39, 40]. Specifically, we split the KGQA task into three sub-tasks: entity linking, relation linking, and answer entity extraction. **Firstly**, we employ a BERT-based [39] pre-trained NER model to detect the surface form of the entity. Additionally, we pre-process and index the contextualized representation of the entities into a dense space for effective and fast candidate entity generation during the inference. The index is utilized to generate a set of candidate entities, which are then disambiguated to obtain the final predicted entity (details in Section 5.3.1). **Secondly**, by combining the 1-hop connected

relations of the entities linked in the previous step, a set of candidate relations for relation linking is created. A pre-trained BART model [40] is then applied to the candidate relations to obtain the most probable relation in a zero-shot manner (details in Section 5.3.2). **Finally**, we construct a set of  $k$ -level trees from the  $k$ -hop sub-graphs of the linked entities. Then, *tree-walking* and *tree-disambiguation* techniques are employed to extract answer entities from the constructed trees (details in Section 5.3.3).

To assess the performance of our proposed approaches, we conduct experiments on four publicly available benchmarks: LC-QuAD 2.0 [265], LC-QuAD 2.0 (KBpearl) [266], QALD-7-Wiki [267], and WebQSP-WD [22]. The empirical study confirms that our proposed system achieves a significant improvement in entity and relation linking sub-tasks. In the entity linking task, we notice an absolute increase of 4.5% on the LC-QuAD 2.0, 7.1% on the LC-QuAD 2.0 (KBpearl), and 0.1% on the QALD-7-Wiki in F1 score. The improvement in relation linking is 5.4% on the LC-QuAD 2.0 (KBpearl) in F1 score. Despite the simplicity, our proposed Tree-KGQA achieves an absolute increase of 1.4% in the F1 score over the state-of-the-art methods without training on WebQSP-WD test set. We have made our code open source<sup>1</sup>.

### Contributions

- An unsupervised entity linking method that achieves state-of-the-art (SOTA) results on LC-QuAD 2.0, LC-QuAD 2.0 (KBpearl), and QALD-7-Wiki datasets.
- A zero-shot relation linking mechanism that achieves SOTA results on the LC-QuAD 2.0 (KBpearl).
- A novel *tree-walking* and *tree-disambiguation* techniques for extracting answer entities. In particular, we propose a modular and unsupervised KGQA system that does not require any training and can be applied to any Wikidata-based KGQA dataset. Finally, we establish a new baseline for KGQA on the LC-QuAD 2.0 KBpearl dataset.

## 5.2 Problem Definition

In this section, first, we define the knowledge graph and knowledge tree. Following that, we discuss each component of our proposed Tree-KGQA system in depth.

[ **Knowledge Tree** ] A knowledge tree with  $k$ -levels  $\mathcal{T}_i^k$ , associated to an entity  $E_i$ , is a labelled and directed tree; consisting of nodes  $\Omega$  and branches  $\Psi$ , where  $\{\Omega, \Psi\} \in \mathcal{G}_i^k$ . A Forest  $\mathcal{F}$ , is denoted as the set of knowledge trees;  $\mathcal{F} = \{\mathcal{T}_1^k, \mathcal{T}_2^k, \dots, \mathcal{T}_p^k\}$  where  $p$  is the number of trees in the forest.

Given a natural language question  $Q$ , our proposed system aims to predict a set of answer entities  $\mathcal{E}^a \subseteq \mathcal{E}$  that answers the question. Table 5.1 provides an overview of the notations of the concepts covered in this research.

<sup>1</sup><https://github.com/rashad101/Tree-KGQA>

Notation	Concept
$e$	Label of the entity $E$
$\vec{e}$	Embedding representation of the entity label $e$
$m_i$	$i$ -th entity mention in the question
$E_i^m$	Linked entity for the entity mention $m_i$
$E_i^c$	A set of candidate entities with labels similar $m_i$
$\mathcal{E}^L$	A set of linked entities corresponding to the entity mentions in $Q$
$\mathcal{R}^L$	Linked relation for a given question
$h_i$	A set of relations connected to 1-hop of entity $E_i$

Table 5.1: Notation of the concepts used in Tree-KGQA.

### 5.3 Approach: Tree-KGQA

Tree-KGQA performs question answering over a knowledge graph in three steps: 1) it links the entities that appear in the question with corresponding knowledge graph entities, 2) it performs relation linking in a zero-shot manner, and 3) finally, leveraging *Tree-walking* and *Tree-disambiguation* techniques it extracts the answer from a forest. Below we provide a detailed overview of these three steps.

#### 5.3.1 Entity Linking

The entity linking task entails a) mention detection – spotting the *surface form* of the entity that appears in the question and b) mapping the detected mention to the corresponding knowledge graph entity. The steps involved in entity linking are described below.

**Mention Detection.** To detect the entity mentions in the question, we employ a BERT-large [39] model pre-trained for the named entity recognition task.

$$W_m = f(Q) \quad (5.1)$$

The function  $f(\cdot)$  in Equation 5.1, is a pre-trained BERT-large model that takes a question  $Q$  as input and predicts a set of named entity word tokens,  $W_m$  as the output. For instance, consider the question, *Which football club does lionel play for?*. The system detects *lionel* as the entity mention in this step using Equation 5.1. In the following steps, the detected entity mention is mapped or in other words linked to the corresponding knowledge graph entity.

**Entity Mapping.** We first index all the entity labels from a target KG into a dense space as a pre-processing step of entity mapping. During inference, the system generates candidate entities from the dense space for each detected entity mention from the previous step. To obtain the final linked entity from the set of candidate entities, an additional entity disambiguation step is performed in the cases where the same entity label appears more than once. The entity mapping technique is explained in detail below.

**Entity Indexing.** In this step, **firstly**, we extract all the entities from the target KG, in our case Wikidata, and store it in an *Entity store* (see Figure 5.2a). The *Entity store* contains all the Wikidata entity labels (e.g., *Lionel Messi*) and their Wikidata ID (e.g., *Q615*). **Secondly**, we encode all the

knowledge graph entity labels using Sentence-BERT [47]. Sentence-BERT captures the overall meaning of the entity label better since entity labels frequently contain multiple words in them. We obtain a vector of dimension  $1 \times 768$  for each entity label from Sentence-BERT. **Finally**, the encoded vector representations of the KG entities are indexed into a dense space using FAISS [268]. During the inference, the system utilises a hierarchical indexing algorithm *IndexHNSWFlat* from FAISS, which enables the system to generate candidate entities (see Figure 5.2b) in an optimized way [150, 151]. Given an entity span, the hierarchical indexing algorithm generates  $N$  candidate entities from the dense space based on  $k$ -nearest neighbors (KNN) approximate search.

For each detected entity span  $m_i \in W_m$ , the system performs entity linking separately. The system generates a set of  $N = 10$  candidate entities  $E_i^c = \{E_1, E_2, \dots, E_N\}$  for each entity mention  $m_i \in W_m$ , using FAISS (Figure 5.2b). Each generated candidate entity has an indexing score (from the FAISS approximate search) indicating how similar they are to the *entity mention* in the dense space. The candidate entity with the highest indexing score is then considered as the linked entity. Henceforth, a disambiguation step between the generated entity candidates is not required if all the candidate entity labels appeared once in the set.

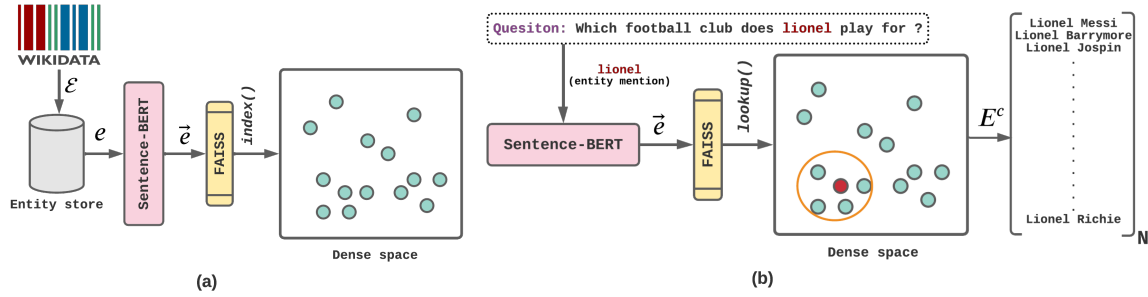


Figure 5.2: Figure (a) illustrates how the entity labels are encoded with Sentence-BERT and then indexed into a dense space using FAISS. The Indexing algorithm *IndexFlatIP* of FAISS, clusters similar entities together into the dense space. Figure (b) demonstrates the candidate entity generation procedure given a detected entity mention. Sentence-BERT is used to obtain the vector representation of the entity mention *lionel*. The encoded vector is then passed to the FAISS module that performs a lookup into the dense space and generates  $N$  candidate entities that are similar to the provided entity span, *lionel*. The red circle represents the given entity mention in the dense space, where the other circles inside the larger orange circle indicate similar entities around it.

**Entity Disambiguation.** The system performs entity disambiguation if an entity label appears multiple times in the candidate entity set. In that case, it firstly predicts a temporary relation  $\mathcal{R}_t$  using Algorithm 1. Although we develop Algorithm 1 to perform relation linking (details in Section 5.3.2), in this section we utilize Algorithm 1 to obtain  $\mathcal{R}_t$ . The question  $Q$ , and a set of all the 1-hop connected relations of the candidate entities are used as input to the Algorithm 1. As the output, Algorithm 1 predicts a relation which we denote as  $\mathcal{R}_t$  in this section. The system selects an entity with the highest similarity score from  $E_i^c$  as linked entity  $E_i^m$ , which is connected to the predicted relation  $\mathcal{R}_t$  at a distance of 1-hop in the KG.

For instance, for the question Which company’s CEO is Tim Cook?, the predicted entity mention is Tim Cook. The entity label Tim Cook appears multiple times in the set of generated candidate entities; hence, entity disambiguation is required. By utilizing Algorithm 1, *CEO* is obtained as  $\mathcal{R}_t$ .

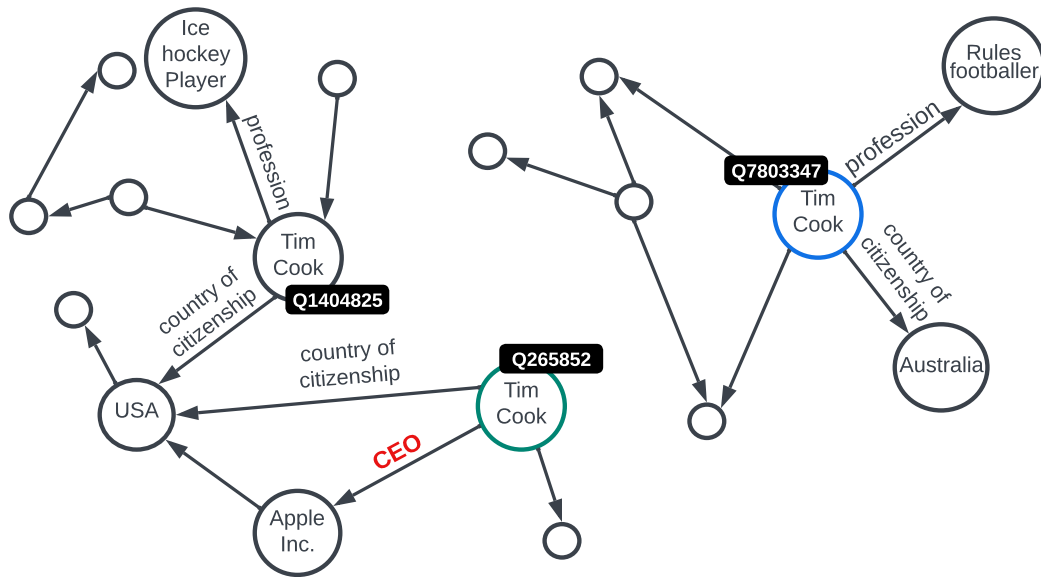


Figure 5.3: An illustration of the entity disambiguation process. A small portion of the Wikidata graph is shown for demonstration purposes.

In the generate candidate entity set, *Tim Cook* (*Q265852*) has the relation *CEO* in its 1-hop connected relations. Where the other candidate entities with the same entity label (e.g., *Tim Cook* (*Q7803347*) an Australian rules footballer, *Tim Cook* (*Q1404825*) an American ice hockey player) do not have the relation *CEO* in their 1-hop connections. Consequently, *Tim Cook* (*Q265852*), an American business executive, gets predicted as the final linked entity. In the cases where there exist multiple candidate entities with the same label, and  $\mathcal{R}_t$  in their 1-hop, the entity with the highest indexing score that contains  $\mathcal{R}_t$  in its 1-hop is selected as the linked entity. Figure 5.3 depicts a high-level overview of the entity disambiguation process.

Finally, after repeating the whole entity mapping process for each entity mention, the system produces the final set of linked entities,  $\mathcal{E}^L$  as follows:

$$\mathcal{E}^L = \bigcup_{m_i \in W_m} E_i^m \quad (5.2)$$

For the running example question, the entity mention `lionel` gets linked to the Wikidata entity, *Lionel Messi* (*Q615*).

### 5.3.2 Zero-shot Relation Linking

We model the relation linking problem as a classification task, where the system aims to link the given natural language question to one of the KG relations based on label information. In our proposed approach, we firstly generate a set of candidate relations  $\mathcal{R}^c$  from all the 1-hop connected relations of

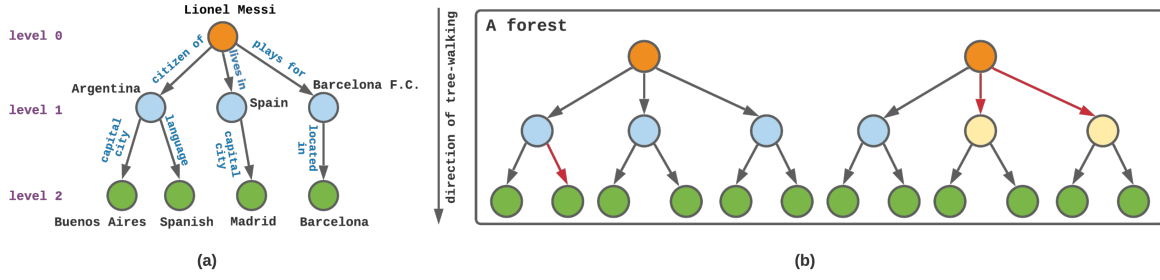


Figure 5.4: Figure a) depicts a  $k$ -level tree (with  $k=2$ ). Since a tree has many nodes and branches (edges), we present a toy example. Figure b) shows a forest consists of a set of trees constructed from the sub-graph of the linked entities. For the demonstration purpose, we show a forest consists of two trees. The red branches show the position of predicted relation in different trees. The green nodes represent the leaf nodes at level- $k$ , where the blue nodes refer to the intermediary nodes between the root and leaf nodes. Furthermore, the yellow nodes represent the predicted answer entity nodes connected by the red branches.

the already linked entities  $\mathcal{E}^L$  as follows:

$$\mathcal{R}^c = \bigcup_{E_i \in \mathcal{E}^L} h_i \quad (5.3)$$

where  $h_i$  denotes the set of 1-hop connected relations of the entity  $E_i$ . For the running example question and linked entity *Lionel Messi*, the set of candidate relations  $\mathcal{R}^c$  is  $\{\textit{citizen of}, \textit{lives in}, \textit{plays for}\}$  (see Figure 5.4a). Furthermore, we mask all the detected entity mentions in the question with a generic token  $\langle \text{ENT} \rangle$ , to obtain a *masked question* representation denoted by  $\hat{Q}$ , which football club does  $\langle \text{ENT} \rangle$  play for?. We mask the entity mentions in the question to reduce noises in the relation classification task. In Algorithm 1, the function  $\textit{maskEnt}(\cdot)$  masks the entities in the question. The system then performs zero-shot relation label classification, leveraging a pre-trained language model called BART [40], which was pre-trained for the natural language inference (NLI) task. In Equation 5.4, function  $\mathcal{Z}(\cdot)$  is a BART-large model [40] that computes the probability of being the correct relation label given the modified question ( $\hat{Q}$ ) and a set of candidate relation labels (labels of relations in  $\mathcal{R}^c$ ).

$$p(r_i | \hat{Q}, \mathcal{R}^c) \leftarrow \mathcal{Z}(\hat{Q}, \mathcal{R}^c) \quad (5.4)$$

Here,  $r_i \in \mathcal{R}^c$  is a candidate relation. Finally, we obtain the predicted relation  $\mathcal{R}^L$  as follows:

$$\mathcal{R}^L = \operatorname{argmax}_{r_i \in \mathcal{R}^c} p(r_i) \quad (5.5)$$

From Equation 5.5, the system obtains *plays for* as the predicted and linked relation  $\mathcal{R}^L$ . Algorithm 1 summarizes the relation linking task described in this section.

### 5.3.3 Answer Entity Extraction

To extract the answer entities from the knowledge graph, firstly, we build a forest utilizing the sub-graph information associated to the linked entities (obtained from Section 5.3.1). Then, we

**Algorithmus 1 : Relation Linking**


---

**Input :** A question  $Q$ , a set of candidate relations  $\mathcal{R}^{cand}$   
**Output :** A relation  $\mathcal{R}^P$

- 1  $\mathcal{R}^P \leftarrow \emptyset$
- 2  $\hat{Q} \leftarrow \text{maskEnt}(Q)$
- 3  $p(r_i|\hat{Q}, \mathcal{R}^{cand}) \leftarrow \mathcal{Z}(\hat{Q}, \mathcal{R}^{cand})$
- 4  $\mathcal{R}^P \leftarrow \text{argmax} p(r_i)$ , where  $r_i \in \mathcal{R}^{cand}$
- 5 **return**  $\mathcal{R}^P$

---

perform tree-walking over all the trees within the constructed forest, using the relation predicted in Section 5.3.2. Finally, we obtain the answer entities from the tree, based on the tree-disambiguation technique following Algorithm 2.

**Building a forest.** In order to build a forest, first we construct a set of knowledge-trees. For each linked entity  $E_i \in \mathcal{E}^L$ , we generate a  $k$ -level tree  $\mathcal{T}_i^k$  constructed from the  $k$ -hop sub-graph associated to  $E_i$  as follows:

$$\mathcal{T}_i^k \leftarrow \text{buildTree}(E_i, \mathcal{G}_i^k) \quad (5.6)$$

The linked entity is designated as the tree's root node (in orange color) at level 0 (Figure 5.4a). In this case, *Lionel Messi* is the root node of a tree. The other nodes and edges in the  $k$ -hop sub-graph of the linked entity are connected to the tree's root node at the same stage as they are in the sub-graph  $\mathcal{G}_i^k$ . The function  $\text{buildTree}(\cdot)$  in Algorithm 2, performs the tree-construction operation. A set of generated  $k$ -level trees are denoted as a forest  $\mathcal{F}$  (as specified by the definition). In cases where no entities are linked, as predicted answer entities the system returns an empty set. For the running example question, the system constructs a forest with one tree for the linked entity *Lionel Messi* (Q615).

Each branch of the tree represents a relation between the parent and the child entity node. For instance in Figure 5.4a, a branch "capital city" connects a parent entity node, "Spain" and a child entity node, "Madrid" ( $\text{Spain} \xrightarrow{\text{capital city}} \text{Madrid}$ ). Each node in a tree preserves a state variable  $\mathcal{V}$ , which holds a set of values  $\{\mathcal{S}_r, \mathcal{K}, \text{and } \mathcal{R}_{max}\}$ . Where  $\mathcal{K}$  denotes the tree level,  $\mathcal{R}_{max}$  the relation for which the node obtained the maximum score, and  $\mathcal{S}_r$  the maximum similarity score for the relation  $\mathcal{R}_{max}$ . During the answer entity extraction process, the values of the state variable aid in the tree-disambiguation process. At this stage, all state variables are initialized with null value.

**Tree-walking.** In this step, the predicted relation  $\mathcal{R}^L$  performs tree-walking across all the trees in the forest, starting from the root node till the nodes at level- $k$  of each tree. During the walk, for each tree  $\mathcal{T}_i^k \in \mathcal{F}$  the system computes embedding-based cosine similarity between the predicted relation  $\mathcal{R}^L$  and all the 1-hop connected branches  $h_i$  of each node  $E_i \in \mathcal{T}_i^k$ . At each step of the walk, the system updates the node state (value of  $\mathcal{S}_r$  and  $\mathcal{R}_{max}$ ) with the similarity scores of the connected 1-hop relations. The values of a node state only get updated when a higher value than the existing  $\mathcal{S}_r$  of that node is obtained for any connected relation (or branch). The function  $\text{updateState}(\cdot)$  in Algorithm 2, updates the node state values with the values passed in as parameters. We employ QuatE [269], a knowledge graph embedding model trained on Wikidata, to compute the similarities between two



**Algorithmus 2 : Answer Entity Extraction**


---

**Input :** A forest  $\mathcal{F}$ , predicted relation  $\mathcal{R}^{pred}$  and hops  $k$   
**Output :** A set of entities  $\mathcal{E}^a$

```

1  $\mathcal{E}^a, \mathcal{S}_r^{max}, \mathcal{R}_{max} \leftarrow \emptyset$ 
2 for  $\mathcal{T}_i^k \in \mathcal{F}$  do
3   for  $E_i \in \mathcal{T}_i^k$  do
4     for  $r_i \in h_i$  do
5        $S_c \leftarrow cosine(emb(\mathcal{R}^{pred}), emb(r_i))$ 
6       if  $S_c > \mathcal{S}_r^{max}$  then
7          $\mathcal{S}_r^{max} \leftarrow S_c ; \mathcal{R}_{max} \leftarrow r_i$ 
8       if  $S_c > E_i[\mathcal{S}_r]$  then
9          $E_i[\mathcal{V}] \leftarrow updateState(S_c, r_i)$ 
10  $h_{low} \leftarrow k$ 
11 for  $\mathcal{T}_i^k \in \mathcal{F}$  do
12   for  $E_i \in \mathcal{T}_i^k$  do
13     if  $E_i[\mathcal{S}_r] = \mathcal{S}_r^{max}$  then
14       if  $E_i[\mathcal{K}] < h_{low}$  then
15          $h_{low} \leftarrow E_i[\mathcal{K}]$ 
16          $\mathcal{E}^a \leftarrow connE(E_i[\mathcal{R}_{max}])$ 
17       else if  $E_i[\mathcal{K}] = h_{low}$  then
18          $E^a \leftarrow connE(E_i[\mathcal{R}_{max}])$ 
19          $\mathcal{E}^a \leftarrow \mathcal{E}^a \cup E^a$ 
20 return  $\mathcal{E}^a$ 

```

---

relations in order to consider KG structural information during the process. In Algorithm 2, the function  $emb(\cdot)$  takes a relation as input and returns the knowledge graph embedding of the relation from QuatE. Finally, the system selects all entities connected to the node with the highest  $\mathcal{S}_r$  value, by  $\mathcal{R}_{max}$ , as answer entities  $\mathcal{E}^a$ .

**Tree-disambiguation.** We introduce a tree disambiguation technique for extracting the answer entities from the forest. In this technique, the system chooses the tree in which the node with the highest score ( $\mathcal{S}_r$ ) resides. If multiple trees have a node with the same maximum score in their node state, the tree with the highest scoring node at the lowest level (lower value of  $k$ ) is chosen (Figure 5.4). Moreover, in rare cases (less than 1% in the WebQSP-WD dataset), when several trees have nodes with the highest scores at the same level ( $k$ ), the system selects all the trees with such cases and extracts all the answer entities connected to the  $\mathcal{R}_{max}$ . In Algorithm 2, line no. 10-19 demonstrate the tree-disambiguation process. Finally, *Barcelona F.C.* is chosen as the answer entity from the tree since the predicted relation *plays for* connects *Barcelona F.C.* to the linked entity *Lionel Messi*. The function  $connE(\cdot)$  in Algorithm 2 selects all the answer entities connected to the entity  $E_i$  by the

relation  $\mathcal{R}_{max}$ .

## 5.4 Experiments and Results

### 5.4.1 Data

We chose Wikidata [16] (based on May 2019 English Wikipedia release) as the knowledge graph to gauge our proposed method since Wikidata is frequently used as a knowledge base for KGQA datasets. We evaluate our proposed method on four publicly available knowledge graph based question answering datasets:

- *LC-QuAD 2.0* [265]: A large-scale dataset on Wikidata Knowledge Graph which was generated semi-automatically and consists of complex questions and their paraphrases.
- *LC-QuAD 2.0 (KBpearl)* [266]: A subset of the LC-Quad 2.0 dataset, selected by [266]. The KBpearl split of the LC-QuAD 2.0 data comprises of 1,942 test questions.
- *QALD-7-Wiki* [267]: A manually constructed small, complex question answering dataset, developed for Task 4 ("English question answering over Wikidata") of the QALD-7 challenge [267].
- *WebQSP-WD* [22]: A Wikidata-based question answering dataset constructed from the original Freebase-based WebQSP dataset [270].

	LC-QuAD 2.0	LC-QuAD 2.0 (KBpearl)	WebQSP-WD	QALD-7-Wiki
<b>Split (train/test)</b>	24,180 / 6,064	24,180 / 1,942	2,880 / 1,033	100 / 50
<b>Number of entities per question</b>	1.47	1.48	1.47	1.08
<b>% of question with no entity</b>	0.02%	0.41%	0.0%	8.0%
<b>Number of words per question</b>	10.61	14.10	6.72	7.62

Table 5.2: Dataset statistics.

It is noteworthy that the system can be extended to different knowledge graphs with low effort (discussed in Section 5.5.4). Table 5.2 lists the statistics of the datasets used in this research.

### 5.4.2 Experimental Setup

We run our experiments on a system with 28 CPU cores, 12GB of GPU memory, and 256GB of RAM. A pre-trained BERT-large [39] model with 340M parameters and BART-large model [40] with 406M parameters are used in this paper. We use macro-F1 score to evaluate the components of our system similar to other baseline models [162, 266].

### 5.4.3 Baselines

We select a wide range of baseline models related to KGQA sub-tasks. The baseline models used in this paper are summarised below:

**DBpedia Spotlight:** An open-source tool and a popular baseline for the entity linking task in TAC-KBP [271, 272].

**TagMe:** An entity linking tool that index Wikipedia pages and performs annotation on a given text [273].

**QKBfly:** An information extraction (IE) tool based on ClausIE [274], which predicts a triple from the KG, on-the-fly [264].

**EARL:** Jointly performs entity and relation linking from the knowledge graph, by solving a *Traveling Salesman Problem* on the candidate nodes [14].

**ReMatch:** A part-of-speech and dependency parsing based relation linking tool for question answering [166].

**Falcon:** A tool that jointly performs entity and relation linking leveraging the concept of morphology and knowledge graph information [149].

**VCG:** A jointly optimized model for entity mention detection and disambiguation using contextual information [160].

**KBPearl-NN:** A neural network based end-to-end system that performs joint entity and relation linking [266].

**PNEL:** A pointer network based entity linking system [162].

**Falcon 2.0:** A morphology based entity and relation linking system [275].

**STAGG:** A semantic parsing approach for question answering over knowledge graph [276]. A re-implementation of STAGG from Sorokin *et al.* [22] to facilitate the KGQA task, is used as a baseline in this work.

**GGNN:** Uses a complex semantic parser for performing question answering over knowledge bases [22]. The baseline scores in this paper are all reported from [162, 266, 22].

### 5.4.4 Results

**Entity Linking.** Table 5.3 shows the entity linking performance of the baseline models and our approach on LC-QuAD 2.0. All the results reported in this section are on the [0,1] scale and test split of the datasets. From the results in Table 5.3, it is evident that our system achieves higher precision, recall and F1 scores as compared to the other baseline models.

Systems	Precision	Recall	F1
OpenTapioca [164]	0.237	0.411	0.301
Falcon 2.0 [275]	0.395	0.268	0.320
VCG [160]	0.403	0.498	0.445
PNEL [162]	0.688	0.516	0.589
<b>Tree-KGQA</b>	<b>0.720</b>	<b>0.566</b>	<b>0.634</b>

Table 5.3: Performance of the entity linking component on LC-QuAD 2.0.

<b>Systems</b>	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
EARL [14]	0.403	0.498	0.445
QKBfly [264]	0.518	0.479	0.498
TagMe [273]	0.352	<b>0.864</b>	0.500
Falcon [149]	0.533	0.598	0.564
KB Pearl-NN [266]	0.561	0.647	0.601
Spotlight [272]	0.585	0.657	0.619
PNEL [162]	<b>0.803</b>	0.517	0.629
<b>Tree-KGQA</b>	0.737	0.666	<b>0.700</b>

Table 5.4: Performance of the entity linking component on the LC-QuAD 2.0 (KBpearl).

<b>Systems</b>	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
TagMe	0.349	0.661	0.457
EARL	0.516	0.460	0.486
QKBfly	0.592	0.510	0.548
Spotlight	0.619	0.634	0.626
Falcon	0.708	0.651	0.678
KB Pearl-NN	0.647	0.715	0.679
<b>Tree-KGQA</b>	<b>0.714</b>	<b>0.648</b>	<b>0.680</b>

Table 5.5: Performance of the entity linking component on the QALD-7-Wiki.

<b>System</b>	<b>Precision</b>	<b>Recall</b>	<b>F1</b>
EARL [14]	0.259	0.251	0.255
ReMatch [166]	0.201	0.214	0.207
Falcon [149]	0.302	0.325	0.313
KB Pearl-NN [266]	0.358	<b>0.479</b>	0.410
<b>Tree-KGQA</b>	<b>0.554</b>	0.400	<b>0.464</b>

Table 5.6: Performance of the relation linking component on the LC-QuAD 2.0 (KBpearl).

We notice a substantial improvement (increment of 7.1%) on LC-QuAD 2.0 KBpearl in entity linking, see Table 5.4. The entity linking result on the small yet challenging dataset (QALD-7-Wiki) is reported in Table 5.5. Improved results across several datasets verify the effectiveness of our unsupervised entity linking approach.

**Relation Linking.** The relation linking performance of the baseline models and our proposed approach on LC-QuAD 2.0 (KBpearl) is reported in Table 5.6. The baseline scores are reported as in Lin *et al.* [266]. Our proposed zero-shot relation label classification approach achieves an increased score of 5.4% over the previous state-of-the-art models.

System	Precision	Recall	F1
STAGG* [276, 277]	0.191	0.227	0.183
Single Edge*	0.224	0.271	0.215
Pooled Edges*	0.209	0.255	0.203
GNN*	0.242	0.289	0.233
GGNN [22]	0.269	<b>0.318</b>	0.259
<b>Tree-KGQA</b>	<b>0.327</b>	0.233	<b>0.273</b>

Table 5.7: Performance of KGQA on WebQSP-WD test set. Models marked with (\*) are the re-implementation from Sorokin *et al.* [22] to meet the KGQA task.

Approach	Precision	Recall	F1
Entity Linking (EL)	0.854	0.810	0.831
Relation Linking (RL)	0.396	0.288	0.334
KGQA <sub>ER</sub>	0.739	0.709	0.724
KGQA <sub>k=1</sub> (with EL and RL)	0.319	0.219	0.260
KGQA <sub>k=2</sub> (with EL and RL)	0.327	0.233	0.273

Table 5.8: Component-wise results of Tree-KGQA.

**KGQA.** We report the KGQA score on WebQSP-WD dataset in Table 5.7. Our introduced Tree-KGQA system achieves an improved result (1.4% rise in F1 score) compared to the previous KGQA baselines. The KGQA scores reported in this paper are computed with  $k = 2$ . Furthermore, we provide a new baseline for the KGQA task on the LC-QuAD 2.0 KBpearl test set in Table 5.9. Moreover, we report the component-wise results of our proposed techniques on WebQSP-WD dataset in Table 5.8. The entries with the approach KGQA<sub>ER</sub> reflect the KGQA score given the ground truth values of EL and RL. We observe an improved KGQA score with  $k = 2$  than  $k = 1$ . Although our system performs remarkably on the EL and answer entity extraction tasks, it has a relatively poor KGQA score due to the low RL score. Nevertheless, relation linking (RL) is a challenging task that is still far from being solved.

System	Precision	Recall	F1
<b>Tree-KGQA</b>	0.526	0.520	0.523

Table 5.9: Our introduced new baseline for the KGQA task on LC-QuAD 2.0 (KBpearl).

## 5.5 Analysis

### 5.5.1 Ablation Study

We conduct an ablation study to investigate the effectiveness of major components of our proposed system. Table 5.10 demonstrates the improvement that each of the components brings to the overall performance of the system. A TF-IDF based entity linking approach exhibits a low F1 score of

0.599, where our proposed indexing mechanism based approach achieves significant gain in the performance (+6.2% using Fasttext and +2.1% using Sentence-BERT embedding). A relation-based entity disambiguation method further improved the result by 1.8%. Our proposed BART-based relation linking approach demonstrates a remarkable improvement (+9.1%) over the cosine similarity based relation linking method.

Furthermore, we assess the performance of the answer extraction component without our proposed tree disambiguation technique. We extract the entities directly connected to the linked entities by the predicted relation as answer entities which achieves a low KGQA F1 score of 0.243. Then, we employ the *tree-walking* and *tree-disambiguation* technique which improves the F1 score by 2.1%. Moreover, we utilized knowledge graph-based embedding during the answer entity extraction procedure to compute the similarity between the predicted relation and the branches of every node in a tree. This method allows the system to surpass Fasttext embedding based similarity calculation by 0.8%.

Task	Approach	F1	$\Delta$
<b>EL</b>	EL (TF-IDF)	0.599	-
	EL (FAISS <sub>KNN</sub> + Fasttext)	0.661	+ 6.2%
	EL (FAISS <sub>KNN</sub> + Sentence-BERT)	0.682	+ 2.1%
	EL (FAISS <sub>KNN</sub> + disambiguation)	0.700	+ 1.8 %
<b>RL</b>	RL (Cosine similarity)	0.373	-
	RL (BART)	0.464	+ 9.1%
<b>KGQA</b>	KGQA (without tree-disambiguation)	0.244	-
	KGQA (with tree-disambiguation + Fasttext)	0.265	+ 2.1%
	KGQA (with tree-disambiguation + KGE)	0.273	+ 0.8%

Table 5.10: Ablation study.

### 5.5.2 Case Study

Table 5.11 shows two cases from the entity linking, relation linking and KGQA tasks. The entity and relation linking cases are from LC-QuAD 2.0, where the KGQA cases are from WebQSP-WD.

**Entity Linking (EL):** Our proposed approach correctly detected and linked the entity in the first case, where Falcon 2.0 and PNEL failed to link the correct entity. This is a challenging case since it contains a long entity span. The underlined texts indicate the entity span in the question. In the second case, all the systems failed to detect *country* as the entity. Although *mahomoud abbas* is correctly detected as entity mention by Falcon 2.0 and PNEL, they linked the entity mention to the wrong KG entity *Mahmoud Abbas (Q10515624)*, who is a footballer. On the contrary, with the help of entity disambiguation where relation information is used, our method correctly linked the mention *mahomoud abbas* to the correct KG entity *Mahmoud Abbas (127998)*, who is the head of a state.

**Relation Linking (RL):** The first case comprises *administrative territorial entity (P150)* and *instance of (P31)* as the ground truth relation. Since *instance of (P31)* does not appear explicitly in the question, it is difficult for the systems to predict it as a relation. In the second case, our proposed

Task	Question	Ground Truth	Falcon 2.0	PNEL	Our approach
EL	What is in work of actor of Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best ?	Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best (Q6675710)	Looney Tunes Super Stars' Pepe Le Pew: (Q6675705), Best (Q4896530)	Looney Tunes Super Stars' Pepe Le Pew: (Q6675705)	Looney Tunes Super Stars' Pepe Le Pew: Zee Best of Zee Best (Q6675710)
	What is the country for head of state of mahmoud abbas?	country (Q6256), Mahmoud Abbas (Q127998)	Mahmoud Abbas (Q10515624)	Mahmoud Abbas (Q10515624)	Mahmoud Abbas (Q127998)

Task	Question	Ground Truth	Falcon 2.0	Our approach
RL	What is the socialist state for contains administrative territorial entity of Beijing?	contains administrative territorial entity (P150), instance of (P31)	contains administrative territorial entity (P131)	contains administrative territorial entity (P150)
	What kind of disease does montel williams have?	medical condition (P1050)	-	medical condition (P1050)

Task	Question	Ground Truth	GGNN	Our approach
KGQA	Where is jamarus russell from?	Mobile (Q79875)	Mobile (Q79875)	Mobile (Q79875)
	Who did tim tebow play college football for?	Florida Gators football (Q5461394)	Florida Gators football (Q5461394)	Florida Gators football (Q5461394), Denver Broncos (Q223507), New York Jets (Q219602), Philadelphia Eagles (Q219714)

Table 5.11: Case study.

Algorithm 1 correctly predicted the relation *medical condition* (P1050). We adopt a BART-large model [40] in Algorithm 1, pre-trained on natural language inference task, which gives better inference capabilities in identifying the correct relation from a set of candidate relations.

**KGQA:** Our proposed unsupervised KGQA approach correctly extracted the answer entity in the first case. In the second case, *Florida Gators football* (Q5461394) is given as the ground truth which can be inferred by the relation *member of sports team* (P54) connected to the entity *Tim Tebow* (Q517467). However, our system extracted all the entities as the answer entities that are connected to *Tim Tebow* (Q517467) by the relation *member of sports team* (P54).

### 5.5.3 Error Analysis and Limitations

We conducted an error analysis to understand the cases where our system is not performing as expected. We observed that our proposed entity linker is unable to detect entities that are not named entities such as *president* (Q30461) and *governor* (Q132050), since it is using NER for detecting the entity mention(s). Here, Q30461 and Q132050 are Wikidata ID of the respective entities.

The most challenging aspect of KGQA is relation identification. Relations with similar labels exist in the Wikidata KG, which are difficult for systems to differentiate. For instance, the relations *head*

of government (*P6*) and head of state (*P35*). This issue becomes more visible when we found that, F1 score on top-3 predicted relation is 49.39 and in top-10 it is 57.66. The relation accuracy results reported in Table 5.6 are based on the top-1 predicted results from the proposed zero-shot relation linker. Our system fails to predict relations requiring more complex reasoning capabilities, such as hierarchical relationships. For instance, for the question "Give me cinematic technique that contains the word tilt in their name", the correct relation that can be used to answer the question is *Instance of* (*P31*), which our system failed to capture. Furthermore, our proposed zero-shot relation linker can only predict one relation. Although this is a limitation of the system, questions generally contain one relation in the context of question answering.

Although our proposed answer extraction method is fairly straightforward, we observe that the KGQA model mainly suffers in the cases where no entities are predicted and the cases where a wrong relation is predicted. Similar to the relation linking, our system also fails to extract the correct answer entities for cases where comparative or logical reasoning is required to answer the questions (E.g., *Is Lake Baikal bigger than the Great Bear Lake?*).

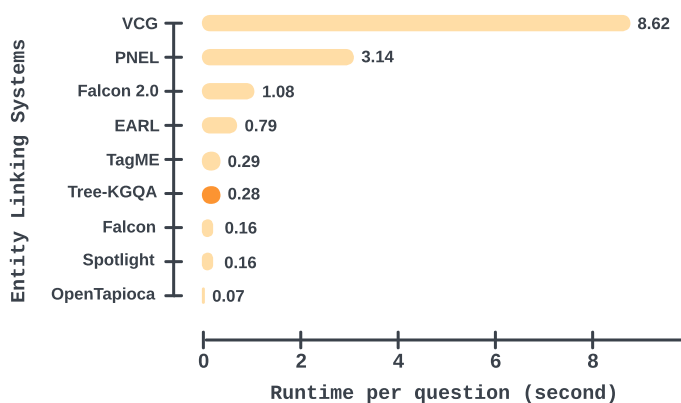


Figure 5.5: Inference time efficiency of the entity linking systems.

#### 5.5.4 Discussion

The improved entity linking performance of our proposed model across all the benchmark datasets provides a solid foundation for the KGQA task. Despite the fact that our proposed relation linking approach outperforming previous methods in complex QA, it could benefit further from better logical inference capabilities. Furthermore, we designed our system in a modular way so that it can be easily extended and used across different KGQA sub-tasks. Within the scope of this work, we explored Wikidata-based datasets. However, from the description of our approaches, we can intuitively say that our system can be adapted for other knowledge graph based datasets. For that, first, the pre-processing step where entity indexing is performed needs to be executed. Then, we need to obtain the relation embedding from a knowledge graph embedding model to perform tree-walking (Section 5.3.3).

Our proposed KGQA system is runtime efficient. Several factors contributed to the fast runtime of our system. In entity linking, the FAISS indexing technique provides fast candidate generation (takes  $\sim 0.04$  seconds to generate 10 candidates per question). The performance of the entity linking baselines is shown in Figure 5.5 (baseline runtimes are reported from Banerjee *et al.* [162]). Furthermore, the



relation linking component requires  $\sim 0.09$  seconds per question. Moreover, our proposed tree-based answer extraction process takes  $\sim 0.39$  seconds per question. Overall, the system takes  $\sim 0.76$  seconds per question to perform the entire KGQA task.

## 5.6 Summary

We presented Tree-KGQA, an unsupervised technique to perform KGQA without any explicit training. Despite the simplicity, our proposed pre-trained language model-based, unsupervised method outperforms existing supervised systems by a fair margin in all the sub-tasks involved in KGQA. The superior performance of Tree-KGQA answers RQ2 (*How effective are pre-trained language models for developing an unsupervised knowledge-graph-based question-answering system without training data?*) by only leveraging pre-trained language and knowledge graph embedding models. To substantiate our claim, we evaluate our proposed system across several benchmark datasets. Although our system proves to be useful for the majority of the types of questions found in the datasets studied, further work is required to tackle more challenging questions requiring counting, comparisons, and logical reasoning capabilities.




## SPARQL Query Generation: A Generative Approach

Traditional pipeline based question answering systems employ formal queries such as SPARQL to extract answer entities from structured knowledge sources (i.e., knowledge graph). SPARQL is a query language used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. In recent years, the conversion of natural language questions (NLQs) to SPARQL queries gained further popularity to the growing number of graph-based applications [278, 279, 280]. Automatic query generation from NLQ is a long-standing research

**Question:** What are the moons of Pluto?

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT DISTINCT ?obj WHERE
{
  wd:Q339 wdt:P398 ?obj . ?obj wdt:P31 wd:Q184246
}
```



```
PREFIX dbc: <http://dbpedia.org/resource/Category/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?obj WHERE
{
  ?obj rdf:type dbo:Planet . ?obj dct:subject dbc:Moons_of_Pluto
}
```




Figure 6.1: An illustration of a SPARQL query used to answer a natural question over Wikidata [16] and DBpedia [195]. Here, Q339, P398, P31, Q184246 are the Wikidata ID of *Pluto*, *child astronomical body*, *instance of*, and *moon of Pluto*, respectively.

challenge with several factors contributing to its difficulty, including but not limited to understanding the complex aspects of syntax and semantics of the natural language question (i.e., ellipsis, ambiguity, lexical gap), error propagation in NLP pipelines, and skewed distribution of question types in training datasets. Additionally, changing the underlying KG requires rewriting the SPARQL query for a given NLQ as illustrated in Figure 6.1.

Chapter 5 discussed about an unsupervised way to extract answer from a knowledge graph, given that the entity and relations are already linked. This chapter describes a generative approach to directly generate SPARQL queries from natural question without entity or relation being already linked. However, the proposed approach also functions if entity information is provided. Specifically, this chapter proposes techniques to embed knowledge graph into language models parameters to generate SPARQL queries from natural questions. This chapter answers the research question RQ3, "**Can a generative language model embed a knowledge graph in its parameters and learn to construct SPARQL queries?**". The discussions of this chapter is based on the following paper:

- **Md Rashad Al Hasan Rony**, Uttam Kumar, Roman Teucher, Liubov Kovriguina and Jens Lehmann, *SGPT: A Generative Approach for SPARQL Query Generation from Natural Language Questions*, in *IEEE Access*, vol. 10, pp. 70712-70723, 2022, doi: 10.1109/ACCESS.2022.3188714.

This chapter is organized into five sections. Section 6.1 discusses the motivation and drawbacks of existing template-based SPARQL query generation methods. A brief overview of the proposed system and its contributions are additionally discussed. The problem statement and the proposed approach is explained in detail in Section 6.2. The proposed metric and experiments are explained in Section 6.3. An in-depth analysis of the proposed system is provided in Section 6.4. Finally, Section 6.5 summarizes the chapter with concluding remarks.

## 6.1 Introduction

Several approaches for SPARQL query generation have been presented recently [197, 198, 200, 97, 98]. The widely adopted approaches involve query schema or template classification and filling in the slots in the templates using available sub-graph information such as linked entities and relations [281, 97, 282, 98]. A different line of research is centered around transforming natural language questions to their corresponding SPARQL queries in a sequence-to-sequence manner [200, 107].

In this chapter, we propose a new approach, dubbed SGPT, for SPARQL query generation. SGPT encodes the linguistic features of an NLQ and corresponding sub-graph information (i.e, entities, if provided), and leverages a generative language model (LM) to generate SPARQL queries. We hypothesize that a deeper understanding of the NLQ is crucial for generating a correct query, since a slight deviation in the syntactic structure of the question may result in a different SPARQL query.

Question 1:	What is the name of the actress married to the prince of England?
SPARQL 1:	<pre>SELECT ?s_label WHERE {   ?s wdt:P106 wd:Q33999 . ?s wdt:P26 ?spouse .   ?spouse wdt:P97 wd:Q4971429 . ?s rdfs:label ?s_label   FILTER ( lang ( ?s_label ) = "en" ) }</pre>
Question 2:	What is the name of the prince of England married to an actress?
SPARQL 2:	<pre>SELECT ?s_label WHERE {   ?spouse wdt:P106 wd:Q33999 . ?s wdt:P97 wd:Q4971429 .   ?s wdt:P26 ?spouse . ?s rdfs:label ?s_label   FILTER ( lang ( ?s_label ) = "en" ) }</pre>

Table 6.1: Comparison of SPARQL queries for two different questions with same wording.

Table 6.1 demonstrates such an example, where the queries are Wikidata knowledge graph-based [16], and *Q33999*, and *Q4971429* are Wikidata entity IDs of the entity *Actor* and *British Prince*, respectively.

The Wikidata relation IDs *P106*, *P26* and *P97* refer to the relations *Occupation*, *Spouse*, and *Noble title*, respectively.

Besides the standard word and positional embedding layers, we design special embedding layers that embed an arbitrary number of linguistic features of an NLQ, such as parts-of-speech (POS) tags and dependency tree features (i.e., dependency relations and information about tree node’s children). The layers proposed in this chapter are different from the ones in Chapter 4. The layers designed in Chapter 4 focuses on distinguishing different word token in the input sequence by defining the type and triple embeddings, whereas this chapter focuses on understanding the content of the question better. A stack of Transformer [37]-encoders is employed to encode the linguistic features. The proposed embedding techniques facilitate SGPT to inject additional knowledge (i.e., entities) as well as allow the integration of SGPT into pipeline-based systems in a modular fashion. Furthermore, we employ the Transformer [37]-decoder based language model GPT-2 [38], to generate SPARQL queries. Our training methodology enables SGPT to embed an arbitrary KG directly into the model parameters. Moreover, the system does not require any query template or KG as input at inference time.

The evaluation of SPARQL query generation is a crucial step for developing NLQ to SPARQL systems. A widely used metric BLEU [32], was primarily designed to evaluate machine translation (MT) and later adopted for evaluating natural language generation (NLG). However, in contrast to natural language sequences, SPARQL is a formal language and includes query-specific terms, patterns and variables which the standard automatic metrics such as BLEU do not consider when computing  $n$ -gram overlaps. To overcome this shortcoming, we propose a variable normalization algorithm to adopt BLEU and F1 score for measuring the performance of SPARQL query generation. We call the adopted metrics SP-BLEU and SP-F1.

To assess the performance of SGPT, we conduct experiments on three publicly available datasets: LC-QuAD 2.0 [283], VQuAnDA [284] and QALD-9 [285]. We evaluate the system-generated SPARQL queries using both human and automatic metrics. Furthermore, by an ablation study we examine the impact of individual components on SGPT’s overall performance to verify their effectiveness. Moreover, we conduct extensive analysis to demonstrate SGPT’s capacity to comprehend diverse, complex questions and generate correct SPARQL queries. The empirical evaluation confirms that SGPT significantly outperforms state-of-the-art methods in generating SPARQL queries from natural language questions across several benchmark datasets.

### Contributions

- A novel embedding technique, that embed the linguistic features of a question and graph information for the SPARQL query generation task.
- A generative system, SGPT, that utilizes the linguistic features of a natural language question and learns to embed the KG into language model’s parameters. SGPT can be used as either as a standalone system or can be integrated into modular pipelines.
- An algorithm to adapt standard evaluation metrics for measuring the performance of SPARQL query generation.

## 6.2 Approach: SGPT

### 6.2.1 Problem Definition

This chapter proposes separate training techniques for two use cases, 1) only a natural question is available, 2) both the question and entities mentioned in the question are provided. In the second case, we consider the provided set of entities as additional knowledge  $\mathcal{K}$ . Given a natural language question  $Q$  (for the first case) or an additional knowledge  $\mathcal{K}$  and a natural question  $Q$  (for the second case), the goal of SGPT is to generate a SPARQL query  $S$ . We define  $\text{SGPT}_Q$  as the system for the first use case and  $\text{SGPT}_{Q,\mathcal{K}}$  for the second use case. Formally, in  $\text{SGPT}_Q$ , the probability distribution of generating a SPARQL query by the language model is defined as:

$$p_\theta(S|Q) = \prod_{i=1}^n p_\theta(s_i | s_1, \dots, s_{i-1}, Q), \quad (6.1)$$

and in  $\text{SGPT}_{Q,\mathcal{K}}$  the probability distribution is as follows:

$$p_\theta(S|Q, \mathcal{K}) = \prod_{i=1}^n p_\theta(s_i | s_1, \dots, s_{i-1}, Q, \mathcal{K}), \quad (6.2)$$

where  $\theta$  is model's parameters,  $n$  is the query length and  $s_i$  is the token generated at  $i$ -th time step. We use the terms "SPARQL query" and "query" interchangeably throughout this chapter. The term "SGPT" refers to both  $\text{SGPT}_Q$  and  $\text{SGPT}_{Q,\mathcal{K}}$ , if there is no design or implementation difference between them for the describe concept or operation. SGPT follows the encoder-decoder design paradigm. The approach is described in depth in the following subsections.

### 6.2.2 Encoding

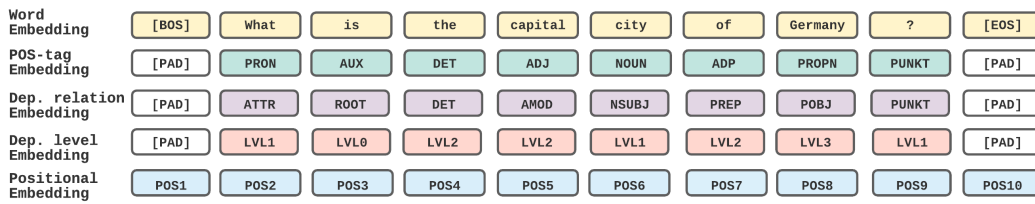
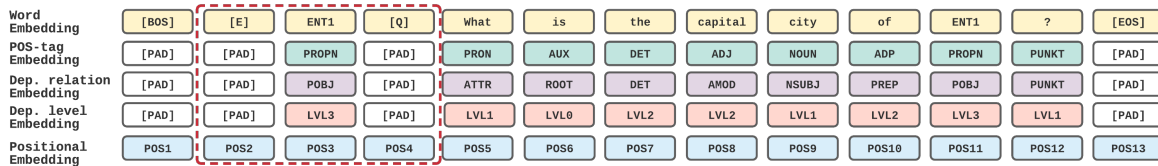
We design special embedding layers to embed the linguistic features of the question. The idea of special embedding was initially suggested by Devlin et al. [45]. The idea has been recently adopted for encoding structural information such as in table parsing [286] and graph-based dialogue generation task [287]. Unlike these prior works, in this work special embedding layers are designed to capture the linguistic characteristics of an NLQ. A stack of Transformer [37]-encoders then encodes these embeddings. Below we describe the process in detail.

#### Input Sequence Construction.

A *Pre-processor* component in SGPT takes  $Q$  and  $\mathcal{K}$  as input and constructs the input sequence. The input sequence in  $\text{SGPT}_Q$  starts with a [BOS] token, then the question  $Q$  and an [EOS] token that marks the end of the sequence as depicted in Figure 6.2.

Similarly, in  $\text{SGPT}_{Q,\mathcal{K}}$  the input sequence starts and ends with [BOS] and [EOS] tokens, respectively. The question is separated by a [Q] token from the additional knowledge  $\mathcal{K}$ , in the input sequence. Furthermore, each entity in the additional knowledge is preceded by an [E] token in the input sequence (see Figure 6.3).

To allow generalisation, the entity positions in the question and query are masked in  $\text{SGPT}_{Q,\mathcal{K}}$ . A *Pre-processor* masks both the entities in the question and the entities (including their prefix) in the

Figure 6.2: An illustration of special embedding layers used in  $SGPT_Q$ .Figure 6.3: The question and knowledge embedding techniques used in  $SGPT_{Q,K}$ . The dotted red box indicates the separation of additional knowledge from the question.

query by a generic ENT token for training. The entities in the query that do not appear in the question are not masked and learned in the model’s parameters. For multiple entities in the question the generic masks are as follows: ENT1, ENT2, ..., ENT $n$ , where  $n$  is the number of entities present in the question. For instance, if the question contains two entities, the first is masked by the ENT1 token and the second by the ENT2 token. Figure 6.3 depicts a masked input sequence.

Generally, relation linking is challenging because, the surface form of the relation in the question often differs from the label of the relation in the KG [14]. This leads to relation linking-based error propagation in the pipeline-based systems [288, 289]. To alleviate the error propagation, we delegated the relation learning task to the GPT-2 model, which learns the KG (i.e., entity and relation) in its parameters.

### Embedding the Input Sequence.

The constructed input sequence is passed through five different embedding layers to capture different properties of the input. The embedding layers are described below:

(i) **Word embedding** layer encodes the token level information of the input sequence. A pre-trained GPT-2 [38] tokenizer is used to tokenize the input sequence.

(ii) **POS-tag embedding** layer embeds the part-of-speech tag of the corresponding token in the word embedding layer. POS-tags are used to understand the use of word in the question better, since a particular word may have different meaning based on the usage in a sentence.

(iii) **Dependency relation embedding** layer encodes the dependency relations between pairs of words in the question.

(iv) **Dependency level embedding** layer embeds the information about the children of the tokens in the word embedding layer, extracted from the dependency tree.

(v) **Positional embedding** layer embeds the absolute position information of the input sequence.

SGPT computes the sum of POS-tag, dependency relation and dependency level embeddings and apply *Layer Normalization* [244] to obtain the linguistic context of the NLQ. *Layer Normalization* normalizes the embedding and prevents the model’s weights from exploding. The encoding of

linguistic context is discussed in the next section. The word and positional embeddings are utilized by a GPT-2 decoder, discussed in §6.2.3.

### Linguistic Context Encoding

To generate a correct SPARQL query, it is crucial for the system to understand the linguistic features of a question that captures various question types and patterns. In contrast to [21], where the context vector is used to learn the mapping between NLQ and its corresponding SPARQL, SGPT learns various linguistic features (i.e., POS-tag and word dependencies) separately leveraging the layers introduced. A stack of Transformer-encoders [37] is employed in this chapter to encode the linguistic context. The output of the  $l$ -th encoder layer is formalized as follows:

$$\begin{aligned}
 h_i^l &= \sum_{j=1}^N \alpha_{ij}^l (h_j^{l-1} W^V) \\
 \alpha_{ij}^l &= \frac{\exp(t_{ij}^l)}{\sum_{p=1}^N \exp(t_{ip}^l)} \\
 t_{ij}^l &= \frac{(h_i^{l-1} W^Q)(h_j^{l-1} W^K)}{\sqrt{d}} \\
 i &= 1, 2, \dots, N
 \end{aligned} \tag{6.3}$$

where  $W^Q$ ,  $W^K$ , and  $W^V$  are trainable weights,  $N$  is the sequence length, and  $d$  is the dimension of query, key and value vectors. The output is then passed to a Feed-Forward Neural Network (FFNN), preceded and followed by residual connections and normalization layers as follows:

$$\begin{aligned}
 h_i^{\prime l} &= \text{LayerNorm}(h_i^l + h_i^{l-1}) \\
 h_i^{\prime\prime l} &= W_2^l \text{ReLU}(W_1^{l+1} h_i^{\prime l} + b_1) + b_2 \\
 \hat{h}_i^l &= \text{LayerNorm}(h_i^{\prime l} + h_i^{\prime\prime l}),
 \end{aligned} \tag{6.4}$$

where  $W_2^l$  and  $W_1^{l+1}$  are trainable weights and  $b_1$  and  $b_2$  are bias terms. A rectified linear unit (ReLU) [290] is employed as the activation function in the FFNN network. The output of the last encoder layer  $\hat{h}_i^l$  is then passed to a GPT-2 model for decoding. Figure 6.4 illustrates a high-level architecture of SGPT.

### 6.2.3 Decoding

A GPT-2 [38] language model is used in this chapter to model SPARQL query generation. However, any Transformer [37] decoder-based LM can be used. GTP-2 is a multi-headed attention-based language model. The attention, computed in each of GPT-2's heads is formalized as follows:

$$\begin{aligned}
 \mathcal{F}(Q, K, V) &= \text{softmax}\left(\frac{1}{\sqrt{d_k}}(QK^T) + M\right)V, \\
 H_i &= \mathcal{F}(QW_i^Q, KW_i^K, VW_i^V),
 \end{aligned} \tag{6.5}$$



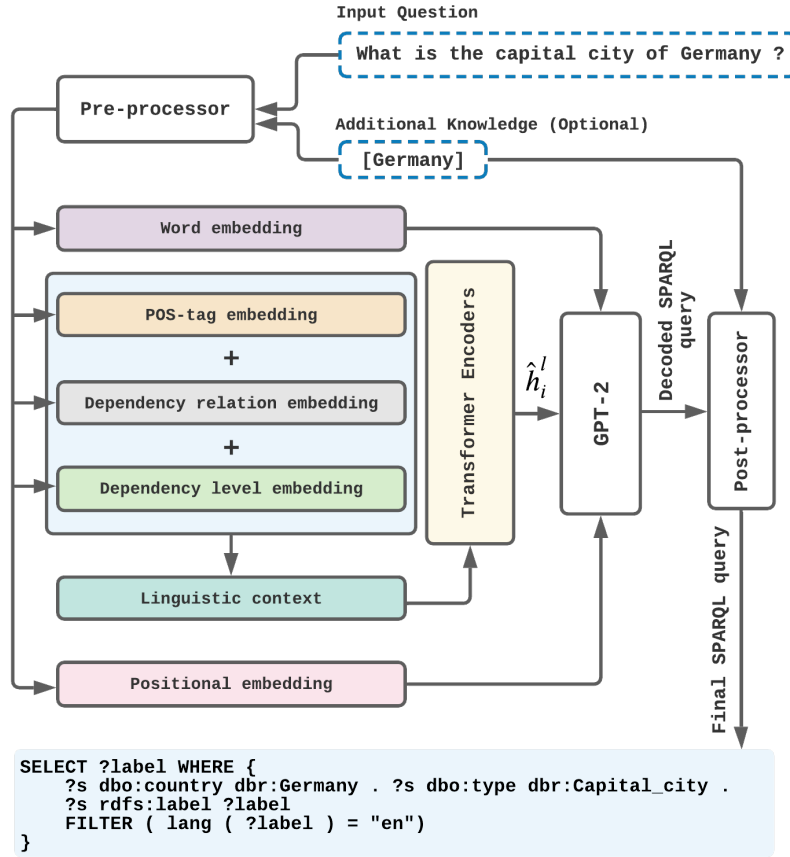


Figure 6.4: System Architecture.

where  $\mathcal{F}(\cdot)$  computes the masked attention. The attention mask is denoted as  $M$ , where  $H_i$  is the  $i$ -th head and  $d_k = d_m/h$ . Here,  $d_m$  is the model's dimension and  $h$  denotes the number of heads.  $Q$ ,  $K$  and  $V$  are query, key and value where  $W_i^Q$ ,  $W_i^K$ ,  $W_i^V$  are trainable weights. Model parameters in  $\theta$  are trained to minimize the negative log-likelihood  $\mathcal{L}_Q$  (for  $\text{SGPT}_Q$ ) and  $\mathcal{L}_{Q,\mathcal{K}}$  (for  $\text{SGPT}_{Q,\mathcal{K}}$ ) for next-token prediction. Formally, the loss  $\mathcal{L}_Q$  and  $\mathcal{L}_{Q,\mathcal{K}}$  are defined as follows:

$$\mathcal{L}_Q = - \sum_i^n \log p(s_i | s_1, \dots, s_{i-1}, Q),$$

$$\mathcal{L}_{Q,\mathcal{K}} = - \sum_i^n \log p(s_i | s_1, \dots, s_{i-1}, Q, \mathcal{K}),$$
(6.6)

where  $n$  is the maximum query length. During inference, *Top-k sampling* decoding [248] is utilized to generate a word token at each time step since Beam Search [291, 292]. is computationally expensive for generating larger sequence It is noteworthy that the entities that were masked by the pre-processor of  $\text{SGPT}_{Q,\mathcal{K}}$  in the question also appear masked in the decoded query. Once the sequence decoding is completed, the *Post-processor* component replaces the entity masks with their corresponding entity identifier.

## 6.3 Experiments and Results

### 6.3.1 Data

We evaluate SGPT on three publicly available datasets.

1) **LC-QuAD 2.0** [283]: A large-scale question answering dataset, which includes for each complex natural language question its corresponding query template, SPARQL query and annotations. We chose LC-QuAD 2.0 to evaluate Wikidata-based questions.

2) **VQuAnDa** [284]: A verbalization dataset which contains natural language questions and their corresponding SPARQL queries for extracting answers. VQuAnDa contains DBpedia-based questions.

3) **QALD-9** [285]: QALD-9 is a small yet challenging multilingual question answering dataset based on DBpedia. The dataset contains questions in 3 to 8 different languages. Within the scope of this chapter, we select the English data.

	LC-QuAD 2.0	VQuAnDa	QALD-9
# train	21,497	3,500	350
# validation	2,389	500	58
# test	5,969	1,000	150
Avg. # tokens in the question	10.55	11.09	7.48
Avg. # tokens in the query	13.68	12.42	13.20
Avg. # keywords in the query	2.08	1.96	2.21

Table 6.2: Dataset statistics.

The dataset statistics are summarized in Table 6.2. The original train and test splits of LC-QuAD 2.0, VQuAnDa and QALD-9 are 24,180/6,046, 4,000/1,000 and 408/150, respectively. For the validation during the training, we split the training set and use 10%-15% data as the validation set, based on the dataset size. In LC-QuAD, 2.0 we removed the data from train set with empty question and query field. Natural language questions are treated as complex when answering them requires multiple graph patterns. Depending on the complexity, the following question types are distinguished [283]:

- *Boolean*: Question where the answer is either True or False.
- *Count*: That computes the number of occurrence of a particular thing.
- *Rank*: Questions seek answer which is in a particular order.
- *Simple*: Questions correspond to semantics of natural language question that is obtained by matching just one hop relations of the entity.
- *String*: Questions, for which answers contain a particular word or letter.
- *Two Hop*: Questions, in which semantic interpretation corresponds to two hop of the entity's connection in the knowledge graph i.e. two set of triples in the where clause of the SPARQL query.
- *Two Intent*: Questions seek for minimum two answer for the same question for example, mother of a person and also the child of same person.

The question types statistics of the benchmark datasets are reported in Table 6.3.

Question Type	LC-QuAD 2.0		VQuAnDa		QALD-9	
	Train	Test	Train	Test	Train	Test
Boolean	2,111	1,433	328	82	40	5
Count	1,134	281	676	181	58	32
Rank	905	204	134	29	59	13
Simple	3,216	824	617	172	158	52
String	5,920	1,433	1	-	32	17
Two hops	20,283	5,049	3,052	744	203	88
Two intents	5,062	1,223	-	-	17	11

Table 6.3: Statistics of question types.

Models	with $\mathcal{K}$	LC-QuAD 2.0				VQuAnDa				QALD-9			
		BLEU	F1	SP-BLEU	SP-F1	BLEU	F1	SP-BLEU	SP-F1	BLEU	F1	SP-BLEU	SP-F1
N <sub>SpM</sub> [200]	✗	34.74	66.47	38.39	70.78	37.75	59.96	37.75	59.96	18.23	45.34	24.18	50.53
SGPT <sub>Q</sub> (ours)	✗	<b>60.50</b>	<b>83.45</b>	<b>63.59</b>	<b>86.22</b>	<b>63.82</b>	<b>87.08</b>	<b>63.82</b>	<b>87.08</b>	<b>29.95</b>	<b>60.22</b>	<b>32.12</b>	<b>64.57</b>
SQG [97]	✓	-	-	-	-	5.09	37.70	33.86	44.67	4.44	27.85	22.14	39.39
TeBaQA [98]	✓	-	-	-	-	13.30	22.41	13.30	22.41	12.82	28.81	17.48	32.24
*BART [21]	✓	-	64.00	-	-	-	-	-	-	-	-	-	-
*T5 (small) [21]	✓	-	<b>92.00</b>	-	-	-	-	-	-	-	-	-	-
*PGNs [21]	✓	-	86.00	-	-	-	-	-	-	-	-	-	-
SGPT <sub>Q, <math>\mathcal{K}</math></sub> (ours)	✓	<b>73.78</b>	89.04	<b>77.85</b>	<b>92.27</b>	<b>72.58</b>	<b>88.87</b>	<b>72.58</b>	<b>88.87</b>	<b>35.68</b>	<b>67.82</b>	<b>41.88</b>	<b>72.98</b>

Table 6.4: Performance of SGPT and baseline models on three benchmark datasets. Best scores are in **bold**. F1 scores computed for the models with \* are against the entity and relation set and do not consider all the tokens in the SPARQL query.

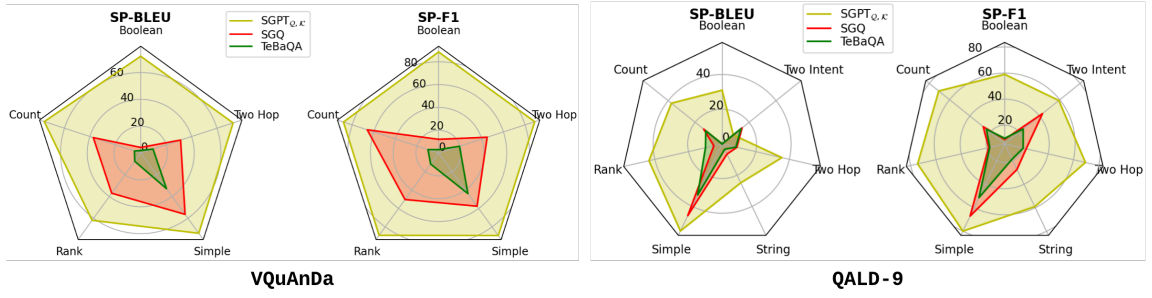
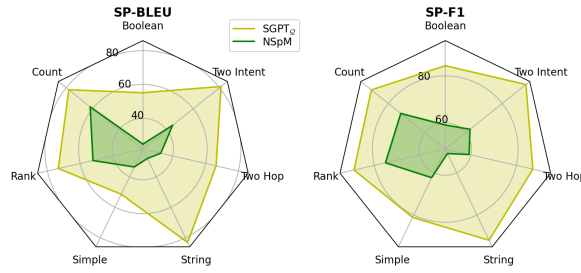
	VQuAnDa		QALD-9	
	SP-BLEU	SP-F1	SP-BLEU	SP-F1
SQG	31.11	46.40	24.30	55.60
SGPT <sub>Q, <math>\mathcal{K}</math></sub>	55.98	80.45	22.21	57.47
TeBaQA	30.55	44.81	25.57	55.14
SGPT <sub>Q, <math>\mathcal{K}</math></sub>	62.92	83.65	33.10	71.89

Table 6.5: Results on data where baseline models could generate queries.

### 6.3.2 Training Settings

We use a stack of Transformer-encoders with 8 heads and 6 layers to encode linguistic features. For decoding we employ the GPT-2 [38] model with 117M parameters throughout this chapter. As an optimizer, AdamW [249] with  $\epsilon = 1e-8$  and a value of  $6.25e-5$  is used as learning rate. GELU [250] is used as the activation function. The optimum hyper-parameters for each dataset were determined using grid search based on the performance on the validation set. We used spaCy<sup>1</sup> to annotate the NLQ with the POS-tags and dependency relations, based on the work of [293]. All experiments were run in a distributed training environment with 2 GPUs, each with 12 GBs of RAM. The training takes 215, 125 and 35 minutes on LC-QuAD 2.0, VQuAnDa, and QALD-9, respectively.

<sup>1</sup><https://spacy.io/>


 Figure 6.5: Question type-wise performance of  $SGPT_{Q,K}$  and baseline models on the test set of VQuAnDa and QALD-9.

 Figure 6.6: Question type-wise performance of  $SGPT_Q$  on LC-QuAD 2.0 test set.

---

**Algorithmus 3 : Query Normalization**


---

**Input :** A SPARQL query  $\mathcal{S}$ 
**Output :** A normalized SPARQL query  $\mathcal{S}_n$ 

- 1  $V_o \leftarrow \emptyset, V_n \leftarrow \emptyset$  initializing empty sets
  - 2  $i \leftarrow 0, i_o \leftarrow 0, \mathcal{S}_n \leftarrow \emptyset$
  - 3 **for**  $w \in \mathcal{S}$  **do**
  - 4     **if**  $w$  *startswith* '?' **then**
  - 5          $t_c \leftarrow \emptyset$
  - 6         **if**  $w \in V_o$  **then**
  - 7              $i_o \leftarrow \text{indexOf}(w, V_o)$  position of  $w$  in  $V_o$
  - 8              $t_c \leftarrow V_o[i_o]$
  - 9         **else**
  - 10              $V_o \leftarrow \text{add}(w, V_o)$  adds a value  $w$  to the set  $V_o$
  - 11              $i \leftarrow i + 1$
  - 12              $t_c \leftarrow \text{string}(?var\ i)$  converts to string
  - 13          $\mathcal{S}_n \leftarrow \text{concat}(\mathcal{S}_n, t_c)$  string concatenation
  - 14     **else**
  - 15          $\mathcal{S}_n \leftarrow \text{concat}(\mathcal{S}_n, t_c)$  string concatenation
  - 16 **return**  $\mathcal{S}_n$
-

### 6.3.3 Evaluation Metrics

**Automatic Metrics.** Following the baseline models, we use BLEU [32] and F1 score as automatic metrics for the evaluation. Generally, SPARQL queries in the used data sets were created manually or semi-automatically by domain experts. This means that the choice of variable names depends on the domain experts and can vary. Hence, we argue that these metrics are incapable of capturing the variations in the variables used in a the query, since BLEU computes  $n$ -gram overlaps and F1 is computed by token-level precision and recall. We propose an adaption, where variables in both reference and predicted queries are normalized before the standard evaluation performed by BLEU and F1 and named them SP-BLEU and SP-F1, respectively. The proposed normalization technique is shown in Algorithm 3. Our proposed variable name normalization technique allows the automatic metric to evaluate a predicted query regardless of the annotated variable names. An example of query normalization is demonstrated in Table 6.6. The normalisation results in the metrics more closely reflecting actual performance. We compare all systems on both the standard automated metrics and the proposed metrics (discussed in §6.3.5).

**Human Evaluation.** We further conducted a human evaluation to manually assess the quality of generated queries. We randomly chose 75 examples (25 from each dataset) and asked two domain experts to evaluate the system generated queries based on the following criteria: 1) *Syntax validity* - how structurally correct the generate queries are, and 2) *Content validity* - how correct the entities and relations are. We asked the reviewers to rate the system generated queries on a scale of 1 to 5 (higher is better). The inter-annotator agreement score (Cohen’s kappa  $\kappa$ ) of the annotated data is 0.86.

<b>Question</b>	How many grand-children did Jacques Cousteau have ?
<b>Reference</b>	SELECT COUNT ( DISTINCT ?y as ?y ) WHERE { dbr:Jacques_Cousteau dbo:child ?x . ?x dbo:child ?y . }
<b>Normalized Reference</b>	SELECT COUNT ( DISTINCT ?var1 as ?var1 ) WHERE { dbr:Jacques_Cousteau dbo:child ?var2 . ?var2 dbo:child ?var1 . }
<b>Prediction</b>	SELECT COUNT ( DISTINCT ?string as ?string ) WHERE { dbr:Jacques_Cousteau dbo:child ?uri . ?uri dbo:child ?string . }
<b>Normalized Prediction</b>	SELECT COUNT ( DISTINCT ?var1 as ?var1 ) WHERE { dbr:Jacques_Cousteau dbo:child ?var2 . ?var2 dbo:child ?var1 . }

Table 6.6: An illustration of query normalization.

### 6.3.4 Baselines

We compare SGPT with both sequence-to-sequence and template-based methods. We train and evaluate the baseline models with their recommended settings. Below we provide a brief description of baseline models:

- **SQG** [97]: A set of candidate queries are created in SQG based on the sub-graph patterns, which are then ranked and arranged based on structural similarity, utilizing Tree-LSTM [202].
- **NSpM** [200]: A sequence-to-sequence strategy in which a Bidirectional Long Short-Term

System	with $\mathcal{K}$	Syntax validity	Content validity
NSpM [200]	✗	4.17	3.00
SGPT <sub>Q</sub> (ours)	✗	<b>4.96</b>	<b>4.10</b>
SQG [97]	✓	3.42	2.73
TeBaQA [98]	✓	3.85	2.99
SGPT <sub><math>\mathcal{K}, Q</math></sub> (ours)	✓	<b>5.00</b>	<b>4.26</b>

Table 6.7: Human evaluation results.

Memory Network (Bi-LSTM) learns to generate a template SPARQL from a natural language question.

- **TeBaQA** [98]: The TeBaQA model depends on template classes which it generates from the training dataset, to predict the SPARQL query. To generate the SPARQL query, the model first classifies the input question and predicts a template class. The slots in the template class are filled in by indexed entities and relations guided by a rule-based lookup. The generated queries are then ranked to obtain the final SPARQL query representation.
- **PLMs** [21]: A sequence-to-sequence approach that investigates the performance of various pre-trained language models (T5 [108], BART [40] and pointer networks [161]) for generating SPARQL queries from NLQ provided that the entity and relations are already linked.

### 6.3.5 Quantitative Results

Table 6.4 summarises the performance of SGPT and baseline systems. In the first set of results where additional knowledge  $\mathcal{K}$  is not provided, SGPT<sub>Q</sub> outperformed the other generative system, NSpM, significantly across all metrics. In many cases NSpM failed to recognize the correct question types, thus frequently generated wrong queries. In the second set of results, using additional knowledge, the baseline models obtained very low scores because of their limited template coverage. We further investigated the performance of baseline models and observed that SQG managed to generate queries for 46% and 45.33% of the test NLQs of VQuAnDa and QALD-9, respectively. TeBaQA could generate queries for 30.55% and 40.67% of the same test NLQs. These template-based systems fail to generate queries primarily for two reasons: 1) They could not classify or find a suitable template for a give question 2) They failed to fill in all the slots in the selected template, resulting in no query predicted. We report the comparison of performance on the data where baseline models could generate queries in Table 6.5. The results suggest that SGPT outperformed the baseline models significantly in most cases. Despite given the correct subjects detected from the question, template-based systems failed to generate queries frequently. The main reason is that SPARQL queries oftentimes include intermediary entities which leads to correct answer but do not appear in the question. Our proposed training technique allows SGPT to learn those entities in the model’s parameters and thus can effectively generate correct SPARQL queries.

Finally, we investigated the capabilities of SGPT and the baselines models on diverse types of questions, depicted in Figure 6.5 and 6.6. The improvements over all the baselines across benchmark

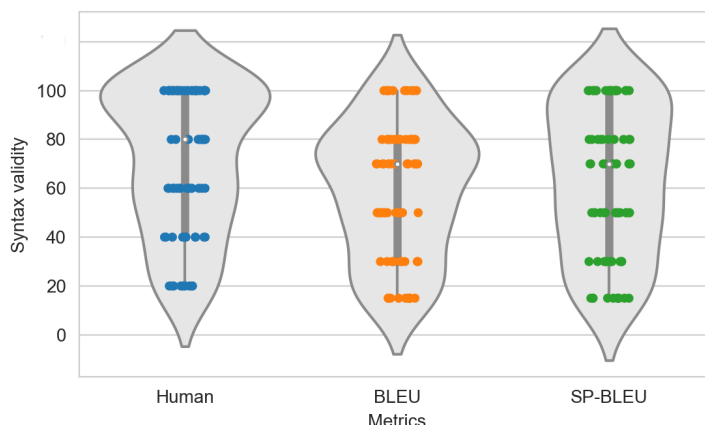


Figure 6.7: Human evaluation score distribution.

datasets confirms SGPT’s capacity to handling diverse types of questions.

### 6.3.6 Qualitative Results

We conducted a human evaluation to assess the system generated SPARQL queries. We observed that SGPT is capable of generating queries with correct syntax, reported in Table 6.7. *Syntax validity* indicates that the generated queries are structure-wise correct. Meaning the parentheses are matching, and the subject, predicate, and object are in their respective orders, although their actual ID might be incorrect, whereas *Content validity* represents the correctness of the entity and relation IDs. We also noticed that template-based approaches obtained comparatively low *Syntax validity* score because they failed to generate queries in some cases. Overall, generative systems received a high *Syntax validity* score as they learned the SPARQL pattern well. Figure 6.7 depicts the score distribution of human annotation and corresponding BLEU and SP-BLEU scores from automatic metrics. Human judgements are normalized to a scale of 0 to 100. The Spearman correlation co-efficient between BLEU and human judgement is 0.94, where for SP-BLEU and human judgement it is 0.97. This confirms that our proposed normalization algorithm enables the metric to correlate better with human judgement.

## 6.4 Analysis

### 6.4.1 Ablation Study

Table 6.8 summarizes the results of the ablation study conducted to investigate how various components of SGPT affect its overall performance. The *seq2seq* approach denotes the SGPT model without the special layers: POS-tag embedding, dependency relation embedding and dependency level embedding. The results in Table 6.8 exhibit that adding syntactic features improves SGPT’s capability to understand the question and generate the correct query. A remarkable gain in the performance is noticeable after adding of the POS-tag and dependency relation embedding layers, in both  $SGPT_{\mathcal{K}}$  and  $SGPT_{\mathcal{K},Q}$ . Adding dependency level embedding which captures information about token’s immediate syntactic dependents, however, only slightly improved the results further.

Approach	SP-BLEU	$\Delta$	SP-F1	$\Delta$
SGPT <sub>Q</sub> (seq2seq)	50.40	-	71.66	-
+ POS-tag emb.	56.74	6.34 ↑	76.92	5.26 ↑
+ Dep. relation emb.	62.91	6.17 ↑	84.21	7.29 ↑
+ Dep. level emb.	63.28	0.37 ↑	86.17	1.96 ↑
SGPT <sub>Q,κ</sub> (seq2seq)	67.76	-	82.00	-
+ POS-tag emb.	72.84	5.08 ↑	88.15	6.15 ↑
+ Dep. relation emb.	77.16	4.32 ↑	91.39	3.24 ↑
+ Dep. level emb.	77.73	0.57 ↑	92.27	0.88 ↑

Table 6.8: Ablation study.

### 6.4.2 Case Study

Table 6.9 shows two NLQs with corresponding reference query and SPARQL queries, generated by the compared systems. The first NLQ is from LCQuAD 2.0 (Wikidata-based), and the second is from the QALD-9 dataset (DBpedia-based). In the first case where no additional knowledge is provided, SGPT<sub>Q</sub> generated a query with correct content and syntax, whereas NSpM failed to generate the correct content. This demonstrates SGPT’s capabilities of understanding the question and generating a query with correct content from the KG. Despite having additional knowledge provided for a challenging question (second case), SQG and TeBaQA failed to generate a correct query. They could not find a template that could both classify and fill in all the slots correctly. This exhibits the advantage of SGPT over template-based and slot-filling approaches in handling complex query patterns.

Question	System	SPARQL query
Does cobalt have a time-weighted average exposure limit of .1?	Reference	<b>ASK WHERE</b> { wd:Q740 wdt:P2404 ?obj <b>FILTER</b> (?obj = 0.1) }
	NSpM	<b>ASK WHERE</b> { wd:Q1049389 wdt:P3737 ?obj <b>FILTER</b> (?obj = 12) }
	SGPT <sub>Q</sub>	<b>ASK WHERE</b> { wd:Q740 wdt:P2404 ?obj <b>FILTER</b> ( ?obj = 0.1 ) }
Which countries have places with more than two caves?	Reference	<b>SELECT DISTINCT</b> ?uri <b>WHERE</b> { ?cave rdf:type dbo:Cave ; dbo:location ?uri . ?uri rdf:type dbo:Country } <b>GROUP BY</b> ?uri <b>HAVING</b> ( <b>COUNT</b> (?cave) > 2 )
	SQG	(no results found)
	TeBaQA	(no results found)
	SGPT <sub>Q,κ</sub>	<b>SELECT DISTINCT</b> ?uri <b>WHERE</b> { ?cave rdf:type dbo:Cave ; dbo:location ?uri . ?uri rdf:type dbo:Country } <b>GROUP BY</b> ?uri <b>HAVING</b> ( <b>COUNT</b> ( ?cave ) > 2 )

Table 6.9: Case study showing a comparison between SGPT and baseline system’s outputs.

### 6.4.3 Effectiveness of Entity Masking Strategy

Masking entities and relations in the question is a widely adopted strategy for generating and classifying SPARQL query templates. In NSpM [200], all entities in a question are masked with a generic <A> token. During inference, the final query is obtained by replacing <A> with all possible entity



	LC-QuAD 2.0		VQuAnDa		QALD-9	
	F1 (E)	F1 (R)	F1 (E)	F1 (R)	F1 (E)	F1 (R)
NSpM	41.52	51.09	29.19	34.62	33.46	30.56
SGPT <sub>Q</sub> (ours)	67.22	83.38	89.95	69.26	40.27	45.21
SQG	-	-	45.31	45.69	55.00	47.17
TeBaQA	-	-	20.71	16.93	48.25	33.21
SGPT <sub>Q,κ</sub> (ours)	97.75	83.60	97.74	70.88	79.14	48.39

Table 6.10: Performance of entity and relation generation.

combinations and ranking. Similarly, in TeBaQA the slots in the predicted template are filled in by checking all possible indexed entities and relations. In contrast, our proposed masking strategy in SGPT<sub>Q,κ</sub> eliminates the need for any slot-filling component. The entity masking strategy used in SGPT<sub>Q,κ</sub>'s training allows the system to learn the patterns of entity positions in the question and generate corresponding correct query. SGPT<sub>Q,κ</sub> achieves an absolute 3.8%, 1.9%, and 1.1% increase of BLEU score on LC-QuAD 2.0, VQuAnDA, and QALD-9 respectively, when the entities are masked in the input sequence instead of keeping their initial mentions.

#### 6.4.4 Effective Entity and Relation Generation

Table 6.10 shows the study results, which we conducted to investigate how well our proposed model learns the KG in its parameters. The performance suggest that SGPT can learn the knowledge graph in its parameters with high accuracy. The metric F1 (E) denotes the F1 scores between the entity sets of ground truth and system generated queries. Similarly, F1 (R) shows the performance of relation prediction. Despite given the correct knowledge, SQG and TeBaQA failed to achieve high F1-scores for entity and relation linking. This is due to the fact that the generated query may include entities from the NLQ as well as intermediate entities and relations that are not explicitly present in the NLQ. The intermediary entities and relations are required to resolve the answers, which is dependent on the complexity of the question and hence cannot be specified in a template-based setting.

#### 6.4.5 Error Analysis and Limitations

We performed an error analysis to inspect whether SGPT has not generated correct SPARQL queries. Table 6.11 shows such erroneous examples where the first one shows an error of SGPT<sub>Q,κ</sub> in generating the wrong masked query. Although the system could infer that the question is about death, it predicted the wrong, though similar relation `dbo:deathPlace` instead of `dbp:placeOfDeath`. The first two error cases are from DBpedia-based questions where the third example is based on Wikidata.

In the second case, SGPT<sub>Q</sub> correctly detected the query type and the topic about British Columbia. However, it generated the wrong entity `dbr:British_Columbia_republic` instead of `dbr:British_Columbia`. Similarly, in the third example, the system could infer, that the question is about a state, but predicted a Wikidata entity ID with the wrong type of state Q3624078 (sovereign state) instate of Q842112 (socialist state). Despite the failed cases, the generated queries in Table 6.11 confirm SPGT's capability of generating queries with correct syntax and query type.

Since SGPT learns the graph patterns in the model's parameters, fine-tuning is required if the graph

Table 6.11: Three error cases where the texts highlighted in green indicate the correct entry in the reference query and red indicating wrong predication in the system generated query. The text in yellow shows the masked entity.

Question	System	SPARQL query
Where did the designer of ENT1 die ?	Reference	SELECT DISTINCT ?uri WHERE { ENT1 dbo:designer ?x . ?x dbp:placeOfDeath ?uri . }
	SGPT <sub>Q,κ</sub>	SELECT DISTINCT ?uri WHERE { ENT1 dbo:designer ?x . ?x dbo:deathPlace ?uri . }
List all the faiths that British Columbian politicians follow ?	Reference	SELECT DISTINCT ?uri WHERE { ?x dbp:residence dbr:British_Columbia . ?x dbp:religion ?uri . ?x <http://www.w3.org/1999/02/22-rdf-syntax-nstye> dbo:Politician }
	SGPT <sub>Q</sub>	SELECT DISTINCT ?uri WHERE { ?x dbp:residence dbr:British_Columbia_republic . ?x dbp:religion ?uri . ?x <http://www.w3.org/1999/02/22-rdf-syntax-nstye> dbo:Politician }
When did socialist state for contains administrative territorial entity of Beijing ?	Reference	SELECT DISTINCT ?sbj WHERE { ?sbj wdt:P150 wd:Q956 . ?sbj wdt:P31 wd:Q842112 }
	SGPT <sub>Q</sub>	SELECT DISTINCT ?sbj WHERE { ?sbj wdt:P150 wd:Q956 . ?sbj wdt:P31 wd:Q3624078 }

is updated. Nevertheless, the proposed training techniques allow the system to learn intermediary graph patterns required to generate a complete SPARQL query, that are not detectable from the input question. The current version of this work only supports English language. To adapt SGPT for other languages, a POS-tagger, a dependency parser and a pre-trained language model of the target language are required. Despite the limitations, SGPT comes with the advantages of a training facility without query templates, adaptable to arbitrary KG, and extendable for pipeline-based systems.

## 6.5 Summary

We have presented SGPT, a SPARQL query generation system, improving the state-of-the-art across multiple benchmark datasets. Our proposed training technique eliminates the need for manual annotation and is applicable to arbitrary RDF datasets. The key contributions of SGPT include **1**) a new encoding technique for the linguistic features of a question and (optionally) entities in the question, that allows deeper question understanding during SPARQL generation, **2**) training techniques that leverage a pre-trained language model to generate a SPARQL query and can be adapted to questions from different knowledge graphs, **3**) improved evaluation metrics to measure the performance of SPARQL query generation. An extensive empirical assessment confirms SGPT effectiveness in handling diverse types of questions and generating correct SPARQL queries. This answers the research question RQ3, "Can a generative language model embed a knowledge graph in its parameters and learn to construct SPARQL queries?". The current version of the work only supports English language. We open source the code and model <sup>2</sup>.

<sup>2</sup><https://github.com/rashad101/SGPT-SPARQL-query-generation>

# Question Answering Over Unstructured Knowledge

---

The previous chapters (i.e., Chapter 4, 5, and 6) focused on conversation and question answering over structured knowledge such as knowledge graphs. This chapter demonstrates a machine reading comprehension application developed by incorporating unstructured text into a learning method and fine-tuning pre-trained language models for question answering on climate change-related unstructured text. The system demonstrator developed in this chapter provides an easy-to-use interface for question answering on climate change data. The content of this chapter is based on the following paper:

- **Md Rashad Al Hasan Rony**, Ying Zuo, Liubov Kovriguina, Roman Teucher, and Jens Lehmann, *Climate Bot: A Machine Reading Comprehension System for Climate Change Question Answering*. In Proceedings of IJCAI 2022, in AI for good track.

The content of this chapter is organized into five sections. Section 7.1 discusses the impact of climate change on the Earth and briefly summarizes the contribution of this chapter. Section 7.2 describes components of the proposed *Climate Bot*. Next, Section 7.3 provides a brief description of the proposed machine reading comprehension dataset on the climate change domain. Section 7.4 demonstrates empirical results conducted in this chapter. Finally, Section 7.5 summarizes the chapter by stating the chapters' contribution and future direction.

## 7.1 Introduction

The impacts of climate change are global in scope and may threaten species and people communities' survival. Among the most serious threats is the growing temperature of the Earth's atmosphere, causing sea levels to rise, ecosystems to collapse, and catastrophic weather events to become more common. Leveraging machine learning (ML) and Artificial Intelligence (AI) has already helped mitigate climate change effects. This has been done using various ML tasks involving predictive modeling, i.e., natural hazard prediction, reducing factory emissions, modeling temperature changes, and ice melting. Conversational AI applications in this domain are not yet numerous. Applying machine reading comprehension (MRC) over climate change documents can expand the benefits of question answering interfaces to this area, such as faster answer spotting in comparison to traditional

search, natural human-machine interaction, and insights extraction from massive document collections. Moreover, the educational impact of such applications is hard to overestimate.

To help people know more about climate change from trusted data sources, we have created a dataset for training MRC and designed and implemented an MRC pipeline providing question answering services over climate change problems and challenges. The motivation behind this work is to speed up access to information about climate change challenges and promote awareness about it by allowing natural questions over trusted sources. We open-source the data and code used in this demonstrator<sup>1</sup>. A video demonstrator of the climate bot can be found on Youtube<sup>2</sup>.

### Contributions

- Climate Bot, a novel MRC system for question answering over climate change documents with publicly available code.
- A climate change dataset CCMRC, as a manually annotated publicly available resource for training QA and MRC applications, having 21,081 question-answer pairs and 7,400 paragraphs, extracted from trusted data sources.

## 7.2 Climate Bot System

The primary goal of the Climate Bot is to perform Machine Reading Comprehension in the climate change domain. Given a user question  $Q$ , the climate bot first fetches documents ( $\mathcal{D}_n$ ) relevant to  $Q$ . Then the system displays the documents and answers highlighted in them using a web interface. Here,  $n$  is the number of documents. Climate bot consists of three main components: 1) a *Retriever* 2) a *Reader*, and 3) a *User Interface (UI)*.

### 7.2.1 Retriever.

The task of the *Retriever* component is to fetch documents relevant to the user question. An indexing step prepares the documents for the retrieval task, whereas a lookup step retrieves relevant documents to answer the question. The steps are described below.

*Indexing*: To set up the retriever, we first pre-process all the documents  $\mathcal{D}$ . Each document  $d \in \mathcal{D}$  is passed through Sentence-BERT [47] to get a contextualized vector representation ( $\vec{d}$ ) of the document which is indexed into a dense space using an hierarchical indexing algorithm from Dense Passage Retriever (DPR) [102]. Indexing the contextualized documents into a dense space closely clusters documents with similar types and content, allowing the climate bot to find relevant documents quickly and efficiently. Only the data pre-processing steps need to be executed to extend the climate change bot with more data. The data pre-processing steps are depicted in Figure 7.1.

---

<sup>1</sup> <https://github.com/rashad101/Climate-Bot-IJCAI22>

<sup>2</sup> <https://youtu.be/DdRh6P4sgQw>

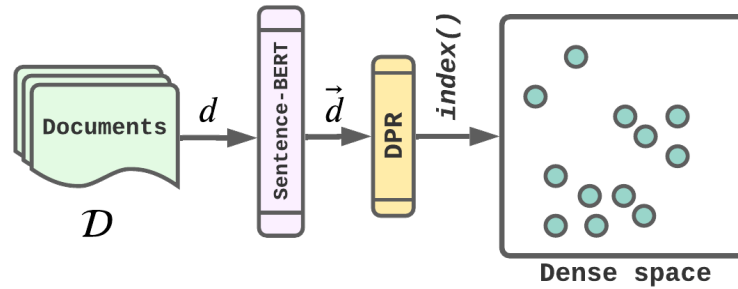


Figure 7.1: The data pre-processing pipeline, showing how documents are stored into a dense space.

*Lookup:* An approximate search algorithm from the Dense Passage Retriever (DPR) [102] is employed to retrieve user-question relevant documents from the indexed dense space. Specifically, first, the contextualized vector of the question is obtained from Sentence-BERT [47]. The DPR then utilizes the contextualized representation to perform lookup following a K-nearest neighbor (KNN) approximate search algorithm and fetch  $n$  number of documents relevant to the user question. The value of  $n$  can be configured from the user interface of the system demonstrator.

### 7.2.2 Reader.

The task of the *Reader* component is to extract a text span from the documents that answers the user question. We leverage the language model ALBERT [48] to extract an answer span given a question and a document as input. The ALBERT model was previously pre-trained on the SQuAD [19] dataset, a widely used cross-domain MRC dataset. We fine-tune the pre-trained ALBERT model on the climate change data. The Reader component extracts one answer per document, which is then displayed in a user interface.

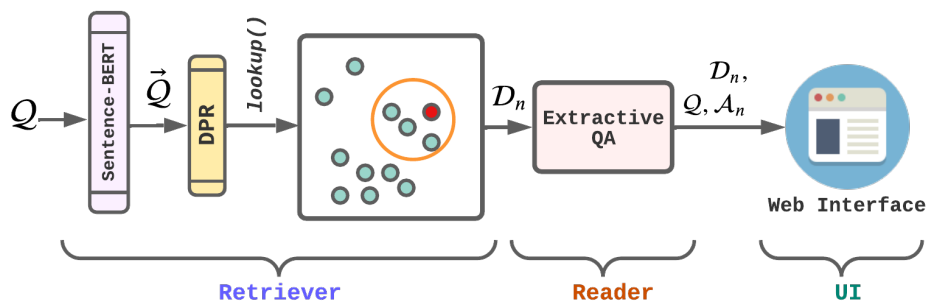


Figure 7.2: System architecture.

### 7.2.3 User Interface (UI).

We developed a web interface that allows a user to type questions and receive the most relevant documents along with highlights of the answer to the question inside the documents (see screenshot in Figure 7.3).

The system architecture is illustrated in Figure 7.2. Furthermore, the system demonstrator we

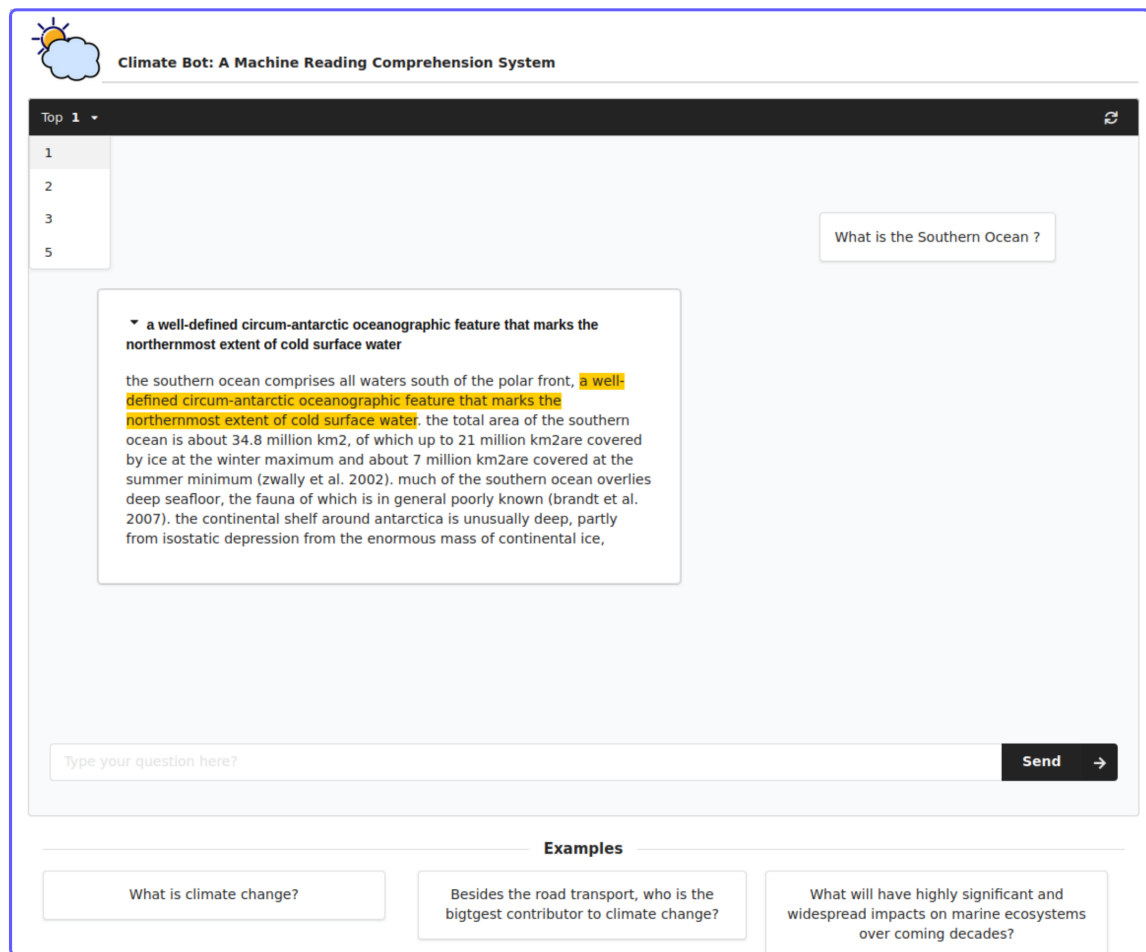


Figure 7.3: System demonstrator.

developed is shown in Figure 7.3. The main functionalities of the demonstrator are described below:

- Example questions, to give the user a starting point to try out our system. Example questions are clickable cards, located at the bottom part of the demonstrator. Once clicked, the question and its answer will pop up in the chat section.
- An input field, where users can type in their question and press *Enter* in their keyboard or click on the *Send* button to get the answer in the same way as with the example questions.
- The main body of the demonstrator is designated for showing the question and answer. The answer is shown in a card where the first line shows the answer in **bold**, followed by a document fetched by the *Retriever* wherein the answer is highlighted in yellow.
- A drop-down button on the top-left corner to configure the value of  $n$  (Top  $n$ ), indicating how many documents the *Retriever* should fetch.

- A reset button, located at the top-right corner of the demonstrator to clear excessive chat contents.

## 7.3 CCMRC: Climate Change Dataset

### 7.3.1 Data Sources and Acquisition

We collected data from various trusted data sources. The data from each source was pre-processed and split into documents/paragraphs. Given these documents, we asked the Amazon Mechanical Turk (AMT) and in-house annotators from the Smart Data Analytics group to manually write question-answer pairs. The trusted data sources used to construct the dataset are listed below.

- **Official Semantic Scholar Dump**<sup>3</sup>, where the research articles with the words *Climate* or *Climate change* in their title, are selected.
- The official reports of the **Intergovernmental Panel on Climate Change Special Reports on Climate Change** for the years 2019-2021<sup>4</sup>.
- **NASA Global Climate Change**<sup>5</sup>.
- **European Commission Climate Change Data**<sup>6</sup>.
- Individual documents and news articles from **CNN, The Guardian, National Geographic, New York Times, World Health Organization**.

We have implemented PDF-parser and used third-party libraries to collect the research articles and reports listed above.

### 7.3.2 Data Annotation

Each article was split into paragraphs/documents for manual annotation. To keep the quality of annotated question-answer pairs consistent, documents with a word count of less than 150 words were excluded from manual annotation. As the train set, 7,527 paragraphs were manually annotated. We asked Amazon's Mechanical Turk (AMT) annotators to write three question-answer-pairs for each document obtained from *Semantic Scholar* articles. The annotated question-answer-pairs were then transformed to the widely used SQuAD [19] dataset format for training and testing of the reader model.

To obtain a realistic estimation of the test performance of the trained question-answer (QA) system, we created an additional test set, from here on referred to as the in-house annotation test set, from the listed data sources except for *Semantic Scholar*. The in-house annotation test set contains documents from 30 articles. Figure 7.4 shows the annotation tool used for generating the in-house test set. The annotation tool allows the annotators to download the annotated data in SQuAD format. Five QA-pairs from each of the 30 articles annotated by the in-house annotators were randomly sampled to assess the

---

<sup>3</sup><https://api.semanticscholar.org/corpus/download/>

<sup>4</sup><https://www.ipcc.ch/reports/>

<sup>5</sup><https://climate.nasa.gov/ask-nasa-climate/>

<sup>6</sup>[https://ec.europa.eu/clima/climate-change\\_en](https://ec.europa.eu/clima/climate-change_en)

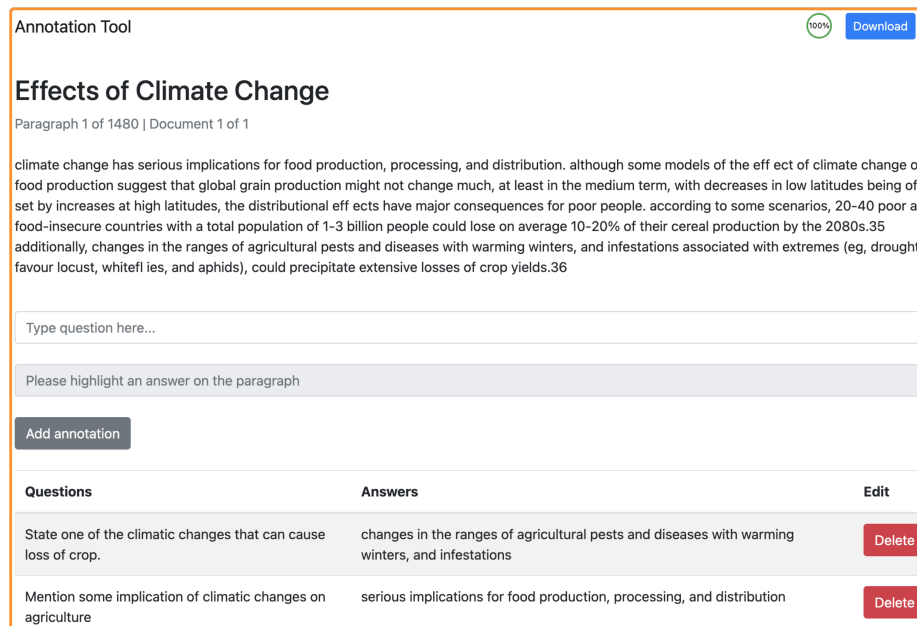


Figure 7.4: The in-house annotation tool used to collect question answer pairs for training the *Reader* module.

quality of annotated QA-pairs. Each QA-pair was evaluated by three different cross-validators who were not previously involved in the annotation process.

We asked each cross-validator to rate the question and answer individually. Based on the grammar, the relevance and the contextual meaning of the QA-pair, each question and answer were scored from 1 to 4, where 1 corresponds to ‘*poor*’ and 4 corresponds to ‘*excellent*’. The inter-annotator agreement score (Cohen’s Kappa  $\kappa$ ) of the evaluated data is 0.86. The evaluation guidelines are provided in our Github repository <sup>7</sup>.

### 7.3.3 Dataset Statistics

The dataset statistic is reported in Table 7.1. There are 7,400 paragraphs and 21,081 QA-pairs used for training and testing the reader model of the Climate bot system. The AMT annotated data were split at the paragraph level into train, validation, and test sets with the ratio of 70/20/10. We created a set of 960 QA-pairs from 495 documents as the in-house annotation test set.

## 7.4 Evaluation

We used automatic metrics (F1 score, BLEU [32] and METEOR [33]) to evaluate the performance of the *Reader* component. It is noteworthy that the *Retriever* component in Climate Bot works in an unsupervised manner. The evaluation result is reported in Table 7.2. We report the performance of the Reader module with and without fine-tuning. The evaluation results demonstrate that the climate bot can answer climate-related user questions with high accuracy. It also reveals that fine-tuning

<sup>7</sup> <https://github.com/SmartDataAnalytics/Climate-Bot>



	<b>Train</b>	<b>Validation</b>	<b>Test</b>
Number of paragraphs	5,180	1,480	740
Number of QA-pairs	14,756	4,229	2,096
Avg. word count (question)	9.59	9.49	9.57
Avg. word count (document)	212.80	210.92	208.01
Avg. word count (answer)	26.06	25.80	25.81

Table 7.1: Dataset statistics.

	<b>F1 score</b>	<b>BLEU</b>	<b>METEOR</b>
Test without fine-tuning	0.672	0.551	0.606
Test	0.816	0.678	0.808
In-house annotation test without fine-tuning	0.438	0.332	0.419
In-house annotation test	0.661	0.416	0.694

Table 7.2: Performance of the *Reader* component.

significantly improves the question answering performance. Because of the fine-tuning, the Reader can now capture climate change-related terminologies better. The proposed climate bot is developed in a modular way, allowing the system to be extendable with minimal effort. We evaluated the quality of the in-house test set by calculating an average over the cross-validators ratings for the questions and answers separately. Given the scale, we received an average score of 3.69 for the questions' quality and 3.79 for the answers' quality.

## 7.5 Summary

We presented Climate Bot, an MRC system for question answering about climate change. The key contributions of this work include 1) A question answering system on climate change and 2) A machine reading comprehension dataset on climate change. The experiment results demonstrate the performance improvement in question answering over unstructured text after fine-tuning a pre-trained language model over a pre-trained only model. Additionally, we made the annotated dataset CCMRC and code open source to encourage further research on the climate change domain.



---

## Dialogue System Evaluation

---

We developed and discussed conversational and question answering systems in the previous chapters (i.e., Chapter 4, 5, 6, and 7). On the contrary, this chapter aims to develop an evaluation metric to measure the performance of generative systems such as dialogue systems. Evaluation metrics are one of the key elements for developing any intelligent system. This chapter addresses the research question RQ4, "**How to effectively employ a pre-trained language model to improve the evaluation of single-reference based generative systems?**". Pre-trained language (PLM) models have revolutionized the field of NLP in recent years. PLMs have become popular because of their capability to understand numerous text patterns. The contextual understanding of these PLMs is embedded in their pre-trained weights. This chapter discusses techniques to evaluate generative systems leveraging pre-trained language models effectively. The content of this chapter is based on the following publication:

- **Md Rashad Al Hasan Rony**, Liubov Kovriguina, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann. 2022. *RoMe: A Robust Metric for Evaluating Natural Language Generation*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5645–5657, Dublin, Ireland. Association for Computational Linguistics.

This chapter is organized into five sections. Section 8.1 describes the limitation of existing metrics and importance of metrics that can tackle various surface forms of the text. Section 8.2 discusses the proposed metric and its components in detail. Section 8.3 describes experiments on various benchmark dataset and their results. A comprehensive robustness analysis is provided in Section 8.4, demonstrating the robustness of the proposed metric in handling various surface forms of system generated sentences. Finally, Section 8.5 summarizes the contribution of this chapter with concluding remarks.

### 8.1 Introduction

Automatic generation of fluent and coherent natural language is a key step for human-computer interaction. Evaluating generative systems such as text summarization, dialogue systems, and machine translation is challenging since the assessment involves several criteria such as content determination, lexicalization, and surface realization [234, 294]. For assessing system-generated outputs, human

judgment is considered to be the best approach. Obtaining human evaluation ratings, on the other hand, is both expensive and time-consuming. As a result, developing automated metrics for assessing the quality of machine-generated text has become an active area of research in NLP.

The quality estimation task primarily entails determining the similarity between the reference and hypothesis as well as assessing the hypothesis for grammatical correctness and naturalness. Word overlap-based metrics (i.e., BLEU [32], METEOR [33], and ROUGE [112]) cannot capture the hypotheses' semantic similarity to reference, naturalness, and fluency. On the other hand, despite the fact that embedding-based metrics (i.e., WMD [115], BERTScore [239] and MoverScore [113]) employ the contextualized representation of words, they do not consider the grammatical acceptability and syntactical similarity of the hypothesis to the reference.

To address these shortcomings, we propose RoMe, an automatic and robust metric for evaluating NLG systems. RoMe employs a neural classifier that uses the generated sentence's grammatical, syntactic, and semantic qualities as features to estimate the quality of the sentence. **Firstly**, it calculates the earth mover's distance (EMD) [41] to determine how much the hypothesis differs from the reference. During the computation of EMD, we incorporate hard word alignment and soft-penalization constants to handle various surface forms of words in a sentence, such as repeated words and the passive form of a sentence. **Secondly**, using a semantically enhanced tree edit distance, the difference in syntactic structures between the reference and hypothesis sentences is quantified. **Thirdly**, the metric incorporates a binary classifier to evaluate the grammatical acceptability of the generated hypotheses. **Finally**, the scores obtained from the preceding steps are combined to form a representation vector, which is subsequently fed into a self-supervised network. The network produces a final score, referred to as RoMe's output which represents the overall quality of the hypothesis statement.

We investigate the effectiveness of our proposed metric by conducting experiments on datasets from various domains of NLG such as knowledge graph based language generation dataset (KELM [295]), dialogue datasets [296, 12], the WebNLG 2017 challenge dataset [297], structured data to language generation dataset (BAGEL [298] and SFHOTEL [123]). The capability of existing metrics to handle various forms of text has lately become a matter of debate in the NLP community [299, 258, 234]. Hence, we conduct an extensive robustness analysis to assess RoMe's performance in handling diverse forms of system-generated sentences. To verify our claim, we design the analysis based on the text perturbation methods used in CHECKLIST [299] and adversarial text transformation techniques from TextFooler [300] and TextAttack [301]. The contributions of this paper can be summarized as follows:

#### Contributions

- A robust evaluation metric to assess the performance of generative systems, considering the semantic, syntactic, and grammatical acceptability of the generated sentences.
- A comprehensive robustness analysis that demonstrates the superior performance of RoMe in handling various surface forms of the generated sentences.

## 8.2 Approach: RoMe

In RoMe, a neural network determines the final evaluation score given a reference-hypothesis pair. The network is trained to predict the evaluation score based on three features: semantic similarity computed by EMD, enhanced TED, and the grammatical acceptability score. We explain these features in the following subsections.

### 8.2.1 Earth Mover's Distance Based Semantic Similarity

The Earth Mover's Distance (EMD) estimates the amount of work required to transform a probability distribution into another [41]. Let us define the reference as  $\mathcal{R} = \{r_1, r_2, \dots, r_p\}$  and the hypothesis as  $\mathcal{H} = \{h_1, h_2, \dots, h_q\}$ , where  $r_i$  and  $h_j$  indicates the  $i$ -th and  $j$ -th word of the reference and hypothesis, respectively. The weight of the word  $r_i$  and  $h_j$  are denoted as  $m_i$  and  $n_j$  respectively. Then, the total weight distribution of  $\mathcal{R}$  and  $\mathcal{H}$  is  $m_\Sigma = \sum_{i=1}^p m_i$  and  $n_\Sigma = \sum_{j=1}^q n_j$ , respectively. Here, the sentence-level and normalized TF-IDF score of a word is considered as the word's weight. Formally, EMD can be defined as:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})} \sum_{i=1}^p \sum_{j=1}^q d_{ij} f_{ij}}{\min(m_\Sigma, n_\Sigma)} \quad (8.1)$$

where  $d_{ij}$  is the distance between the words  $r_i$  and  $h_j$  in the space and  $\mathcal{F}(\mathcal{H}, \mathcal{R})$  is a set of possible flows between the two distributions that the system tries to optimize. In Equation 8.1,  $EMD(\mathcal{H}, \mathcal{R})$  denotes the amount of work required to match the hypothesis with the reference. The optimization is done following four constraints:

$$\begin{aligned} f_{ij} &\geq 0 & i = 1, 2, \dots, p \text{ and } j = 1, 2, \dots, q, \\ \sum_{j=1}^q f_{ij} &\leq m_i & i = 1, 2, \dots, p, \\ \sum_{i=1}^p f_{ij} &\leq n_j & j = 1, 2, \dots, q, \\ \sum_{i=1}^p \sum_{j=1}^q f_{ij} &= \min(m_\Sigma, n_\Sigma) \end{aligned} \quad (8.2)$$

The first constraint indicates that each flow must be non-negative. The second constraint limits the total weights flowing from  $r_i$  to less than or equal to  $m_i$ . Similarly, the third constraint restricts the total weights flowing from  $h_j$  to less than or equal to  $n_j$ . The final constraint indicates that the total flow of weights must be equal to the minimum weight distribution.

During the computation of EMD, we employ *hard word alignment* and *soft-penalization* techniques to tackle repetitive words and passive forms of a sentence. We compute a distance matrix and a flow matrix as described below and finally obtain EMD utilizing Equation 8.1.

#### Hard Word Alignment

We first align the word pairs between reference and hypothesis based on their semantic similarities. The alignment is performed by computing all paired cosine similarities while taking word position

information into account, as in [35]. In contrast to [35], we use contextualized pre-trained word embedding from the language model ALBERT [48]. ALBERT uses sentence-order prediction loss, focusing on modeling inter-sentence coherence, which improves multi-sentence encoding tasks. The word alignment score is computed as follows:

$$\mathcal{A}(r_i, h_j) = \frac{\vec{r}_i \cdot \vec{h}_j}{\|\vec{r}_i\| \|\vec{h}_j\|} \cdot \frac{|q(i+1) - p(j+1)|}{pq} \quad (8.3)$$

where  $\vec{r}_i$  and  $\vec{h}_j$  denote the contextualized word embedding of  $r_i$  and  $h_j$ , respectively. The first part of the right side of the equation computes the cosine similarity between  $\vec{r}_i$  and  $\vec{h}_j$ , and the second part calculates the relative position information as proposed in [35].

		Reference						
		tesla	motors	is	founded	by	elon	musk
Hypothesis	elon	0.11	0.14	0.17	0.10	0.07	0.34	0.07
	musk	0.11	0.20	0.23	0.13	0.08	0.20	0.40
	founded	0.08	0.14	0.18	0.97	0.23	0.12	0.09
	tesla	0.34	0.20	0.09	0.11	0.15	0.11	0.12
	motors	0.06	0.29	0.07	0.12	0.14	0.14	0.23

Figure 8.1: An example word alignment matrix for the reference sentence: "tesla motors is founded by elon musk" and its passive form: "elon musk founded tesla motors" is illustrated here.

Figure 8.1 depicts a matrix of word alignment scores generated on an example pair of sentences. This alignment strategy fails to handle repetitive words where a word from the hypothesis may get aligned to several words in the reference (see Figure 8.2). To tackle such cases, we restrict the word alignment by imposing a hard constraint. In the hard constraint, we prevent the words in the hypothesis from getting aligned to multiple words in the reference as illustrated by the dotted arrows in Figure 8.2. We denote the resulting set of hard-aligned word pairs as  $\mathcal{A}_{hc}$ .

### Transport Distance

A distance matrix  $\mathcal{D}$  is required to compute the final EMD score. For each aligned pair  $(r_i, h_j) \in \mathcal{A}_{hc}$  where  $\frac{\vec{r}_i \cdot \vec{h}_j}{\|\vec{r}_i\| \|\vec{h}_j\|} > \delta$ , the distance between  $r_i$  and  $h_j$  is computed as follows:

$$d_{ij} = 1.0 - \frac{\vec{r}_i \cdot \vec{h}_j}{\|\vec{r}_i\| \|\vec{h}_j\|} \cdot e^{\gamma \cdot \frac{|q(i+1) - p(j+1)|}{pq}} \quad (8.4)$$

where  $d_{ij} \in \mathcal{D}$  and  $\delta$  is a confidence threshold found via hyper-parameter search,  $\gamma \in [-1, 0)$  is a soft-penalization constant. For all the non-hard-aligned pairs and aligned pairs with value less than

$\delta$ , the distance  $d_{ij}$  receives a maximum value of 1.0. Intuitively, a lower value of  $d_{ij}$  implies that the word needs to travel a shorter distance in the transportation problem of EMD. In Equation 8.4,  $e^{\gamma \cdot \frac{|q(i+1)-p(j+1)|}{pq}}$  works as a penalty where a higher position difference multiplied with the negative constant  $\gamma$  will result in low  $d_{ij}$  score. The role of  $\gamma$  is explained below.

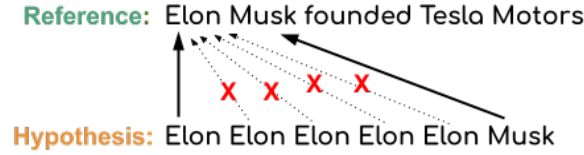


Figure 8.2: An example hypothesis containing repetitive words.

### Soft-penalization

Existing metrics often impose hard penalties for words with different order than the reference sentence [113, 35]. For instance, sentences phrased in the passive form obtain a very low score in those metrics. Addressing this issue, we introduce a soft-penalization constant  $\gamma = -\frac{|j-i|}{\max(p,q)}$  in Equation 8.4 to handle the passive form of a sentence better. Let us consider a reference, "*Shakespeare has written Macbeth*" and the passive form of the sentence as hypothesis, "*The Macbeth is written by Shakespeare*". The word *Shakespeare* appears at the beginning of the reference and at the end of the hypothesis, thus the position difference is larger. In such scenario,  $\gamma$  imposes a lower penalty as it divides the position difference by the length  $\max(p, q)$ .

Finally, following the optimization constraints of Equation 8.2, we obtain the transportation flow  $\mathcal{F}(\mathcal{H}, \mathcal{R})$ . For the optimized flow  $f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})$ , the final equation of EMD is as follows:

$$EMD(\mathcal{H}, \mathcal{R}) = \frac{\min_{f_{ij} \in \mathcal{F}(\mathcal{H}, \mathcal{R})} \sum_{i=1}^p \sum_{j=1}^q d_{ij} f_{ij}}{\min(m_{\Sigma}, n_{\Sigma})} \quad (8.5)$$

The semantic similarity between hypothesis and reference is denoted as  $\mathcal{F}_{sem} = 1.0 - EMD$ . The normalized value of EMD is used to calculate  $\mathcal{F}_{sem}$ .

### 8.2.2 Semantically Enhanced TED

To estimate the difference between the syntactic structures of reference and hypothesis, we extend the TED algorithm [116]. The original TED algorithm performs edit operations based on an exact match between two nodes in the dependency trees of hypothesis and reference. In this work, we modify the TED algorithm and compute a word embedding-based cosine similarity to establish the equivalence of two nodes. Two nodes are considered equal, if the cosine similarity of their embedding representations exceeds the threshold  $\theta$ . This allows the semantically enhanced TED to process synonyms and restricts it from unnecessary editing of similar nodes. We call the resulting algorithm TED-SE. The normalized value of TED-SE is denoted as  $\mathcal{F}_{ted}$ . We compute TED-SE over the lemmatized reference and hypothesis since lemmatized text exhibits improved performance in such use cases [302]. The lemmatizer and dependency parser from Stanza [303] are utilised to obtain the tree representation of the text.

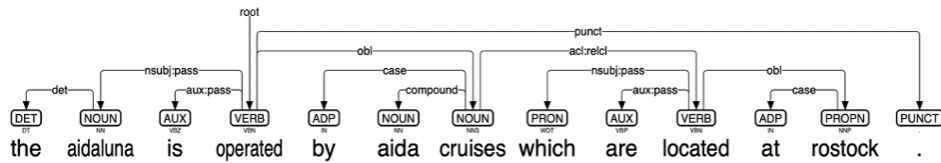
Let us consider a reference statement "the aidaluna is operated by aida cruises which are located at rostock." and a hypothesis, "aida cruises, which is in rostock, operates aidaluna.". First, a dependency tree is parsed utilizing the Stanza dependency parser [303] and then converted to an adjacency list. The adjacency list contains a key-value pair oriented data structure where each key corresponds to a node's index in the tree, and the value is a list of edges on which the head node is incident. List of nodes and adjacency lists are then fed into the TED-SE algorithm to calculate semantically enhanced tree edit distance. Figure 8.3 demonstrates the dependency trees and their corresponding adjacency lists for the given reference and hypothesis.

### 8.2.3 Grammatical Acceptability Classification

Linguistic competence assumes that native speakers can judge the grammatical acceptability of a sentence. However, system-generated sentences are not always grammatically correct or acceptable. Therefore, we train a binary classifier on the Corpus of Linguistic Acceptability (CoLA) [304], predicting the probability that the hypothesis is grammatically acceptable. CoLA is a collection of sentences from the linguistics literature with binary expert acceptability labels containing over 10k examples [304]<sup>1</sup>. The classifier is based on BERT-large [45] and trained to optimize binary cross-entropy loss. A text sequence is fed as input and as output, the classifier produces the class

**Ref: the aidaluna is operated by aida cruises which are located at rostock.**

**Dependency tree:**

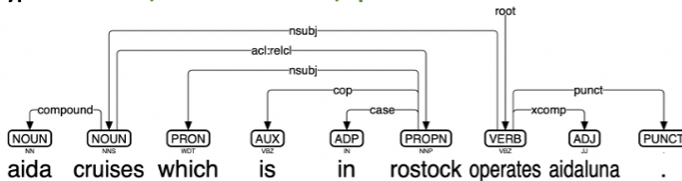


**Adjacency list:** [ {0: [], 1: [0], 2: [], 3: [1, 2, 6], 4: [], 5: [], 6: [4, 5, 9], 7: [], 8: [], 9: [7, 8, 11], 10: [], 11: [10]}]

**Nodes:** ['the', 'aidaluna', 'be', 'operate', 'by', 'aida', 'cruise', 'which', 'be', 'locate', 'at', 'rostock']

**Ref-tree (lemmas):** operate(aidaluna(the), be, cruise(by, aida, locate(which, be, rostock(at))))

**Hyp: aida cruises, which is in rostock, operates aidaluna.**



**Adjacency list:** [ {0: [], 1: [0, 5], 2: [], 3: [], 4: [], 5: [2, 3, 4], 6: [1, 7], 7: []}]

**Nodes:** ['aida', 'cruise', 'which', 'be', 'in', 'rostock', 'operate', 'aidaluna']

**Hyp-tree (lemmas):** operate(cruise(aida, rostock(which, be, in)), aidaluna)

Figure 8.3: Dependency trees of reference and hypothesis, pre-processed for the TED-SE calculation.

<sup>1</sup>with 70.5% examples manually labeled *acceptable*.



membership probability (grammatically acceptable, grammatically unacceptable). The model achieves an accuracy of 80.6% on the out-of-domain CoLA test set [304, p. 8]. We denote the score from the classifier as the feature  $\mathcal{F}_g$ , which is used to train a neural network (see §8.2.4).

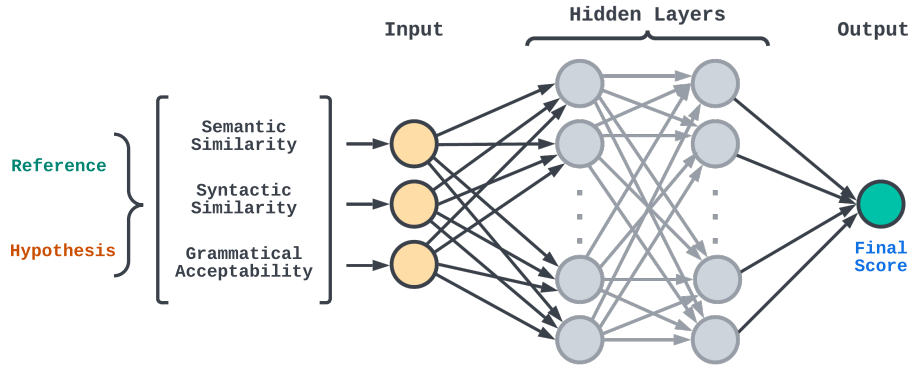


Figure 8.4: A high-level illustration of RoMe.

## 8.2.4 Final Scorer Network

A feed-forward neural network takes the previously computed features as input and learns a function  $f(\mathcal{F}_{sem}; \mathcal{F}_{ted}; \mathcal{F}_g)$  in the final step, yielding a final output score in the  $[0, 1]$  interval. The output score is regarded as the overall quality of the hypothesis. Figure 8.4 depicts a high-level overview of RoMe. Following a self-supervised paradigm, the network is trained on artificially generated training samples from the KELM dataset [295]. We chose KELM because 1) it contains knowledge-grounded natural sentences, and 2) it includes triple-to-text verbalization data where the passive form of sentences frequently appears, which RoMe tries to tackle specifically besides other surface forms. We randomly choose 2,500 sentence pairs from the KELM dataset and generate 2,500 more negative samples by randomly augmenting the sentences using TextAttack [301] and TextFooler [300]. Following a similar approach, we additionally generate 1,000 test sentence pairs from the KELM dataset. Overall, we then have 5,000 training and 1,000 test examples. The network is a simple, two-layered feed-forward network optimized with stochastic gradient descent using a learning rate of  $1e-4$ .

## 8.3 Experiments and Results

### 8.3.1 Data

To assess RoMe’s overall performance, first, we benchmark on two language generation datasets, BAGEL [298] and SFHOTEL [123], containing 404 and 796 data points, respectively. Each data point contains a meaning representation (MR) and a system generated output. Human evaluation scores of these datasets are obtained from [258]. Furthermore, we evaluate dialogue system’s outputs on Stanford in-car dialogues [296] containing 2,510 data points and the soccer dialogue dataset [11] with 2,990 data points. Each data point of these datasets includes a user query, a reference response, and a system response as a hypothesis. Three different system outputs are evaluated for each dialogue

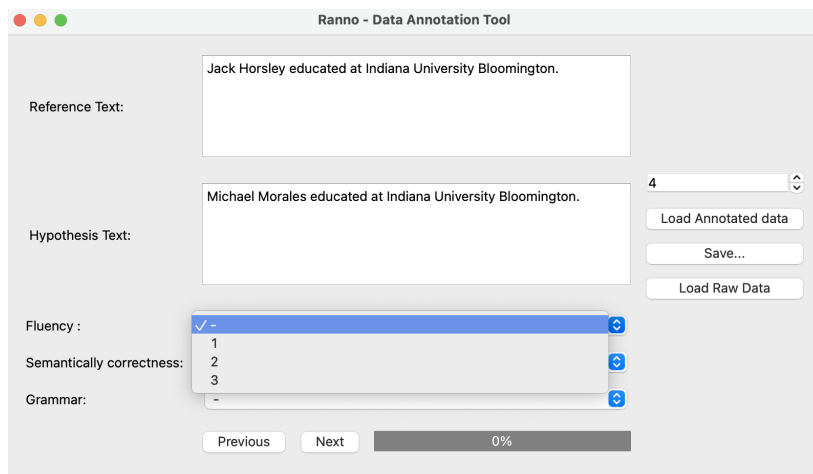


Figure 8.5: The annotation tool used by the annotators.

dataset. We use the human annotated data provided by [12]. Moreover, we evaluate the metrics on the system generated outputs from the WebNLG 2017 challenge [297].

Finally, to conduct robustness analysis, we randomly sample data points from KELM [295] and perturb them with adversarial text transformation techniques. Three annotators participated in the data annotation process (two of them are from a Computer Science and one from a non-Computer Science background), where they annotated the perturbed data. We provided the annotators with an annotation tool which displays the reference sentence and the system output for each data point. The annotators were asked to choose a value from a range of [1,3], for each of the categories: *Fluency*, *Semantic Correctness*, and *Grammatical correctness*. In this case, the values stand for 1: *poor*, 2: *average*, and 3: *good*. The overall inter-annotator agreement score,  $\kappa$  is 0.78.

For all the annotation processes, we use the annotation tool shown in Figure 8.5. The tool is developed using Python programming language. Annotators can load their data into the tool in JSON format by selecting the *Load Raw Data* button. An example annotation step is shown in Figure 8.5. The reference and hypothesis sentences are displayed in different text windows. The annotators were asked to annotate the data based on *Fluency*, *Semantically correctness* and *Grammar*. Annotators can choose a value on a scale of [1,3] for each category, from the corresponding drop-down option. Finally, the annotated text can be saved for evaluation using the *save* button, which saves the annotated data in JSON format.

### 8.3.2 Hyper-parameter Settings

We use  $\delta = 0.60$  and  $\theta = 0.65$  in §8.2.1. Best values are found by a hyper-parameter search from a range of [0,1.0] with an interval of 0.1. RoMe obtained the best result by utilizing ALBERT-large [48] model with 18M parameters and 24 layers. Furthermore, we use the English word embedding of dimension 300 to obtain results from Fasttext [43] throughout the paper. As the grammatical acceptability classifier, we train a BERT-base model with 110M parameters and 12 layers. The hidden layer size is 768 with a hidden layer dropout of 0.1. A layer norm epsilon of  $1e-12$  was used for layer normalization. GELU [250] was used as the activation function. We use a single GPU with 12GBs of memory for all the evaluations.

### 8.3.3 Baselines

We select both the word-overlap and embedding-based metrics as strong baselines. For the experiment and robustness analysis we choose BLEU [32], METEOR [33], BERTScore [239] and MoverScore [113]. We evaluate the metrics on the sentence level to make a fair comparison.

Metrics	BLEU			METEOR			BERTScore			MoverScore			RoMe		
Systems	$\rho$	$r$	$\tau$	$\rho$	$r$	$\tau$	$\rho$	$r$	$\tau$	$\rho$	$r$	$\tau$	$\rho$	$r$	$\tau$
ADAPT	0.38	0.39	0.27	0.57	0.58	0.41	0.61	0.72	0.50	0.68	0.73	0.49	0.72	0.70	0.51
Baseline	0.35	0.42	0.26	0.49	0.49	0.33	0.49	0.50	0.35	0.59	0.61	0.43	0.53	0.53	0.37
melbourne	0.32	0.31	0.21	0.35	0.35	0.24	0.33	0.33	0.26	0.40	0.39	0.28	0.44	0.50	0.35
Pkuwriter	0.37	0.38	0.28	0.47	0.47	0.31	0.48	0.53	0.38	0.57	0.56	0.39	0.58	0.56	0.39
tilburg-nmt	0.25	0.20	0.13	0.26	0.26	0.18	0.38	0.39	0.30	0.49	0.50	0.36	0.64	0.68	0.50
tilburg-pipe	0.38	0.41	0.30	0.52	0.43	0.30	0.53	0.48	0.33	0.62	0.50	0.35	0.38	0.42	0.27
tilburg-smt	0.25	0.20	0.13	0.21	0.19	0.13	0.33	0.30	0.25	0.40	0.38	0.27	0.50	0.51	0.36
upf-forge	0.14	0.13	0.08	0.13	0.11	0.08	0.26	0.25	0.19	0.27	0.27	0.18	0.42	0.42	0.30
vietnam	0.73	0.80	0.62	0.87	0.90	0.72	0.81	0.76	0.70	0.90	0.78	0.73	0.84	0.89	0.83

Table 8.1: Metrics correlation with human judgment on system outputs from the WebNLG 2017 challenge. Here,  $r$ : Pearson correlation co-efficient,  $\rho$ : Spearman’s correlation co-efficient,  $\tau$ : Kendall’s Tau.

### 8.3.4 Results

Table 8.3 shows the performance of different metrics on data to language generation datasets (BAGEL and SFHOTEL). In both the BAGEL and SFHOTEL, a meaning representation (MR), for instance *inform(name='hotel drisco',price\_range='pricey')* is given as a reference sentence, where the system output is: *the hotel drisco is a pricey hotel*, in this case. Although, RoMe outperformed the baseline metrics in evaluating the *informativeness*, *naturalness* and *quality* score, the correlation scores remain low with regard to human judgment. This is because the MR, which is not a natural sentence, is the reference statement in this scenario. For all the experiments, we take the normalized human judgement scores. We firstly evaluate our model using Fasttext [43] word embedding. We notice a significant improvement in results when we replace the Fasttext embedding with contextualized word embedding obtained from BERT [45]. Furthermore, we experiment with multiple language models and finally, we reach to our best performing model with ALBERT-large [48]. In all the experiments, we report the results of RoMe, using ALBERT-large [48]. In Table 8.3, WMD and SDM refer to word mover distance and sentence mover distance, respectively, used in MoverScore. We report the results of WDM and SMD from [113].

Metrics	BLEU			METEOR			BERTScore			MoverScore			RoMe		
Perturbation methods	$f$	$s$	$g$	$f$	$s$	$g$	$f$	$s$	$g$	$f$	$s$	$g$	$f$	$s$	$g$
Entity replacement	0.06	0.04	0.06	0.09	0.09	0.08	0.11	0.07	0.09	0.16	0.13	0.11	0.16	0.19	0.14
Adjective replacement	0.07	0.06	0.07	0.09	0.13	0.11	0.11	0.11	0.13	0.13	0.17	0.16	0.18	0.23	0.18
Random word replacement	0.05	0.06	0.03	0.06	0.06	0.05	0.11	0.10	0.08	0.11	0.13	0.09	0.15	0.15	0.23
Text transformation	0.03	0.01	0.03	0.08	0.09	0.07	0.13	0.15	0.15	0.15	0.18	0.19	0.18	0.19	0.21
Passive form	0.02	0.01	0.04	0.08	0.10	0.08	0.19	0.24	0.21	0.23	0.24	0.22	0.25	0.28	0.28

Table 8.2: Metrics Spearman correlation score against human judgment on perturbed texts. Here,  $f$ : fluency,  $s$ : semantic similarity,  $g$ : grammatical correctness.

Settings	Metrics	BAGEL			SFHOTEL		
		Info	Nat	Qual	Info	Nat	Qual
Baselines	BLEU-1	0.225	0.141	0.113	0.107	0.175	0.069
	BLEU-2	0.211	0.152	0.115	0.097	0.174	0.071
	METEOR	0.251	0.127	0.116	0.163	0.193	0.118
	BERTScore	0.267	0.210	0.178	0.163	0.193	0.118
	SMD+W2V	0.024	0.074	0.078	0.022	0.025	0.011
	SMD+ELMO+PMEANS	0.251	0.171	0.147	0.130	0.176	0.096
	SMD+BERT+MNLI+PMAENS	0.280	0.149	0.120	0.205	0.239	0.147
	WMD-1+ELMO+PMEANS	0.261	0.163	0.148	0.147	0.215	0.136
	WMD-1+BERT+PMEANS	0.298	0.212	0.163	0.203	0.261	0.182
	WMD-1+BERT+MNLI+PMEANS	0.285	0.195	0.158	0.207	0.270	0.183
RoMe	RoMe (Fasttext)	0.112	0.163	0.132	0.172	0.190	0.231
	RoMe (BERT)	0.160	0.251	0.202	0.212	0.283	0.300
	RoMe (ALBERT-base)	0.162	0.259	0.222	0.231	0.295	0.315
	<b>RoMe (ALBERT-large)</b>	0.170	0.274	0.241	0.244	0.320	0.327

Table 8.3: Spearman correlation ( $\rho$ ) scores computed from the metric scores with respect to the human evaluation scores on BAGEL and SFHOTEL. Baseline model’s results are reported from [113]. Here, **Info**, **Nat** and **Qual** refer to *informativeness*, *naturalness*, and *quality*, respectively.

	Text	EMD	TED-SE	Grammar	RoMe
$\mathcal{R}$	Munich is located at the southern part of Germany.	0.83	1.0	0.94	0.80
$\mathcal{H}$	Munich is situated in the south of Germany.				
$\mathcal{R}$	Tesla motors is founded by Elon Musk.	0.70	0.85	0.96	0.69
$\mathcal{H}$	Elon Musk has founded Tesla Motors.				
$\mathcal{R}$	Elon musk has founded tesla motors.	0.01	0.50	0.17	0.11
$\mathcal{H}$	Elon elon elon elon elon founded tesla tesla tesla.				

Table 8.4: Component-wise qualitative analysis.

Table 8.6 demonstrates the evaluation results on dialogue datasets. We evaluated the system-generated dialogues from three dialogue system models: Mem2Seq [91], GLMP [20], and DialoGPT [58]. In case of in-car dataset, all the non-word-overlap metric achieved a better correlation score than the word-overlap based metrics. This is because generated responses in dialogue systems are assessed based on the overall semantic meaning and correctness of the information. Overall, RoMe achieves stronger correlation scores on both in-car and soccer dialogue datasets in evaluating several dialogue system outputs. Finally, we investigate the outputs of nine distinct systems that competed in the WebNLG 2017 competition and report the correlation scores in Table 8.1. Although RoMe achieves the best correlation in most of the cases, we notice a comparable and in some cases better results achieved by the MoverScore [113].

A correlation graph is plotted in Figure 8.6 to investigate the metrics’ performance correlations further. The graph is constructed from RoMe and baseline metrics’ scores on the BAGEL dataset. As observed from the correlation graph, we can infer that our proposed metric, RoMe correlates highly with the MoverScore. However, since RoMe handles both the syntactic and semantic properties of the text it achieved better results in all the datasets across different NLG tasks.

	Text	BLEU	BERTScore	MoverScore	RoMe
$\mathcal{R}$	James Craig Watson, who died from peritonitis, discovered 101 Helena.	0.0	0.81	0.54	0.15
$\mathcal{H}$	The Polish Academy of Science is regionserved.				
$\mathcal{R}$	1001 gaussia was formerly known as 1923 oaa907 xc.	0.0	0.79	0.51	0.13
$\mathcal{H}$	The former name for the former name for 11 gunger is 1923. One of the former name is 1923.				

Table 8.5: Qualitative analysis.

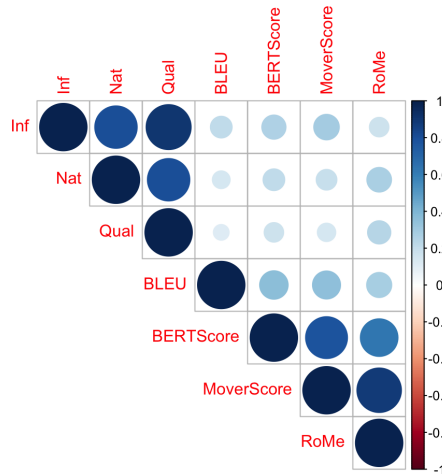


Figure 8.6: Correlation between the explored metrics.

Dialogue dataset	Models	SentBLEU	METEOR	BERTScore	MoverScore	RoMe
In-car dialogue	Mem2Seq	0.07	0.35	0.40	0.49	0.51
	GLMP	0.04	0.29	0.32	0.31	0.32
	DialoGPT	0.17	0.60	0.62	0.73	0.78
Soccer dialogue	Mem2Seq	0.03	0.08	0.08	0.11	0.11
	GLMP	0.02	0.08	0.03	0.12	0.14
	DialoGPT	0.04	0.26	0.31	0.39	0.43

Table 8.6: Metrics Spearman’s correlation coefficient ( $\rho$ ) with human judgment on dialogue datasets.

### 8.3.5 Ablation Study

We conduct an ablation study to investigate the impact of the RoMe’s components on its overall performance. Table 8.7 exhibits the incremental improvement in Spearman’s correlation coefficient, that each of the components brings to the metric. We randomly choose 100 system-generated dialogue utterances from the dialogue datasets, since they frequently contain sentences in passive form and repetitive words. The correlation of standard EMD with the human judgement is denoted as "RoMe with  $EMD_{std}$ ". Inclusion of semantic word alignment ( $EMD_{align}$ ) and soft-penalization ( $EMD_{soft}$ ) further improved the correlation score. The classifier was not used until this point in the ablation since there was just one score. Moreover, the correlation score improved significantly when the semantically enhanced TED and grammatical acceptability were introduced as features in addition to the EMD score to a neural classifier. We hypothesize that the inclusion of language features related to grammar and syntactic similarity helped the neural network achieve better performance.

Approaches	Correlation ( $\rho$ )
RoMe with $EMD_{std}$	64.8
+ $EMD_{align}$	66.0
+ $EMD_{soft}$	66.9
+ TED-SE	69.1
+ Grammar	70.1

Table 8.7: Ablation Study.

### 8.3.6 Qualitative Analysis

RoMe is developed in a modular fashion, so it may be used to generate scores for semantic similarity, syntactic similarity, and grammatical acceptability separately. Table 8.4 shows the component-wise score and the final score of RoMe on three example data points. In the first example, RoMe demonstrates its ability of capturing similar sentences by obtaining high score. The scores from several components in the second example demonstrate RoMe’s ability to handle passive form. The final example in Table 8.4 demonstrates that RoMe penalizes sentence with repetitive word.

Table 8.5 shows the performance of the three baselines and RoMe in handling erroneous cases. Although the first example contains a completely different hypothesis and the second case with repetitive hypothesis both BERTScore and MoverScore exhibit high score. On the contrary, BLEU score is unable to handle such scenarios. However, by obtaining low scores, RoMe demonstrates its ability to understand such cases better.

## 8.4 Robustness Analysis

In this section, we design five test cases to stress the models’ capabilities. For the analysis purpose, we randomly sample data points from KELM [295] (cases 1, 2, and 4) and BAGEL [298] (cases 3 and 5). The annotators annotate the sampled data points on the following criteria: *fluency*, *semantic correctness*, *grammatical correctness*.

**Case 1: Entity replacement.** We perform invariance test (INV) from [299] to check the metrics’ NER capability in assessing the text quality. In this approach, we replace the entities present in the text partially or fully with other entities in the dataset. For instance, "*The population of Germany*" gets transformed to "*The population of England*".

**Case 2: Adjective replacement.** Similar to the entity replacement, in this case we choose 100 data points from KELM that contain adjective in them. Then we replace the adjectives with a synonym and an antonym word to generate two sentences from a single data point. For instance, the adjective *different* is replaced with *unlike* and *same*. At the end of this process, we obtain 200 data points.

**Case 3: Random word replacement.** The words in different positions in the text are replaced by a generic token AAA following the adversarial text attack method from [301], in this case. For instance, the sentence, "*x is a cheap restaurant near y*" is transformed into "*x is a cheap restaurant*

AAA AAA". We select the greedy search method with the constraints on stop-words modification from the TextAttack tool. This approach generates repetitive words when two consecutive words are replaced.

**Case 4: Text transformation.** We leverage TextFooler [300] to replace two words in the texts by similar words, keeping the semantic meaning and grammar preserved.

**Case 5: Passive forms.** In this case, we randomly choose 200 data points from the KELM [295] dataset where the system generated responses are in passive form.

From the results of robustness analysis in Table 8.2, it is evident that almost all the metrics obtain very low correlation scores with respect to human judgment. Word-overlap based metrics such as BLEU and METEOR mostly suffer from it. Although RoMe achieves higher correlation scores in most of the cases, there are still scope for improvement in handling the fluency of the text better. Text perturbation techniques used to design the test cases often generate disfluent texts. In some cases, the texts' entities or subjects get replaced by words from out of the domain. From our observation, we hypothesize that handling keywords such as entities may lead to a better correlation score.

## 8.5 Summary

We have presented RoMe, an automatic and robust evaluation metric for evaluating a variety of NLG tasks. The key contributions of RoMe include 1) **EMD-based semantic similarity**, where *hard word alignment* and *soft-penalization* techniques are employed into the EMD for tackling repetitive words and passive form of the sentence, 2) **semantically enhanced TED** that computes the syntactic similarity based on the node-similarity of the parsed dependency trees, 3) **grammatical acceptability classifier**, which evaluates the text's grammatical quality, and 4) **robustness analysis**, which assesses the metric's capability of handling various form of the text. Both quantitative and qualitative analyses exhibit that RoMe highly correlates with human judgment. This answers the research question RQ4, "*How to effectively employ a pre-trained language model to improve the evaluation of single-reference based generative systems?*". An empirical assessment on benchmark datasets and the robustness analysis results exhibit that RoMe can handle various surface forms and generate an evaluation score, which highly correlates with human judgment. However, RoMe does not handle entity and relation of a knowledge graph separately, which we intend to tackle in our future work. RoMe is designed to function at the sentence level and can be used to evaluate English sentences in the current version of the implementation. We released the code and annotation tool publicly<sup>2</sup>.

---

<sup>2</sup><https://github.com/rashad101/RoMe>





## Conclusion and Future Directions

### 9.1 Conclusion

Language models have revolutionized the field of NLP in recent years. The primary research objective of this thesis is to improve knowledge-enhanced conversational systems leveraging language models. Through comprehensive evaluations, this thesis demonstrates that pre-trained language models can be utilized in various ways to improve conversational systems. For example, for incorporating structured knowledge into a language model (*DialoKG* in Chapter 4), developing unsupervised KGQA (*Tree-KGQA* in Chapter 5), learning knowledge graph facts in language models' parameters (*SGPT* in Chapter 6), developing a machine reading comprehension system (*Climate Bot* in Chapter 7), designing a metric for evaluating generative systems (*RoMe* in Chapter 8). Based on the type of knowledge (structured or unstructured), this thesis proposes four systems, one dataset, and one evaluation metric. Table 9.1 demonstrates a high-level overview of this thesis's contributions that correspond to the research questions (see Chapter 1.2). We provide a summary of the contributions of this thesis below:

Knowledge type	Models	Contribution type	Research question
Structured	<i>DialoKG</i>	System	<b>RQ1:</b> Does incorporating structural information into a language model improve knowledge graph-based dialogue generation?
	<i>Tree-KGQA</i>	System	<b>RQ2:</b> How effective are pre-trained language models for developing an unsupervised knowledge-graph-based question-answering system without training data?
	<i>SGPT</i>	System	<b>RQ3:</b> Can a generative language model embed a knowledge graph in its parameters and learn to construct SPARQL queries?
Unstructured	<i>RoMe</i>	Metric	<b>RQ4:</b> How to effectively employ a pre-trained language model to improve the evaluation of single-reference based generative systems?

Table 9.1: A high-level overview of the contributions correspond to the research questions.

**Contributions for RQ1:** We proposed *DialoKG*, a novel task-oriented dialogue system that learns to incorporate structured knowledge into a language model for dialogue generation. Particularly, *DialoKG* introduced a knowledge embedding technique and knowledge graph-weighted attention masking method to facilitate a GPT-2 model with the understanding of external knowledge for dialogue generation. An empirical study on three benchmark datasets demonstrates the effectiveness of *DialoKG* by achieving an improved performance over the state-of-the-art models.

**Contributions for RQ2:** We presented *Tree-KGQA*, an unsupervised question answering system leveraging pre-trained language models. *Tree-KGQA* can be used to perform question answering over knowledge graphs without labeled and training data. Specifically, several pre-trained language models were employed to perform KGQA sub-tasks (i.e., entity and relation linking). Besides, Tree-walking and Tree-disambiguation based traversal techniques were proposed to find the answer entity from the knowledge graph. A comprehensive evaluation of the benchmark datasets confirmed the effectiveness of *Tree-KGQA* over state-of-the-art supervised methods.

**Contributions for RQ3:** We introduced *SGPT*, a generative approach for SPARQL query generation from natural language questions. *SGPT* proposed a technique to embed semantic and syntactic features of a question and train a language model to learn knowledge graph facts in its parameters. Experimental results on the dataset from two different KG suggest significant performance gain of *SGPT* over the state-of-the-art methods.

**Contributions for RQ4:** The evaluation of generative systems is a difficult task because the metric must be robust enough to handle a wide range of surface forms of generated sentences. We designed and proposed a robust evaluation metric, RoMe, which considers the generated sentence’s semantic, syntactic, and grammatical acceptability for the evaluation. A comprehensive robustness analysis confirms the effectiveness of RoMe in evaluating diverse surface forms of various generative systems such as Dialogue, data-to-text, and text generation systems.

## 9.2 Future Directions

**Knowledge Injection into a Language Model for Dialogue Generation.** Recent research revealed that language models could be regarded as knowledge bases [305]. Large-scale pre-train language models contain a vast amount of language patterns. The knowledge that comes with the pre-trained weight can be exploited either by conditioning over the contextual embedding space or leveraging the pre-trained weights to initialize models for tackling downstream tasks. Empirical evaluation verifies that models such as BERT-large [45] exhibit high accuracy in capturing relational knowledge comparable to entity and relation linking based knowledge extraction systems [305].

To capture the relational knowledge better, recent researches focus on pre-training language models jointly with knowledge bases [306, 307, 308]. The intuition behind this approach is that the trained language model may better understand the facts from relational knowledge bases. In a different line of work, several researches aim at pre-training language models on dialogue corpus. In the future, we would like to explore techniques that have the capability of pre-training a language model that considers both knowledge graphs and dialogue data. Developing such a dataset would be a prerequisite for designing such techniques.

**Prompt-based Approaches for Unsupervised Question Answering.** Language models are widely adopted for various NLP downstream tasks for their capabilities of understanding a wide range of text patterns [207, 309]. These models are trained on a large corpus of unlabelled text. Pre-trained language models are typically fine-tuned to apply them for downstream tasks. However, fine-tuning requires labeled data. Obtaining labeled training data for conversational question answering systems is resource-intensive and time-consuming.

Pre-trained language model comes with a knowledge of a large number of textual patterns. Recently, prompt-based learning has become a popular choice for utilizing large pre-trained language models for applying on the downstream task in a few-shot or zero-shot manner [310, 311]. Although few approaches have recently developed prompts for tackling question answering and dialogue generation tasks, a deep investigation into this direction is still required for the maximum utilization of pre-trained language models. In the future, we would like to investigate prompt-based approaches to alleviate the need for training data to develop high-performing conversational question answering agents.

**Evaluation Metrics for Knowledge-based Generative Systems.** With the increasing number of knowledge graphs in recent years, a large number of knowledge-based systems have been developed over the past few years. Performance assessment is a key factor in developing any intelligent system. To evaluate generative conversational systems, researchers still utilize metrics such as BLEU, METEOR, and ROUGE. However, these metrics were borrowed from the machine translation and summarization community. Existing metrics are not sufficient for evaluating knowledge-based generative systems. Because for a generated word  $w$ , these metrics cannot capture the relation between facts in the knowledge graph connected to  $w$ . For evaluating knowledge-based generative systems, besides the contextual meaning of a word  $w$  in the generated systems, it is crucial that the metric also considers the connection information of  $w$ . However, it is challenging to develop such a metric because of the scale of KG and the complexities (entity and relation linking) required for the evaluation. Developing such a metric would not only help evaluate generative conversational systems but also in evaluating any knowledge-based generative systems.



---

## List of Publications

---

A list of accepted conference and journal papers, contributing to this thesis:

### Conference Papers (peer reviewed)

- **Md Rashad Al Hasan Rony**, Ricardo Usbeck, and Jens Lehmann. 2022. *DialoKG: Knowledge-Structure Aware Task-Oriented Dialogue Generation*. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 2557–2571, Seattle, United States. Association for Computational Linguistics.
- **Md Rashad Al Hasan Rony**, Liubov Kovriguina, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann. 2022. *RoMe: A Robust Metric for Evaluating Natural Language Generation*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5645–5657, Dublin, Ireland. Association for Computational Linguistics.

### System Demo Papers (peer reviewed)

- **Md Rashad Al Hasan Rony**, Ying Zuo, Liubov Kovriguina, Roman Teucher and Jens Lehmann, *Climate Bot: A Machine Reading Comprehension System for Climate Change Question Answering*. In Proceedings of IJCAI 2022, in AI for good track.

### Journal Papers (peer reviewed)

- **Md Rashad Al Hasan Rony**, Debanjan Chaudhuri, Ricardo Usbeck, and Jens Lehmann, *Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs*, in IEEE Access, vol. 10, pp. 50467-50478, 2022, doi: 10.1109/ACCESS.2022.3173355.
- **Md Rashad Al Hasan Rony**, Uttam Kumar, Roman Teucher, Liubov Kovriguina and Jens Lehmann, *SGPT: A Generative Approach for SPARQL Query Generation from Natural Language Questions*, in IEEE Access, vol. 10, pp. 70712-70723, 2022, doi: 10.1109/ACCESS.2022.3188714.



# Bibliography

---

- [1] J. Weizenbaum, *ELIZA—a computer program for the study of natural language communication between man and machine*, *Communications of the ACM* **9** (1966) 36 (cit. on pp. 1, 32).
- [2] K. M. Colby, S. Weber, and F. D. Hilf, *Artificial paranoia*, *Artificial Intelligence* **2** (1971) 1 (cit. on pp. 1, 32).
- [3] K. M. Colby, F. D. Hilf, S. Weber, and H. C. Kraemer, *Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes*, *Artificial Intelligence* **3** (1972) 199 (cit. on p. 1).
- [4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., *Language models are few-shot learners*, *Advances in neural information processing systems* **33** (2020) 1877 (cit. on p. 1).
- [5] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, “DialogPT: Large-Scale Generative Pre-training for Conversational Response Generation,” *ACL, system demonstration*, 2020 (cit. on pp. 1, 2, 32).
- [6] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., *Palm: Scaling language modeling with pathways*, arXiv preprint arXiv:2204.02311 (2022) (cit. on p. 1).
- [7] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al., *Opt: Open pre-trained transformer language models*, arXiv preprint arXiv:2205.01068 (2022) (cit. on pp. 1, 13).
- [8] A. Madotto, S. Cahyawijaya, G. I. Winata, Y. Xu, Z. Liu, Z. Lin, and P. Fung, “Learning Knowledge Bases with Parameters for Task-Oriented Dialogue Systems,” *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, 2020 2372, URL: <https://aclanthology.org/2020.findings-emnlp.215> (cit. on pp. 2, 19, 32, 47–49).
- [9] M. R. A. H. Rony, R. Usbeck, and J. Lehmann, “DialogKG: Knowledge-Structure Aware Task-Oriented Dialogue Generation,” *Findings of the Association for Computational Linguistics: NAACL 2022*, Association for Computational Linguistics, 2022 2557, URL: <https://aclanthology.org/2022.findings-naacl.195> (cit. on pp. 2, 25).

- [10] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, "A simple language model for task-oriented dialogue," *Advances in Neural Information Processing Systems* **33** (2020) 20179 (cit. on pp. 2, 31).
- [11] D. Chaudhuri, M. R. A. H. Rony, S. Jordan, and J. Lehmann, "Using a KG-copy network for non-goal oriented dialogues," *International Semantic Web Conference*, Springer, 2019 93 (cit. on pp. 2, 20, 31, 103).
- [12] D. Chaudhuri, M. R. A. H. Rony, and J. Lehmann, "Grounding Dialogue Systems via Knowledge Graph Aware Decoding with Pre-trained Transformers," *European Semantic Web Conference*, Springer, 2021 323 (cit. on pp. 2, 31, 32, 98, 104).
- [13] A. Madotto, C.-S. Wu, and P. Fung, "Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2018 1468, URL: <http://aclweb.org/anthology/P18-1136> (cit. on p. 2).
- [14] M. Dubey, D. Banerjee, D. Chaudhuri, and J. Lehmann, "EARL: joint entity and relation linking for question answering over knowledge graphs," *International Semantic Web Conference*, Springer, 2018 108 (cit. on pp. 2, 21, 34, 65, 66, 77).
- [15] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017 1870, URL: <https://aclanthology.org/P17-1171> (cit. on pp. 2, 25, 36).
- [16] D. Vrandečić, "Wikidata: A new platform for collaborative data collection," *Proceedings of the 21st international conference on world wide web*, 2012 1063 (cit. on pp. 2, 17, 18, 21, 35, 55, 56, 64, 73, 74).
- [17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," *The semantic web*, Springer, 2007 722 (cit. on pp. 2, 17, 21).
- [18] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008 1247 (cit. on pp. 2, 17, 55).
- [19] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, 2018 784, URL: <https://aclanthology.org/P18-2124> (cit. on pp. 2, 37, 91, 93).
- [20] C.-S. Wu, R. Socher, and C. Xiong, "Global-to-local Memory Pointer Networks for Task-Oriented Dialogue," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019 (cit. on pp. 2, 20, 31, 47, 49, 106).



- 
- [21] D. Banerjee, P. A. Nair, J. N. Kaur, R. Usbeck, and C. Biemann, “Modern Baselines for SPARQL Semantic Parsing,” *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22*, Association for Computing Machinery, 2022 2260, ISBN: 9781450387323, URL: <https://doi.org/10.1145/3477495.3531841> (cit. on pp. 2, 78, 81, 84).
- [22] D. Sorokin and I. Gurevych, “Modeling Semantics with Gated Graph Neural Networks for Knowledge Base Question Answering,” *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, 2018 3306, URL: <https://www.aclweb.org/anthology/C18-1280> (cit. on pp. 2, 34, 57, 64, 65, 67).
- [23] L. Luo, J. Xu, J. Lin, Q. Zeng, and X. Sun, “An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 702, URL: <https://aclanthology.org/D18-1075> (cit. on pp. 2, 20, 32).
- [24] A. Kulshreshtha, D. D. F. Adiwardana, D. R. So, G. Nemade, J. Hall, N. Fiedel, Q. V. Le, R. Thoppilan, T. Luong, Y. Lu, et al., *Towards a Human-like Open-Domain Chatbot*, (2020) (cit. on pp. 2, 20, 32).
- [25] N. Dziri, A. Madotto, O. Zaiane, and A. J. Bose, “Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding,” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021 2197, URL: <https://aclanthology.org/2021.emnlp-main.168> (cit. on pp. 2, 20, 32).
- [26] Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li, “Sequential Matching Network: A New Architecture for Multi-turn Response Selection in Retrieval-Based Chatbots,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017 496, URL: <https://aclanthology.org/P17-1046> (cit. on pp. 2, 20, 32).
- [27] C. Tao, W. Wu, C. Xu, W. Hu, D. Zhao, and R. Yan, “One Time of Interaction May Not Be Enough: Go Deep with an Interaction-over-Interaction Network for Response Selection in Dialogues,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 1, URL: <https://aclanthology.org/P19-1001> (cit. on pp. 2, 20, 32).
- [28] T. Whang, D. Lee, D. Oh, C. Lee, K. Han, D.-h. Lee, and S. Lee, “Do response selection models really know what’s next? utterance manipulation strategies for multi-turn response selection,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 16, 2021 14041 (cit. on pp. 2, 20).

- [29] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-to-End Memory Networks,” *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS’15*, MIT Press, 2015 2440 (cit. on pp. 2, 20).
- [30] J. Gu, Z. Lu, H. Li, and V. O. Li, “Incorporating Copying Mechanism in Sequence-to-Sequence Learning,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016 1631, URL: <https://aclanthology.org/P16-1154> (cit. on pp. 2, 20).
- [31] X. Lin, W. Jian, J. He, T. Wang, and W. Chu, “Generating Informative Conversational Response using Recurrent Knowledge-Interaction and Knowledge-Copy,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 41, URL: <https://aclanthology.org/2020.acl-main.6> (cit. on pp. 2, 20, 32).
- [32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002 311 (cit. on pp. 3, 25, 38, 47, 75, 83, 94, 98, 105).
- [33] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005 65 (cit. on pp. 3, 25, 26, 38, 94, 98, 105).
- [34] T. Zhang\*, V. Kishore\*, F. Wu\*, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” *International Conference on Learning Representations*, 2020, URL: <https://openreview.net/forum?id=SkeHuCVFDr> (cit. on pp. 3, 25, 29).
- [35] H. Echizen-ya, K. Araki, and E. Hovy, “Word Embedding-Based Automatic MT Evaluation Metric using Word Position Information,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019 1874 (cit. on pp. 3, 30, 38, 100, 101).
- [36] T. Sellam, D. Das, and A. P. Parikh, “BLEURT: Learning Robust Metrics for Text Generation,” *Proceedings of ACL*, 2020 (cit. on pp. 3, 30).
- [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, 2017 5998 (cit. on pp. 6–8, 11, 14, 15, 33, 38, 39, 45, 75, 76, 78).
- [38] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language Models are Unsupervised Multitask Learners*, (2019) (cit. on pp. 7, 8, 11, 14, 15, 31, 39, 42, 45, 46, 75, 77, 78, 81).
- [39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805 (2018) (cit. on pp. 8, 11, 13, 33, 56, 58, 64).

- 
- [40] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 7871, URL: <https://www.aclweb.org/anthology/2020.acl-main.703> (cit. on pp. 8, 11, 14, 33, 56, 57, 61, 64, 69, 84).
- [41] Y. Rubner, C. Tomasi, and L. J. Guibas, “A metric for distributions with applications to image databases,” *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, IEEE, 1998 59 (cit. on pp. 9, 26, 98, 99).
- [42] X. Zhang, J. Zhao, and Y. LeCun, *Character-level convolutional networks for text classification*, *Advances in neural information processing systems* **28** (2015) (cit. on p. 11).
- [43] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, *Enriching Word Vectors with Subword Information*, *Transactions of the Association for Computational Linguistics* **5** (2017) 135, ISSN: 2307-387X (cit. on pp. 11, 104, 105).
- [44] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep Contextualized Word Representations,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, 2018 2227, URL: <https://aclanthology.org/N18-1202> (cit. on p. 11).
- [45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019 4171, URL: <https://www.aclweb.org/anthology/N19-1423> (cit. on pp. 11, 14, 15, 32, 33, 36, 76, 102, 105, 112).
- [46] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, *Advances in neural information processing systems* **32** (2019) (cit. on p. 11).
- [47] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2019, URL: <https://arxiv.org/abs/1908.10084> (cit. on pp. 11, 15, 59, 90, 91).

- [48] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations,” *International Conference on Learning Representations*, 2020, URL: <https://openreview.net/forum?id=H1eA7AetvS> (cit. on pp. 11, 15, 91, 100, 104, 105).
- [49] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013) (cit. on p. 11).
- [50] J. Pennington, R. Socher, and C. Manning, “GloVe: Global Vectors for Word Representation,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014 1532, URL: <https://aclanthology.org/D14-1162> (cit. on pp. 11, 12).
- [51] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” *International conference on machine learning*, PMLR, 2014 1188 (cit. on pp. 11, 12).
- [52] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of Tricks for Efficient Text Classification,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Association for Computational Linguistics, 2017 427, URL: <https://aclanthology.org/E17-2068> (cit. on p. 12).
- [53] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, *Neural Computation* **9** (1997) 1735 (cit. on p. 12).
- [54] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, “Object recognition with gradient-based learning,” *Shape, contour and grouping in computer vision*, Springer, 1999 319 (cit. on p. 13).
- [55] Y. Luan and S. Lin, “Research on Text Classification Based on CNN and LSTM,” *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019 352 (cit. on p. 13).
- [56] Y. Zhou, J. Li, J. Chi, W. Tang, and Y. Zheng, *Set-CNN: A text convolutional neural network based on semantic extension for short text classification*, *Knowledge-Based Systems* **257** (2022) 109948 (cit. on p. 13).
- [57] D. Bahdanau, K. Cho, and Y. Bengio, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473 (2014) (cit. on p. 13).
- [58] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, “DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, 2020 270, URL: <https://www.aclweb.org/anthology/2020.acl-demos.30> (cit. on pp. 13, 32, 106).

- 
- [59] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, “Conditional Prompt Learning for Vision-Language Models,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 16816 (cit. on p. 13).
- [60] R. Kiros, R. Salakhutdinov, and R. Zemel, “Multimodal neural language models,” *International conference on machine learning*, PMLR, 2014 595 (cit. on p. 13).
- [61] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer,” *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019 1441 (cit. on p. 13).
- [62] Y. Ma, B. Narayanaswamy, H. Lin, and H. Ding, “Temporal-contextual recommendation in real-time,” *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020 2291 (cit. on p. 13).
- [63] N. Choudhary, N. Rao, K. Subbian, and C. K. Reddy, *Graph-based multilingual language model: Leveraging product relations for search relevance*, (2022) (cit. on p. 13).
- [64] R. Biswas, R. Sofronova, M. Alam, and H. Sack, “Contextual Language Models for Knowledge Graph Completion.,” *MLSMKG@ PKDD/ECML*, 2021 (cit. on p. 13).
- [65] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” *International conference on machine learning*, PMLR, 2015 2048 (cit. on p. 13).
- [66] G. Lembersky, N. Ordan, and S. Wintner, *Language Models for Machine Translation: Original vs. Translated Texts*, *Computational Linguistics* **38** (2012) 799, URL: <https://aclanthology.org/J12-4004> (cit. on p. 17).
- [67] B. Zhang, B. Ghorbani, A. Bapna, Y. Cheng, X. Garcia, J. Shen, and O. Firat, “Examining Scaling and Transfer of Language Model Architectures for Machine Translation,” *Proceedings of the 39th International Conference on Machine Learning*, ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, vol. 162, *Proceedings of Machine Learning Research*, PMLR, 2022 26176, URL: <https://proceedings.mlr.press/v162/zhang22h.html> (cit. on p. 17).
- [68] A. Chronopoulou, D. Stojanovski, and A. Fraser, “Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020 2703, URL: <https://www.aclweb.org/anthology/2020.emnlp-main.214> (cit. on p. 17).
- [69] D. Aksenov, J. Moreno-Schneider, P. Bourgonje, R. Schwarzenberg, L. Hennig, and G. Rehm, “Abstractive Text Summarization based on Language Model Conditioning and Locality Modeling,” English, *Proceedings of the Twelfth Language Resources and Evaluation Conference*,

- European Language Resources Association, 2020 6680, ISBN: 979-10-95546-34-4,  
URL: <https://aclanthology.org/2020.lrec-1.825> (cit. on p. 17).
- [70] Y. Zhang, P. Nie, A. Ramamurthy, and L. Song,  
*Ddrqa: Dynamic document reranking for open-domain multi-hop question answering*,  
arXiv e-prints (2020) arXiv (cit. on pp. 17, 37).
- [71] P. Qi, X. Lin, L. Mehr, Z. Wang, and C. D. Manning,  
“Answering Complex Open-domain Questions Through Iterative Query Generation,”  
*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing  
and the 9th International Joint Conference on Natural Language Processing  
(EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 2590,  
URL: <https://aclanthology.org/D19-1261> (cit. on pp. 17, 37).
- [72] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao,  
“Coupled multi-layer attentions for co-extraction of aspect and opinion terms,”  
*Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 1, 2017 (cit. on p. 17).
- [73] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification,”  
*Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics  
(Volume 1: Long Papers)*, Association for Computational Linguistics, 2018 328,  
URL: <https://aclanthology.org/P18-1031> (cit. on p. 17).
- [74] Y. Arslan, K. Allix, L. Veiber, C. Lothritz, T. F. Bissyandé, J. Klein, and A. Goujon,  
“A comparison of pre-trained language models for multi-class text classification in the  
financial domain,” *Companion Proceedings of the Web Conference 2021*, 2021 260  
(cit. on p. 17).
- [75] M. Kejriwal, *Knowledge Graphs: A Practical Review of the Research Landscape*,  
*Information* **13** (2022) 161 (cit. on p. 17).
- [76] P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, V. Denny, P. Groth, N. Noy,  
K. Janowicz, C. Goble, and et al., *The Semantic Web - ISWC 2014 13th International  
Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*,  
Springer International Publishing, 2014 (cit. on p. 17).
- [77] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,”  
*Proceedings of the 16th international conference on World Wide Web*, 2007 697  
(cit. on p. 17).
- [78] G. Blog, *Introducing the knowledge graph: thing, not strings*,  
*Introducing the Knowledge Graph: things, not strings* (2012) (cit. on p. 17).
- [79] L. Ehrlinger and W. Wöb, *Towards a definition of knowledge graphs.*,  
*SEMANTiCS (Posters, Demos, SuCCESS)* **48** (2016) 2 (cit. on p. 17).
- [80] P. A. Bonatti, S. Decker, A. Polleres, and V. Presutti, “Knowledge graphs: New directions for  
knowledge representation on the semantic web (dagstuhl seminar 18371),” *Dagstuhl reports*,  
vol. 8, 9, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019 (cit. on p. 17).
- [81] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger,  
*Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago*,  
*Semantic Web* **9** (2018) 77 (cit. on p. 17).

- 
- [82] H. Paulheim, *Knowledge graph refinement: A survey of approaches and evaluation methods*, *Semantic web* **8** (2017) 489 (cit. on pp. 17, 18).
- [83] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann, “Message Passing for Hyper-Relational Knowledge Graphs,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020 7346, URL: <https://aclanthology.org/2020.emnlp-main.596> (cit. on p. 18).
- [84] D. Alivanistos, M. Berrendorf, M. Cochez, and M. Galkin, “Query Embedding on Hyper-Relational Knowledge Graphs,” *International Conference on Learning Representations*, 2022, URL: <https://openreview.net/forum?id=4rLw09TgRw9> (cit. on p. 18).
- [85] S. Young, M. Gašić, B. Thomson, and J. D. Williams, *Pomdp-based statistical spoken dialog systems: A review*, *Proceedings of the IEEE* **101** (2013) 1160 (cit. on p. 19).
- [86] S. Young, “Using POMDPs for dialog management,” *2006 IEEE Spoken Language Technology Workshop*, IEEE, 2006 8 (cit. on p. 19).
- [87] N. Mrkšić, D. Ó Séaghdha, T.-H. Wen, B. Thomson, and S. Young, “Neural Belief Tracker: Data-Driven Dialogue State Tracking,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017 1777, URL: <https://aclanthology.org/P17-1163> (cit. on pp. 19, 31).
- [88] S. Gao, A. Sethi, S. Agarwal, T. Chung, and D. Hakkani-Tur, “Dialog State Tracking: A Neural Reading Comprehension Approach,” *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, Association for Computational Linguistics, 2019 264, URL: <https://aclanthology.org/W19-5932> (cit. on p. 19).
- [89] Z. Zhang, L. Liao, X. Zhu, T.-S. Chua, Z. Liu, Y. Huang, and M. Huang, “Learning Goal-oriented Dialogue Policy with opposite Agent Awareness,” *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, 2020 122, URL: <https://aclanthology.org/2020.aacl-main.16> (cit. on p. 19).
- [90] Z. Lipton, X. Li, J. Gao, L. Li, F. Ahmed, and L. Deng, “Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 1, 2018 (cit. on p. 19).
- [91] A. Madotto, C.-S. Wu, and P. Fung, “Mem2Seq: Effectively Incorporating Knowledge Bases into End-to-End Task-Oriented Dialog Systems,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2018 1468, URL: <https://www.aclweb.org/anthology/P18-1136> (cit. on pp. 20, 106).
- [92] Y. Song, R. Yan, C.-T. Li, J.-Y. Nie, M. Zhang, and D. Zhao, *An Ensemble of Retrieval-Based and Generation-Based Human-Computer Conversation Systems.*, (2018) (cit. on p. 20).

- [93] Q. Zhu, L. Cui, W.-N. Zhang, F. Wei, and T. Liu, “Retrieval-Enhanced Adversarial Training for Neural Response Generation,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 3763, URL: <https://aclanthology.org/P19-1366> (cit. on p. 20).
- [94] M. R. A. H. Rony, D. Chaudhuri, R. Nedelchev, A. Fischer, and J. Lehmann, *End-to-End Entity Linking and Disambiguation leveraging Word and Knowledge Graph Embeddings*, () (cit. on p. 21).
- [95] A. Tonon, G. Demartini, and P. Cudré-Mauroux, “Combining inverted indices and structured search for ad-hoc object retrieval,” *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 2012 125 (cit. on p. 22).
- [96] N. Zhiltsov, A. Kotov, and F. Nikolaev, “Fielded sequential dependence model for ad-hoc entity retrieval in the web of data,” *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015 253 (cit. on p. 22).
- [97] H. Zafar, G. Napolitano, and J. Lehmann, “Formal query generation for question answering over knowledge bases,” *European semantic web conference*, Springer, 2018 714 (cit. on pp. 22, 34, 35, 74, 81, 83, 84).
- [98] D. Vollmers, R. Jalota, D. Moussallem, H. Topiwala, A.-C. N. Ngomo, and R. Usbeck, *Knowledge Graph Question Answering using Graph-Pattern Isomorphism*, arXiv preprint arXiv:2103.06752 (2021) (cit. on pp. 22, 35, 74, 81, 84).
- [99] D. Chen, *Neural reading comprehension and beyond*, Stanford University, 2018 (cit. on p. 22).
- [100] K. S. Jones, *A statistical interpretation of term specificity and its application in retrieval*, *Journal of documentation* (1972) (cit. on pp. 23, 36).
- [101] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al., *Okapi at TREC-3*, Nist Special Publication Sp **109** (1995) 109 (cit. on pp. 23, 36).
- [102] V. Karpukhin, B. Oguz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, “Dense Passage Retrieval for Open-Domain Question Answering,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by B. Webber, T. Cohn, Y. He, and Y. Liu, Association for Computational Linguistics, 2020 6769, URL: <https://doi.org/10.18653/v1/2020.emnlp-main.550> (cit. on pp. 23, 36, 37, 90, 91).
- [103] G. Izacard and E. Grave, “Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering,” *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Association for Computational Linguistics, 2021 874, URL: <https://aclanthology.org/2021.eacl-main.74> (cit. on p. 24).



- 
- [104] N. Arabzadeh, X. Yan, and C. L. Clarke, “Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection,” *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021 2862 (cit. on p. 24).
- [105] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, “Retrieval augmented language model pre-training,” *International Conference on Machine Learning*, PMLR, 2020 3929 (cit. on p. 25).
- [106] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin, “End-to-End Open-Domain Question Answering with BERTserini,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Association for Computational Linguistics, 2019 72, URL: <https://aclanthology.org/N19-4013> (cit. on pp. 25, 36).
- [107] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*, 2019, arXiv: 1910.13461 [cs.CL] (cit. on pp. 25, 74).
- [108] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., *Exploring the limits of transfer learning with a unified text-to-text transformer.*, *J. Mach. Learn. Res.* **21** (2020) 1 (cit. on pp. 25, 84).
- [109] S. Liu, X. Zhang, S. Zhang, H. Wang, and W. Zhang, *Neural machine reading comprehension: Methods and trends*, *Applied Sciences* **9** (2019) 3698 (cit. on pp. 25, 37).
- [110] M. R. A. H. Rony, D. Chaudhuri, R. Usbeck, and J. Lehmann, *Tree-KGQA: An Unsupervised Approach for Question Answering Over Knowledge Graphs*, *IEEE Access* (2022) (cit. on p. 25).
- [111] M. R. A. H. Rony, Y. Zuo, L. Kovriguina, R. Teucher, and J. Lehmann, “Climate Bot: A Machine Reading Comprehension System for Climate Change Question Answering,” *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, ed. by L. D. Raedt, AI for Good - Demos, International Joint Conferences on Artificial Intelligence Organization, 2022 5249, URL: <https://doi.org/10.24963/ijcai.2022/729> (cit. on pp. 25, 37).
- [112] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” *Text Summarization Branches Out*, Association for Computational Linguistics, 2004 74, URL: <https://www.aclweb.org/anthology/W04-1013> (cit. on pp. 25, 38, 98).
- [113] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, “MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 563, URL: <https://aclanthology.org/D19-1053> (cit. on pp. 25–27, 38, 47, 98, 101, 105, 106).

- [114] R. Nedelchev, J. Lehmann, and R. Usbeck, “Language model transformers as evaluators for open-domain dialogues,” *Proceedings of the 28th International Conference on Computational Linguistics*, 2020 6797 (cit. on p. 26).
- [115] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, “From word embeddings to document distances,” *International conference on machine learning*, 2015 957 (cit. on pp. 26, 27, 38, 98).
- [116] K. Zhang and D. Shasha, *Simple fast algorithms for the editing distance between trees and related problems*, *SIAM journal on computing* **18** (1989) 1245 (cit. on pp. 27, 101).
- [117] P. N. Klein, “Computing the Edit-Distance Between Unrooted Ordered Trees,” *Algorithms — ESA’ 98*, ed. by G. Bilardi, G. F. Italiano, A. Pietracaprina, and G. Pucci, Springer Berlin Heidelberg, 1998 91, ISBN: 978-3-540-68530-2 (cit. on p. 29).
- [118] D. Shasha and K. Zhang, *Fast algorithms for the unit cost editing distance between trees*, *Journal of algorithms* **11** (1990) 581 (cit. on p. 29).
- [119] K. Zhang, “Efficient parallel algorithms for tree editing problems,” *Annual Symposium on Combinatorial Pattern Matching*, Springer, 1996 361 (cit. on p. 29).
- [120] C.-S. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung, “Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 808, URL: <https://aclanthology.org/P19-1078> (cit. on p. 31).
- [121] S. Chen and S. Yu, “Wais: Word attention for joint intent detection and slot filling,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 01, 2019 9927 (cit. on p. 31).
- [122] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young, “Latent intention dialogue models,” *International Conference on Machine Learning*, PMLR, 2017 3732 (cit. on p. 31).
- [123] T.-H. Wen, M. Gašić, N. Mrkšić, P.-H. Su, D. Vandyke, and S. Young, “Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems,” *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015 1711, URL: <https://www.aclweb.org/anthology/D15-1199> (cit. on pp. 31, 98, 103).
- [124] L. Shu, P. Molino, M. Namazifar, B. Liu, H. Xu, H. Zheng, and G. Tur, “Incorporating the structure of the belief state in end-to-end task-oriented dialogue systems,” *2nd Workshop on Conversational AI at Neural Information Processing Systems*, vol. 32, 2018 (cit. on p. 31).
- [125] B. Liu, G. Tür, D. Hakkani-Tür, P. Shah, and L. Heck, “Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems,” *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, 2018 2060, URL: <https://aclanthology.org/N18-1187> (cit. on p. 31).

- 
- [126] Y. Zhang, Z. Ou, and Z. Yu, “Task-oriented dialog systems that consider multiple appropriate responses under the same context,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 05, 2020 9604 (cit. on p. 31).
- [127] A. Neelakantan, S. Yavuz, S. Narang, V. Prasad, B. Goodrich, D. Duckworth, C. Sankar, and X. Yan, *Neural Assistant: Joint Action Prediction, Response Generation, and Latent Knowledge Reasoning*, (2019) (cit. on p. 31).
- [128] T. Zhao, K. Xie, and M. Eskenazi, “Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019 1208, URL: <https://aclanthology.org/N19-1123> (cit. on p. 31).
- [129] A. Bordes, Y.-L. Boureau, and J. Weston, “Learning End-to-End Goal-Oriented Dialog,” *International Conference on Learning Representations*, 2017, URL: <https://openreview.net/forum?id=S1Bb3D5gg> (cit. on p. 31).
- [130] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher, *Ctrl: A conditional transformer language model for controllable generation*, arXiv preprint arXiv:1909.05858 (2019) (cit. on p. 31).
- [131] S. Bao, H. He, F. Wang, H. Wu, and H. Wang, “PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 85, URL: <https://aclanthology.org/2020.acl-main.9> (cit. on p. 32).
- [132] H. Wang, Z. Lu, H. Li, and E. Chen, “A Dataset for Research on Short-Text Conversations,” *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2013 935, URL: <https://aclanthology.org/D13-1096> (cit. on p. 32).
- [133] Z. Ji, Z. Lu, and H. Li, *An information retrieval approach to short text conversation*, arXiv preprint arXiv:1408.6988 (2014) (cit. on p. 32).
- [134] X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, and R. Yan, “Multi-view Response Selection for Human-Computer Conversation,” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2016 372, URL: <https://aclanthology.org/D16-1036> (cit. on p. 32).
- [135] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan, “A Persona-Based Neural Conversation Model,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016 994, URL: <https://aclanthology.org/P16-1094> (cit. on p. 32).

- [136] J.-C. Gu, Z.-H. Ling, X. Zhu, and Q. Liu, “Dually Interactive Matching Network for Personalized Response Selection in Retrieval-Based Chatbots,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 1845, URL: <https://aclanthology.org/D19-1193> (cit. on p. 32).
- [137] Z. Lin, D. Cai, Y. Wang, X. Liu, H. Zheng, and S. Shi, “The World is Not Binary: Learning to Rank with Grayscale Data for Dialogue Response Selection,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020 9220, URL: <https://aclanthology.org/2020.emnlp-main.741> (cit. on p. 32).
- [138] L. Shang, Z. Lu, and H. Li, “Neural Responding Machine for Short-Text Conversation,” *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2015 1577, URL: <https://aclanthology.org/P15-1152> (cit. on p. 32).
- [139] O. Vinyals and Q. Le, *A neural conversational model*, arXiv preprint arXiv:1506.05869 (2015) (cit. on p. 32).
- [140] P. Parthasarathi and J. Pineau, “Extending Neural Generative Conversational Model using External Knowledge Sources,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 690, URL: <https://aclanthology.org/D18-1073> (cit. on p. 32).
- [141] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, “Toward an architecture for never-ending language learning,” *Twenty-Fourth AAAI conference on artificial intelligence*, 2010 (cit. on p. 32).
- [142] H. Zhang, Y. Lan, L. Pang, H. Chen, Z. Ding, and D. Yin, “Modeling Topical Relevance for Multi-Turn Dialogue Generation,” *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, ed. by C. Bessiere, Main track, International Joint Conferences on Artificial Intelligence Organization, 2020 3737, URL: <https://doi.org/10.24963/ijcai.2020/517> (cit. on p. 32).
- [143] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley, “A knowledge-grounded neural conversation model,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 1, 2018 (cit. on p. 32).
- [144] Y. Zheng, Z. Chen, R. Zhang, S. Huang, X. Mao, and M. Huang, “Stylized dialogue response generation using stylized unpaired texts,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 16, 2021 14558 (cit. on p. 32).
- [145] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, *The design and implementation of xiaoice, an empathetic social chatbot*, *Computational Linguistics* **46** (2020) 53 (cit. on p. 32).

- 
- [146] Z. Lin, A. Madotto, Y. Bang, and P. Fung, “The Adapter-Bot: All-In-One Controllable Conversational Model,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 18, 2021 16081 (cit. on p. 32).
- [147] K. Shuster, E. M. Smith, D. Ju, and J. Weston, “Multi-Modal Open-Domain Dialogue,” *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021 4863, URL: <https://aclanthology.org/2021.emnlp-main.398> (cit. on p. 32).
- [148] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, “Zero-Shot Entity Linking by Reading Entity Descriptions,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019 (cit. on p. 33).
- [149] A. Sakor, I. O. Mulang, K. Singh, S. Shekarpour, M. E. Vidal, J. Lehmann, and S. Auer, “Old is gold: linguistic driven approach for entity and relation linking of short text,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019 2336 (cit. on pp. 33, 34, 65, 66).
- [150] B. Z. Li, S. Min, S. Iyer, Y. Mehdad, and W.-t. Yih, “Efficient One-Pass End-to-End Entity Linking for Questions,” *EMNLP*, 2020 (cit. on pp. 33, 56, 59).
- [151] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, “Scalable Zero-shot Entity Linking with Dense Entity Retrieval,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by B. Webber, T. Cohn, Y. He, and Y. Liu, Association for Computational Linguistics, 2020 6397, URL: <https://doi.org/10.18653/v1/2020.emnlp-main.519> (cit. on pp. 33, 56, 59).
- [152] W. Yu, L. Wu, Y. Deng, R. Mahindru, Q. Zeng, S. Guven, and M. Jiang, “A Technical Question Answering System with Transfer Learning,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, 2020 92, URL: <https://www.aclweb.org/anthology/2020.emnlp-demos.13> (cit. on p. 33).
- [153] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, “Scalable Zero-shot Entity Linking with Dense Entity Retrieval,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, ed. by B. Webber, T. Cohn, Y. He, and Y. Liu, Association for Computational Linguistics, 2020 6397, URL: <https://doi.org/10.18653/v1/2020.emnlp-main.519> (cit. on p. 33).

- [154] E. Boros, E. L. Pontes, L. A. Cabrera-Diego, A. Hamdi, J. G. Moreno, N. Sidère, and A. Doucet, “Robust named entity recognition and linking on historical multilingual documents,” *Conference and Labs of the Evaluation Forum (CLEF 2020)*, vol. 2696, Paper 171, CEUR-WS Working Notes, 2020 1 (cit. on p. 33).
- [155] N. Kolitsas, O.-E. Ganea, and T. Hofmann, “End-to-End Neural Entity Linking,” *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2018 519, URL: <https://aclanthology.org/K18-1050> (cit. on p. 33).
- [156] K. Labusch and C. Neudecker, “Named Entity Disambiguation and Linking Historic Newspaper OCR with BERT.,” *CLEF (Working Notes)*, 2020 (cit. on p. 33).
- [157] B. Huang, H. Wang, T. Wang, Y. Liu, and Y. Liu, “Entity Linking for Short Text Using Structured Knowledge Graph via Multi-Grained Text Matching.,” *INTERSPEECH*, 2020 4178 (cit. on p. 33).
- [158] V. Provatorova, S. Vakulenko, E. Kanoulas, K. Dercksen, and J. M. van Hulst, *Named entity recognition and linking on historical newspapers: UvA. ILPS & REL at CLEF HIPE 2020*, (2020) (cit. on p. 33).
- [159] I. O. Mulang, K. Singh, A. Vyas, S. Shekarpour, M.-E. Vidal, J. Lehmann, and S. Auer, “Encoding knowledge graph entity aliases in attentive neural network for wikidata entity linking,” *International Conference on Web Information Systems Engineering*, Springer, 2020 328 (cit. on p. 33).
- [160] D. Sorokin and I. Gurevych, “Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories,” *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, Association for Computational Linguistics, 2018 65, URL: <https://www.aclweb.org/anthology/S18-2007> (cit. on pp. 33, 65).
- [161] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer Networks,” *Advances in Neural Information Processing Systems*, ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, vol. 28, Curran Associates, Inc., 2015, URL: <https://proceedings.neurips.cc/paper/2015/file/29921001f2f04bd3baee84a12e98098f-Paper.pdf> (cit. on pp. 33, 84).
- [162] D. Banerjee, D. Chaudhuri, M. Dubey, and J. Lehmann, “PNEL: Pointer Network based End-To-End Entity Linking over Knowledge Graphs,” *International Semantic Web Conference*, Springer, 2020 21 (cit. on pp. 33, 64–66, 70).
- [163] C. Möller, J. Lehmann, and R. Usbeck, *Survey on English Entity Linking on Wikidata: Datasets and approaches*, Semantic Web (2022) 1 (cit. on p. 33).
- [164] A. Delpuch, *Opentapioca: Lightweight entity linking for wikidata*, arXiv preprint arXiv:1904.09131 (2019) (cit. on pp. 34, 65).

- 
- [165] M. Miwa and M. Bansal, “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures,” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2016 1105, URL: <https://www.aclweb.org/anthology/P16-1105> (cit. on p. 34).
- [166] I. O. Mulang, K. Singh, and F. Orlandi, “Matching natural language relations to knowledge graph properties for question answering,” *Proceedings of the 13th International Conference on Semantic Systems*, 2017 89 (cit. on pp. 34, 65, 66).
- [167] K. Singh, I. O. Mulang’, I. Lytra, M. Y. Jaradeh, A. Sakor, M.-E. Vidal, C. Lange, and S. Auer, “Capturing knowledge in semantically-typed relational patterns to enhance relation linking,” *Proceedings of the Knowledge Capture Conference*, 2017 1 (cit. on p. 34).
- [168] J. Z. Pan, M. Zhang, K. Singh, F. v. Harmelen, J. Gu, and Z. Zhang, “Entity enabled relation linking,” *International Semantic Web Conference*, Springer, 2019 523 (cit. on p. 34).
- [169] J. Ganitkevitch, B. Van Durme, and C. Callison-Burch, “PPDB: The Paraphrase Database,” *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2013 758, URL: <https://aclanthology.org/N13-1092> (cit. on p. 34).
- [170] N. Nakashole, G. Weikum, and F. Suchanek, “PATTY: A Taxonomy of Relational Patterns with Semantic Types,” *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 2012 1135, URL: <https://aclanthology.org/D12-1104> (cit. on p. 34).
- [171] A. Sakor, I. Onando Mulang’, K. Singh, S. Shekarpour, M. Esther Vidal, J. Lehmann, and S. Auer, “Old is Gold: Linguistic Driven Approach for Entity and Relation Linking of Short Text,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019 2336, URL: <https://aclanthology.org/N19-1243> (cit. on p. 34).
- [172] N. Mihindukulasooriya, G. Rossiello, P. Kapanipathi, I. Abdelaziz, S. Ravishankar, M. Yu, A. Gliozzo, S. Roukos, and A. Gray, “Leveraging semantic parsing for relation linking over knowledge bases,” *International Semantic Web Conference*, Springer, 2020 402 (cit. on p. 34).
- [173] T. Naseem, S. Ravishankar, N. Mihindukulasooriya, I. Abdelaziz, Y.-S. Lee, P. Kapanipathi, S. Roukos, A. Gliozzo, and A. Gray, “A Semantics-aware Transformer Model of Relation Linking for Knowledge Base Question Answering,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*,

- Association for Computational Linguistics, 2021 256,  
URL: <https://aclanthology.org/2021.acl-short.34> (cit. on p. 34).
- [174] Y. Zhang, P. Qi, and C. D. Manning, “Graph Convolution over Pruned Dependency Trees Improves Relation Extraction,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 2205,  
URL: <https://www.aclweb.org/anthology/D18-1244> (cit. on p. 34).
- [175] P. Wu, S. Huang, R. Weng, Z. Zheng, J. Zhang, X. Yan, and J. Chen, “Learning Representation Mapping for Relation Detection in Knowledge Base Question Answering,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 6130,  
URL: <https://www.aclweb.org/anthology/P19-1616> (cit. on p. 34).
- [176] R. Nedelchev, D. Chaudhuri, J. Lehmann, and A. Fischer, *End-to-End Entity Linking and Disambiguation leveraging Word and Knowledge Graph Embeddings*, arXiv preprint arXiv:2002.11143 (2020) (cit. on p. 34).
- [177] X. Huang, J. Zhang, D. Li, and P. Li, “Knowledge graph embedding based question answering,” *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019 105 (cit. on p. 34).
- [178] O. Levy, M. Seo, E. Choi, and L. Zettlemoyer, “Zero-Shot Relation Extraction via Reading Comprehension,” *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, 2017 333 (cit. on p. 34).
- [179] Y. Zhao, J. Huang, W. Hu, Q. Chen, X. Qiu, C. Huo, and W. Ren, “Implicit Relation Linking for Question Answering over Knowledge Graph,” *Findings of the Association for Computational Linguistics: ACL 2022*, Association for Computational Linguistics, 2022 3956,  
URL: <https://aclanthology.org/2022.findings-acl.312> (cit. on p. 34).
- [180] Y. Lan and J. Jiang, “Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 969,  
URL: <https://aclanthology.org/2020.acl-main.91> (cit. on p. 34).
- [181] C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao, “Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017 23,  
URL: <https://aclanthology.org/P17-1003> (cit. on p. 34).



- 
- [182] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, “Key-Value Memory Networks for Directly Reading Documents,” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2016 1400, URL: <https://aclanthology.org/D16-1147> (cit. on p. 34).
- [183] H. Sun, T. Bedrax-Weiss, and W. Cohen, “PullNet: Open Domain Question Answering with Iterative Retrieval on Knowledge Bases and Text,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 2380, URL: <https://aclanthology.org/D19-1242> (cit. on p. 34).
- [184] G. Maheshwari, P. Trivedi, D. Lukovnikov, N. Chakraborty, A. Fischer, and J. Lehmann, “Learning to rank query graphs for complex question answering over knowledge graphs,” *International semantic web conference*, Springer, 2019 487 (cit. on pp. 34, 55, 56).
- [185] S. Vakulenko, J. D. Fernandez Garcia, A. Polleres, M. de Rijke, and M. Cochez, “Message passing for complex question answering over knowledge graphs,” *Proceedings of the 28th acm international conference on information and knowledge management*, 2019 1431 (cit. on pp. 34, 56).
- [186] Y. Guo, Z. Pan, and J. Heflin, *LUBM: A benchmark for OWL knowledge base systems*, *Journal of Web Semantics* **3** (2005) 158 (cit. on p. 35).
- [187] A. Owens, N. Gibbins, et al., *Effective benchmarking for RDF stores using synthetic data*, (2008) (cit. on p. 35).
- [188] C. Bizer and A. Schultz, “Benchmarking the performance of storage systems that expose SPARQL endpoints,” *Proc. 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS)*, Citeseer, 2008 39 (cit. on p. 35).
- [189] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, “SP<sup>2</sup>Bench: a SPARQL performance benchmark,” *2009 IEEE 25th International Conference on Data Engineering*, IEEE, 2009 222 (cit. on p. 35).
- [190] P. Haase, T. Mathäß, and M. Ziller, “An evaluation of approaches to federated query processing over linked data,” *Proceedings of the 6th international conference on semantic systems*, 2010 1 (cit. on p. 35).
- [191] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri, “Test-driven evaluation of linked data quality,” *Proceedings of the 23rd international conference on World Wide Web*, 2014 747 (cit. on p. 35).
- [192] O. Görlitz, M. Thimm, and S. Staab, “Splodge: Systematic generation of sparql benchmark queries for linked open data,” *International Semantic Web Conference*, Springer, 2012 116 (cit. on p. 35).

- [193] S. Qiao and Z. M. Özsoyoğlu, “RBench: Application-specific RDF benchmarking,” *Proceedings of the 2015 acm sigmod international conference on management of data*, 2015 1825 (cit. on p. 35).
- [194] H. Dibowski and K. Kabitzsch, *Ontology-based device descriptions and device repository for building automation devices*, *EURASIP Journal on Embedded Systems* **2011** (2011) 1 (cit. on p. 35).
- [195] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer, et al., *Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia*, *Semantic web* **6** (2015) 167 (cit. on pp. 35, 55, 73).
- [196] C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano, “Template-based question answering over RDF data,” *WWW*, 2012 (cit. on p. 35).
- [197] G. Aluç, O. Hartig, M. T. Özsu, and K. Daudjee, “Diversified stress testing of RDF data management systems,” *International Semantic Web Conference*, Springer, 2014 197 (cit. on pp. 35, 74).
- [198] G. Bagan, A. Bonifati, R. Ciucanu, G. H. Fletcher, A. Lemay, and N. Advokaat, *gMark: Schema-driven generation of graphs and queries*, *IEEE Transactions on Knowledge and Data Engineering* **29** (2016) 856 (cit. on pp. 35, 74).
- [199] G. Zenz, X. Zhou, E. Minack, W. Siberski, and W. Nejdl, *From keywords to semantic queries—Incremental query construction on the Semantic Web*, *Journal of Web Semantics* **7** (2009) 166 (cit. on p. 35).
- [200] T. Soru, E. Marx, A. Valdestilhas, D. Esteves, D. Moussallem, and G. Publio, “Neural Machine Translation for Query Construction and Composition,” 2018, URL: <https://arxiv.org/abs/1806.10478> (cit. on pp. 35, 74, 81, 83, 84, 86).
- [201] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural computation* **9** (1997) 1735 (cit. on p. 35).
- [202] K. S. Tai, R. Socher, and C. D. Manning, “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks,” *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2015 1556, URL: <https://aclanthology.org/P15-1150> (cit. on pp. 35, 83).
- [203] J. Xu and W. B. Croft, “Query expansion using local and global document analysis,” *Acm sigir forum*, vol. 51, 2, ACM New York, NY, USA, 2017 168 (cit. on p. 36).
- [204] C. Carpineto and G. Romano, *A survey of automatic query expansion in information retrieval*, *Acm Computing Surveys (CSUR)* **44** (2012) 1 (cit. on p. 36).
- [205] K. Lee, M.-W. Chang, and K. Toutanova, “Latent Retrieval for Weakly Supervised Open Domain Question Answering,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 6086, URL: <https://aclanthology.org/P19-1612> (cit. on p. 36).

- 
- [206] M. Seo, J. Lee, T. Kwiatkowski, A. Parikh, A. Farhadi, and H. Hajishirzi, “Real-Time Open-Domain Question Answering with Dense-Sparse Phrase Index,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 4430, URL: <https://aclanthology.org/P19-1436> (cit. on p. 36).
- [207] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, “REALM: Retrieval-Augmented Language Model Pre-Training,” *Proceedings of the 37th International Conference on Machine Learning, ICML’20*, JMLR.org, 2020 (cit. on pp. 36, 112).
- [208] Y. Nie, S. Wang, and M. Bansal, “Revealing the Importance of Semantic Retrieval for Machine Reading at Scale,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 2553, URL: <https://aclanthology.org/D19-1258> (cit. on p. 36).
- [209] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional Attention Flow for Machine Comprehension,” *International Conference on Learning Representations*, 2017, URL: <https://openreview.net/forum?id=HJ0UKP9ge> (cit. on p. 36).
- [210] K. Nishida, I. Saito, A. Otsuka, H. Asano, and J. Tomita, “Retrieve-and-Read: Multi-Task Learning of Information Retrieval and Reading Comprehension,” *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM ’18*, Association for Computing Machinery, 2018 647, ISBN: 9781450360142, URL: <https://doi.org/10.1145/3269206.3271702> (cit. on p. 36).
- [211] O. Khattab, C. Potts, and M. Zaharia, *Relevance-guided Supervision for OpenQA with ColBERT*, *Transactions of the Association for Computational Linguistics* **9** (2021) 929, URL: <https://aclanthology.org/2021.tacl-1.55> (cit. on p. 36).
- [212] T. Zhao, X. Lu, and K. Lee, “SPARTA: Efficient Open-Domain Question Answering via Sparse Transformer Matching Retrieval,” *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021 565, URL: <https://aclanthology.org/2021.naacl-main.47> (cit. on p. 36).
- [213] Y. Zhang, P. Nie, X. Geng, A. Ramamurthy, L. Song, and D. Jiang, *DC-BERT: Decoupling Question and Document for Efficient Contextual Encoding*, *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020) (cit. on p. 36).
- [214] S. Min, D. Chen, L. Zettlemoyer, and H. Hajishirzi, *Knowledge Guided Text Retrieval and Reading for Open Domain Question Answering*, arXiv preprint arXiv:1911.03868 (2019) (cit. on p. 37).

- [215] R. Das, S. Dhuliawala, M. Zaheer, and A. McCallum, “Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering,” *International Conference on Learning Representations*, 2019, URL: <https://openreview.net/forum?id=HkfPSh05K7> (cit. on p. 37).
- [216] W. Xiong, X. Li, S. Iyer, J. Du, P. Lewis, W. Y. Wang, Y. Mehdad, S. Yih, S. Riedel, D. Kiela, and B. Oguz, “Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval,” *International Conference on Learning Representations*, 2021, URL: <https://openreview.net/forum?id=EMHoBG0avc1> (cit. on p. 37).
- [217] Y. Feldman and R. El-Yaniv, “Multi-Hop Paragraph Retrieval for Open-Domain Question Answering,” *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019 2296, URL: <https://aclanthology.org/P19-1222> (cit. on p. 37).
- [218] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen, “Generation-Augmented Retrieval for Open-Domain Question Answering,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021 4089, URL: <https://aclanthology.org/2021.acl-long.316> (cit. on p. 37).
- [219] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, “Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering,” *International Conference on Learning Representations*, 2020 (cit. on p. 37).
- [220] Y. Lin, H. Ji, Z. Liu, and M. Sun, “Denoising Distantly Supervised Open-Domain Question Answering,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2018 1736, URL: <https://aclanthology.org/P18-1161> (cit. on p. 37).
- [221] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, and C. Xiong, “Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering,” *International Conference on Learning Representations*, 2020, URL: <https://openreview.net/forum?id=SJgVHkrYDH> (cit. on p. 37).
- [222] G. Izacard and E. Grave, “Distilling Knowledge from Reader to Retriever for Question Answering,” *International Conference on Learning Representations*, 2021, URL: <https://openreview.net/forum?id=NTEz-6wysdb> (cit. on p. 37).
- [223] C. Tan, F. Wei, N. Yang, B. Du, W. Lv, and M. Zhou, “S-net: From answer extraction to answer synthesis for machine reading comprehension,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 1, 2018 (cit. on p. 37).

- 
- [224] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al.,  
*Retrieval-augmented generation for knowledge-intensive nlp tasks*,  
Advances in Neural Information Processing Systems **33** (2020) 9459 (cit. on p. 37).
- [225] F. Zhu, W. Lei, C. Wang, J. Zheng, S. Poria, and T.-S. Chua,  
*Retrieving and reading: A comprehensive survey on open-domain question answering*,  
arXiv preprint arXiv:2101.00774 (2021) (cit. on p. 37).
- [226] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang,  
“SQuAD: 100,000+ Questions for Machine Comprehension of Text,”  
*Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*,  
Association for Computational Linguistics, 2016 2383,  
URL: <https://aclanthology.org/D16-1264> (cit. on p. 37).
- [227] Y. Jing, D. Xiong, and Z. Yan, “BiPaR: A Bilingual Parallel Dataset for Multilingual and Cross-lingual Reading Comprehension on Novels,” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 2452,  
URL: <https://aclanthology.org/D19-1249> (cit. on p. 37).
- [228] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, *Teaching machines to read and comprehend*,  
Advances in neural information processing systems **28** (2015) (cit. on p. 37).
- [229] W. Xiong, J. Wu, H. Wang, V. Kulkarni, M. Yu, S. Chang, X. Guo, and W. Y. Wang,  
“TWEETQA: A Social Media Focused Question Answering Dataset,”  
*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*,  
Association for Computational Linguistics, 2019 5020,  
URL: <https://aclanthology.org/P19-1496> (cit. on p. 37).
- [230] S. Šuster and W. Daelemans,  
“CliCR: a Dataset of Clinical Case Reports for Machine Reading Comprehension,”  
*Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*,  
Association for Computational Linguistics, 2018 1551,  
URL: <https://aclanthology.org/N18-1140> (cit. on p. 37).
- [231] M. Saeidi, M. Bartolo, P. Lewis, S. Singh, T. Rocktäschel, M. Sheldon, G. Bouchard, and S. Riedel, “Interpretation of Natural Language Rules in Conversational Machine Reading,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 2087,  
URL: <https://aclanthology.org/D18-1233> (cit. on p. 37).
- [232] S. Reddy, D. Chen, and C. D. Manning,  
*Coqa: A conversational question answering challenge*,  
Transactions of the Association for Computational Linguistics **7** (2019) 249 (cit. on p. 37).

- [233] A. Kembhavi, M. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi, “Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension,” *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, 2017 4999 (cit. on p. 37).
- [234] C.-W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau, “How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation,” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2016 2122, URL: <https://aclanthology.org/D16-1230> (cit. on pp. 38, 47, 97, 98).
- [235] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in neural information processing systems*, 2013 3111 (cit. on p. 38).
- [236] J. Matsuo, M. Komachi, and K. Sudoh, *Word-alignment-based segment-level machine translation evaluation using word embeddings*, arXiv preprint arXiv:1704.00380 (2017) (cit. on p. 38).
- [237] R. Nedelchev, R. Usbeck, and J. Lehmann, “Treating Dialogue Quality Evaluation as an Anomaly Detection Problem,” English, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, European Language Resources Association, 2020 508, ISBN: 979-10-95546-34-4, URL: <https://aclanthology.org/2020.lrec-1.64> (cit. on p. 38).
- [238] R. Lowe, M. Noseworthy, I. V. Serban, N. Angelard-Gontier, Y. Bengio, and J. Pineau, “Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses,” *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, 2017 1116, URL: <https://www.aclweb.org/anthology/P17-1103> (cit. on p. 38).
- [239] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” *International Conference on Learning Representations*, 2020, URL: <https://openreview.net/forum?id=SkeHuCVFDr> (cit. on pp. 38, 98, 105).
- [240] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *International Conference on Learning Representations (ICLR)*, 2017 (cit. on pp. 40, 43).
- [241] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, “Key-Value Retrieval Networks for Task-Oriented Dialogue,” *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, Association for Computational Linguistics, 2017 37, URL: <https://aclanthology.org/W17-5506> (cit. on pp. 40, 45–47).

- 
- [242] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, “A Network-based End-to-End Trainable Task-oriented Dialogue System,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Association for Computational Linguistics, 2017 438, URL: <https://aclanthology.org/E17-1042> (cit. on pp. 40, 45, 46).
- [243] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, “MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling,” *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2018 5016, URL: <https://aclanthology.org/D18-1547> (cit. on pp. 40, 45, 46).
- [244] L. J. Ba, J. R. Kiros, and G. E. Hinton, *Layer Normalization*, CoRR **abs/1607.06450** (2016), arXiv: 1607.06450, URL: <http://arxiv.org/abs/1607.06450> (cit. on pp. 43, 77).
- [245] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, *Roberta: A robustly optimized bert pretraining approach*, arXiv preprint arXiv:1907.11692 (2019) (cit. on p. 43).
- [246] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, “QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering,” *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021 535, URL: <https://aclanthology.org/2021.naacl-main.45> (cit. on p. 43).
- [247] S. Vashishth, P. Yadav, M. Bhandari, and P. Talukdar, “Confidence-based Graph Convolutional Networks for Semi-Supervised Learning,” *Proceedings of Machine Learning Research*, ed. by K. Chaudhuri and M. Sugiyama, vol. 89, Proceedings of Machine Learning Research, PMLR, 2019 1792, URL: <http://proceedings.mlr.press/v89/vashishth19a.html> (cit. on p. 43).
- [248] A. Fan, M. Lewis, and Y. Dauphin, *Hierarchical neural story generation*, arXiv preprint arXiv:1805.04833 (2018) (cit. on pp. 45, 79).
- [249] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” *International Conference on Learning Representations*, 2019, URL: <https://openreview.net/forum?id=Bkg6RiCqY7> (cit. on pp. 46, 81).
- [250] D. Hendrycks and K. Gimpel, *Gaussian error linear units (gelus)*, arXiv preprint arXiv:1606.08415 (2016) (cit. on pp. 46, 81, 104).
- [251] R. Gangi Reddy, D. Contractor, D. Raghu, and S. Joshi, “Multi-Level Memory for Task Oriented Dialogs,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*,

- Association for Computational Linguistics, 2019 3744,  
URL: <https://aclanthology.org/N19-1375> (cit. on pp. 47, 49).
- [252] L. Qin, Y. Liu, W. Che, H. Wen, Y. Li, and T. Liu,  
“Entity-Consistent End-to-end Task-Oriented Dialogue System with KB Retriever,”  
*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 133,  
URL: <https://aclanthology.org/D19-1013> (cit. on pp. 47, 49).
- [253] W. He, M. Yang, R. Yan, C. Li, Y. Shen, and R. Xu, “Amalgamating Knowledge from Two Teachers for Task-oriented Dialogue System with Adversarial Training,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020 3498,  
URL: <https://aclanthology.org/2020.emnlp-main.281> (cit. on pp. 47, 49).
- [254] L. Qin, X. Xu, W. Che, Y. Zhang, and T. Liu,  
“Dynamic Fusion Network for Multi-Domain End-to-end Task-Oriented Dialog,”  
*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 6344,  
URL: <https://www.aclweb.org/anthology/2020.acl-main.565>  
(cit. on pp. 47, 49).
- [255] Z. He, J. Wang, and J. Chen,  
“Task-Oriented Dialog Generation with Enhanced Entity Representation.,” *INTERSPEECH*, 2020 3905 (cit. on pp. 47–49).
- [256] Z. He, Y. He, Q. Wu, and J. Chen,  
“Fg2seq: Effectively Encoding Knowledge for End-To-End Task-Oriented Dialog,”  
*ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020 8029 (cit. on pp. 47–49).
- [257] D. Raghu, A. Jain, Mausam, and S. Joshi,  
“Constraint based Knowledge Base Distillation in End-to-End Task Oriented Dialogs,”  
*Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, 2021 5051,  
URL: <https://aclanthology.org/2021.findings-acl.448>  
(cit. on pp. 47–49).
- [258] J. Novikova, O. Dušek, A. Cercas Curry, and V. Rieser,  
“Why We Need New Evaluation Metrics for NLG,”  
*Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2017 2241,  
URL: <https://aclanthology.org/D17-1238> (cit. on pp. 47, 98, 103).
- [259] W. Zhao, T. Chung, A. Goyal, and A. Metallinou,  
“Simple Question Answering with Subgraph Ranking and Joint-Scoring,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*,



- 
- Association for Computational Linguistics, 2019 324,  
URL: <https://www.aclweb.org/anthology/N19-1029> (cit. on p. 56).
- [260] S. Mohammed, P. Shi, and J. Lin, “Strong Baselines for Simple Question Answering over Knowledge Graphs with and without Neural Networks,”  
*Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, 2018 291,  
URL: <https://www.aclweb.org/anthology/N18-2047> (cit. on p. 56).
- [261] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, *Shortcut learning in deep neural networks*,  
*Nature Machine Intelligence* **2** (2020) 665, ISSN: 2522-5839,  
URL: <https://doi.org/10.1038/s42256-020-00257-z> (cit. on p. 56).
- [262] A. Fabbri, P. Ng, Z. Wang, R. Nallapati, and B. Xiang, “Template-Based Question Generation from Retrieved Sentences for Improved Unsupervised Question Answering,”  
*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 4508,  
URL: <https://aclanthology.org/2020.acl-main.413> (cit. on p. 56).
- [263] O. Ram, Y. Kirstain, J. Berant, A. Globerson, and O. Levy,  
“Few-Shot Question Answering by Pretraining Span Selection,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021 3066,  
URL: <https://aclanthology.org/2021.acl-long.239> (cit. on p. 56).
- [264] D. B. Nguyen, A. Abujabal, K. Tran, M. Theobald, and G. Weikum,  
*Query-driven on-the-fly knowledge base construction*,  
*Proceedings of the VLDB Endowment* **11** (2017) 66 (cit. on pp. 56, 65, 66).
- [265] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, “LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia,”  
*Proceedings of the 18th International Semantic Web Conference (ISWC)*, Springer, 2019 (cit. on pp. 57, 64).
- [266] X. Lin, H. Li, H. Xin, Z. Li, and L. Chen,  
*KB Pearl: a knowledge base population system supported by joint entity and relation linking*,  
*Proceedings of the VLDB Endowment* **13** (2020) 1035 (cit. on pp. 57, 64–66).
- [267] R. Usbeck, A.-C. N. Ngomo, B. Haarmann, A. Krithara, M. Röder, and G. Napolitano,  
“7th open challenge on question answering over linked data (QALD-7),”  
*Semantic web evaluation challenge*, Springer, 2017 59 (cit. on pp. 57, 64).
- [268] J. Johnson, M. Douze, and H. Jégou, *Billion-scale similarity search with GPUs*,  
*IEEE Transactions on Big Data* (2019) 1 (cit. on p. 59).

- [269] S. Zhang, Y. Tay, L. Yao, and Q. Liu, “Quaternion Knowledge Graph Embeddings,” *Advances in Neural Information Processing Systems*, ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, vol. 32, Curran Associates, Inc., 2019, URL: <https://proceedings.neurips.cc/paper/2019/file/d961e9f236177d65d21100592edb0769-Paper.pdf> (cit. on p. 62).
- [270] J. Berant, A. Chou, R. Frostig, and P. Liang, “Semantic parsing on freebase from question-answer pairs,” *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013 1533 (cit. on p. 64).
- [271] P. N. Mendes, J. Daiber, M. Jakob, and C. Bizer, “Evaluating dbpedia spotlight for the tac-kbp entity linking task,” *Proceedings of the TAC-KBP 2011 Workshop*, vol. 116, 2011 118 (cit. on p. 65).
- [272] P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer, “DBpedia spotlight: shedding light on the web of documents,” *Proceedings of the 7th international conference on semantic systems*, 2011 1 (cit. on pp. 65, 66).
- [273] P. Ferragina and U. Scaiella, “Tagme: on-the-fly annotation of short text fragments (by wikipedia entities),” *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010 1625 (cit. on pp. 65, 66).
- [274] L. Del Corro and R. Gemulla, “Clausie: clause-based open information extraction,” *Proceedings of the 22nd international conference on World Wide Web*, 2013 355 (cit. on p. 65).
- [275] A. Sakor, K. Singh, A. Patel, and M.-E. Vidal, “Falcon 2.0: An entity and relation linking tool over wikidata,” *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020 3141 (cit. on p. 65).
- [276] W.-t. Yih, M.-W. Chang, X. He, and J. Gao, “Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base,” *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2015 1321, URL: <http://www.aclweb.org/anthology/P15-1128> (cit. on pp. 65, 67).
- [277] J. Bao, N. Duan, Z. Yan, M. Zhou, and T. Zhao, “Constraint-Based Question Answering with Knowledge Graph,” *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, The COLING 2016 Organizing Committee, 2016 2503, URL: <https://www.aclweb.org/anthology/C16-1236> (cit. on p. 67).

- 
- [278] Y. Khan, M. Saleem, A. Iqbal, M. Mehdi, A. Hogan, A.-C. N. Ngomo, S. Decker, and R. Sahay, “SAFE: policy aware SPARQL query federation over RDF data cubes,” *Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, Berlin, Germany, December 9-11, 2014*. 2014 (cit. on p. 73).
- [279] M. Kulmanov, S. Kafkas, A. Karwath, A. Malic, G. Gkoutos, M. Dumontier, and R. Hoehndorf, *Vec2SPARQL: integrating SPARQL queries and knowledge graph embeddings*, (2018) (cit. on p. 73).
- [280] S. Rudolph, L. Schweizer, and Z. Yao, “SPARQL Queries over Ontologies Under the Fixed-Domain Semantics,” *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2019 486 (cit. on p. 73).
- [281] S. Purkayastha, S. Dana, D. Garg, D. Khandelwal, and G. Bhargav, *Knowledge Graph Question Answering via SPARQL Silhouette Generation*, arXiv preprint arXiv:2109.09475 (2021) (cit. on p. 74).
- [282] Y. Chen, H. Li, G. Qi, T. Wu, and T. Wang, *Outlining and Filling: Hierarchical Query Graph Generation for Answering Complex Questions over Knowledge Graph*, arXiv preprint arXiv:2111.00732 (2021) (cit. on p. 74).
- [283] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, “Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia,” *International semantic web conference*, Springer, 2019 69 (cit. on pp. 75, 80).
- [284] E. Kacupaj, H. Zafar, J. Lehmann, and M. Maleshkova, “Vquanda: Verbalization question answering dataset,” *European Semantic Web Conference*, Springer, 2020 531 (cit. on pp. 75, 80).
- [285] N. Ngomo, *9th challenge on question answering over linked data (QALD-9)*, language 7 (2018) 58 (cit. on pp. 75, 80).
- [286] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos, “TaPas: Weakly Supervised Table Parsing via Pre-training,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 4320, URL: <https://aclanthology.org/2020.acl-main.398> (cit. on p. 76).
- [287] F. Galetzka, J. Rose, D. Schlangen, and J. Lehmann, “Space Efficient Context Encoding for Non-Task-Oriented Dialogue Generation with Graph Attention Transformer,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, 2021 7028, URL: <https://aclanthology.org/2021.acl-long.546> (cit. on p. 76).
- [288] S. W.-t. Yih, M.-W. Chang, X. He, and J. Gao, *Semantic parsing via staged query graph generation: Question answering with knowledge base*, (2015) (cit. on p. 77).
- [289] D. Sorokin, *Knowledge Graphs and Graph Neural Networks for Semantic Parsing*, (2021) (cit. on p. 77).

- [290] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *ICML*, 2010 807,  
URL: <https://icml.cc/Conferences/2010/papers/432.pdf> (cit. on p. 78).
- [291] A. Graves, *Sequence transduction with recurrent neural networks*, arXiv preprint arXiv:1211.3711 (2012) (cit. on p. 79).
- [292] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Audio Chord Recognition with Recurrent Neural Networks.,” *ISMIR*, Citeseer, 2013 335 (cit. on p. 79).
- [293] M. Honnibal and M. Johnson, “An Improved Non-monotonic Transition System for Dependency Parsing,” *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2015 1373,  
URL: <https://aclanthology.org/D15-1162> (cit. on p. 81).
- [294] R. Dale and C. Mellish, “Towards evaluation in natural language generation,” *In Proceedings of First International Conference on Language Resources and Evaluation*, 1998 (cit. on p. 97).
- [295] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, “Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training,” *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2021 3554,  
URL: <https://www.aclweb.org/anthology/2021.naacl-main.278> (cit. on pp. 98, 103, 104, 108, 109).
- [296] M. Eric, L. Krishnan, F. Charette, and C. D. Manning, “Key-Value Retrieval Networks for Task-Oriented Dialogue,” *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, Association for Computational Linguistics, 2017 37 (cit. on pp. 98, 103).
- [297] A. Shimorina, C. Gardent, S. Narayan, and L. Perez-Beltrachini, *WebNLG Challenge: Human Evaluation Results*, Technical Report, Loria & Inria Grand Est, 2018,  
URL: <https://hal.archives-ouvertes.fr/hal-03007072> (cit. on pp. 98, 104).
- [298] F. Mairesse, M. Gašić, F. Jurčiček, S. Keizer, B. Thomson, K. Yu, and S. Young, “Phrase-Based Statistical Language Generation Using Graphical Models and Active Learning,” *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, Association for Computational Linguistics, 2010 1552 (cit. on pp. 98, 103, 108).
- [299] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2020 4902,

- 
- URL: <https://www.aclweb.org/anthology/2020.acl-main.442>  
(cit. on pp. 98, 108).
- [300] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, “Is bert really robust? a strong baseline for natural language attack on text classification and entailment,” *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 05, 2020 8018 (cit. on pp. 98, 103, 109).
- [301] J. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, “TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020 119 (cit. on pp. 98, 103, 108).
- [302] A. Kutuzov and E. Kuzmenko, “To Lemmatize or Not to Lemmatize: How Word Normalisation Affects ELMo Performance in Word Sense Disambiguation,” *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, Linköping University Electronic Press, 2019 22,  
URL: <https://aclanthology.org/W19-6203> (cit. on p. 101).
- [303] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, “Stanza: A Python Natural Language Processing Toolkit for Many Human Languages,” *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020 (cit. on pp. 101, 102).
- [304] A. Warstadt, A. Singh, and S. R. Bowman, *Neural Network Acceptability Judgments*, *Transactions of the Association for Computational Linguistics* 7 (2019) 625,  
URL: <https://aclanthology.org/Q19-1040> (cit. on pp. 102, 103).
- [305] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, “Language Models as Knowledge Bases?” *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019 2463,  
URL: <https://aclanthology.org/D19-1250> (cit. on p. 112).
- [306] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, *KEPLER: A unified model for knowledge embedding and pre-trained language representation*, *Transactions of the Association for Computational Linguistics* 9 (2021) 176 (cit. on p. 112).
- [307] D. Yu, C. Zhu, Y. Yang, and M. Zeng, “Jaket: Joint pre-training of knowledge graph and language understanding,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 10, 2022 11630 (cit. on p. 112).
- [308] L. He, S. Zheng, T. Yang, and F. Zhang, “KLMo: Knowledge Graph Enhanced Pretrained Language Model with Fine-Grained Relationships,” *Findings of the Association for Computational Linguistics: EMNLP 2021*, Association for Computational Linguistics, 2021 4536,  
URL: <https://aclanthology.org/2021.findings-emnlp.384>  
(cit. on p. 112).

- [309] R. Ma, X. Zhou, T. Gui, Y. Tan, L. Li, Q. Zhang, and X. Huang, “Template-free Prompt Tuning for Few-shot NER,” *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2022 5721, URL: <https://aclanthology.org/2022.naacl-main.420> (cit. on p. 112).
- [310] Y. Sun, Y. Zheng, C. Hao, and H. Qiu, “NSP-BERT: A Prompt-based Few-Shot Learner through an Original Pre-training Task — Next Sentence Prediction,” *Proceedings of the 29th International Conference on Computational Linguistics*, International Committee on Computational Linguistics, 2022 3233, URL: <https://aclanthology.org/2022.coling-1.286> (cit. on p. 113).
- [311] J. Zhang, P. Lertvittayakumjorn, and Y. Guo, “Integrating Semantic Knowledge to Tackle Zero-shot Text Classification,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, 2019 1031, URL: <https://aclanthology.org/N19-1108> (cit. on p. 113).

# List of Figures

---

1.1	Conversational and question answering systems explored in this thesis. . . . .	3
1.2	A breakdown of research questions. . . . .	5
2.1	An illustration of pre-trained language model family. Model names are in <b>bold</b> text. Figure adapted from THUNLP. . . . .	14
2.2	Transformer model architecture (Figure taken and adapted from Vaswani <i>et al.</i> [37]).	15
2.3	An illustration of a sub-graph of the Wikidata knowledge graph. The entity and relation IDs are shown in the text with black and yellow background, respectively. . .	17
2.4	An example of hyper-relational model (Figure from Galkin <i>et al.</i> [83]). . . . .	18
2.5	A high-level illustration of task-oriented dialogue system pipeline. . . . .	19
2.6	An illustration of SPARQL query components. . . . .	22
2.7	Architecture of a span prediction-based machine reading comprehension system. . .	24
2.8	Figure (a) depicts a high level overview of the Earth Mover’s Distance, where the weight-flow constraints are demonstrated in Figure (b). . . . .	27
2.9	Tree transformation. . . . .	28
2.10	TED operations required for transforming tree $\mathcal{T}_1$ into $\mathcal{T}_2$ . . . . .	28
2.11	BERT-Score (image from Zhang <i>et al.</i> [34]). . . . .	29
4.1	An illustration of knowledge-based multi-turn dialogue where DialoKG models the knowledge base as a Knowledge Graph. The user utterance is denoted by <b>Q</b> , the ground-truth response by <b>Gold</b> , and the words in <b>orange</b> are knowledge graph entries. 40	
4.2	A high-level overview of DialoKG is shown in Figure (a). Figure (b) depicts the input and output of the <i>Graph Weight Computer</i> module of DialoKG. . . . .	41
4.3	An illustration of knowledge and dialogue embedding techniques. . . . .	42
4.4	For the graph in Figure (a) and the question " <i>Find me the quickest route to the restaurant?</i> " the computation of the relation weight is shown in Figure (b), where $\hat{A} = A + I$ . . . . .	44
4.5	Distribution of human evaluation scores. . . . .	48
4.6	The interface of the annotation tool to obtained the human annotation scores. . . . .	50
4.7	Case study: comparison between ground truth and system-generated responses. . . . .	52
4.8	DialoKG’s performance on benchmark datasets for different number of dialogue contexts. 53	
5.1	An illustration of question answering over a knowledge graph. Figure a) depicts a sub-graph of the Wikidata KG, where Figure b) demonstrates sample question-answer pairs based on the example sub-graph. In the sample question-answer pairs, the surface form of the entities and relations are in red and green, respectively. . . . .	56

5.2	Figure (a) illustrates how the entity labels are encoded with Sentence-BERT and then indexed into a dense space using FAISS. The Indexing algorithm <i>IndexFlatIP</i> of FAISS, clusters similar entities together into the dense space. Figure b) demonstrates the candidate entity generation procedure given a detected entity mention. Sentence-BERT is used to obtain the vector representation of the entity mention <i>lionel</i> . The encoded vector is then passed to the FAISS module that performs a lookup into the dense space and generates $N$ candidate entities that are similar to the provided entity span, <i>lionel</i> . The red circle represents the given entity mention in the dense space, where the other circles inside the larger orange circle indicate similar entities around it.	59
5.3	An illustration of the entity disambiguation process. A small portion of the Wikidata graph is shown for demonstration purposes.	60
5.4	Figure a) depicts a $k$ -level tree (with $k=2$ ). Since a tree has many nodes and branches (edges), we present a toy example. Figure b) shows a forest consists of a set of trees constructed from the sub-graph of the linked entities. For the demonstration purpose, we show a forest consists of two trees. The red branches show the position of predicted relation in different trees. The green nodes represent the leaf nodes at level- $k$ , where the blue nodes refer to the intermediary nodes between the root and leaf nodes. Furthermore, the yellow nodes represent the predicted answer entity nodes connected by the red branches.	61
5.5	Inference time efficiency of the entity linking systems.	70
6.1	An illustration of a SPARQL query used to answer a natural question over Wikidata [16] and DBpedia [195]. Here, Q339, P398, P31, Q184246 are the Wikidata ID of <i>Pluto</i> , <i>child astronomical body</i> , <i>instance of</i> , and <i>moon of Pluto</i> , respectively.	73
6.2	An illustration of special embedding layers used in $SGPT_Q$ .	77
6.3	The question and knowledge embedding techniques used in $SGPT_{Q,\mathcal{K}}$ . The dotted red box indicates the separation of additional knowledge from the question.	77
6.4	System Architecture.	79
6.5	Question type-wise performance of $SGPT_{Q,\mathcal{K}}$ and baseline models on the test set of VQuAnDa and QALD-9.	82
6.6	Question type-wise performance of $SGPT_Q$ on LC-QuAD 2.0 test set.	82
6.7	Human evaluation score distribution.	85
7.1	The data pre-processing pipeline, showing how documents are stored into a dense space.	91
7.2	System architecture.	91
7.3	System demonstrator.	92
7.4	The in-house annotation tool used to collect question answer pairs for training the <i>Reader</i> module.	94
8.1	An example word alignment matrix for the reference sentence: " <i>tesla motors is founded by elon musk</i> " and its passive form: " <i>elon musk founded tesla motors</i> " is illustrated here.	100
8.2	An example hypothesis containing repetitive words.	101
8.3	Dependency trees of reference and hypothesis, pre-processed for the TED-SE calculation.	102
8.4	A high-level illustration of RoMe.	103
8.5	The annotation tool used by the annotators.	104



8.6 Correlation between the explored metrics. . . . . 107



# List of Tables

---

4.1	Dataset statistics. . . . .	46
4.2	Training parameters. . . . .	46
4.3	Decoding parameters. . . . .	46
4.4	Performance of DialoKG and baseline models on three benchmark datasets. Best scores in <b>bold</b> and second-best <u>underlined</u> . . . . .	47
4.5	Human evaluation results. . . . .	48
4.6	Domain-wise results on SMD dataset. . . . .	49
4.7	Domain-wise results on MWOZ dataset. . . . .	49
4.8	Ablation study. . . . .	51
4.9	Effect of triple selection on the performance. . . . .	51
5.1	Notation of the concepts used in Tree-KGQA. . . . .	58
5.2	Dataset statistics. . . . .	64
5.3	Performance of the entity linking component on LC-QuAD 2.0. . . . .	65
5.4	Performance of the entity linking component on the LC-QuAD 2.0 (KBpearl). . . . .	66
5.5	Performance of the entity linking component on the QALD-7-Wiki. . . . .	66
5.6	Performance of the relation linking component on the LC-QuAD 2.0 (KBpearl). . . . .	66
5.7	Performance of KGQA on WebQSP-WD test set. Models marked with (*) are the re-implementation from Sorokin <i>et al.</i> [22] to meet the KGQA task. . . . .	67
5.8	Component-wise results of Tree-KGQA. . . . .	67
5.9	Our introduced new baseline for the KGQA task on LC-QuAD 2.0 (KBpearl). . . . .	67
5.10	Ablation study. . . . .	68
5.11	Case study. . . . .	69
6.1	Comparison of SPARQL queries for two different questions with same wording. . . . .	74
6.2	Dataset statistics. . . . .	80
6.3	Statistics of question types. . . . .	81
6.4	Performance of SGPT and baseline models on three benchmark datasets. Best scores are in <b>bold</b> . F1 scores computed for the models with * are against the entity and relation set and do not consider all the tokens in the SPARQL query. . . . .	81
6.5	Results on data where baseline models could generate queries. . . . .	81
6.6	An illustration of query normalization. . . . .	83
6.7	Human evaluation results. . . . .	84
6.8	Ablation study. . . . .	86
6.9	Case study showing a comparison between SGPT and baseline system’s outputs. . . . .	86
6.10	Performance of entity and relation generation. . . . .	87

6.11	Three error cases where the texts highlighted in green indicate the correct entry in the reference query and red indicating wrong predication in the system generated query. The text in yellow shows the masked entity. . . . .	88
7.1	Dataset statistics. . . . .	95
7.2	Performance of the <i>Reader</i> component. . . . .	95
8.1	Metrics correlation with human judgment on system outputs from the WebNLG 2017 challenge. Here, $r$ : Pearson correlation co-efficient, $\rho$ : Spearman’s correlation co-efficient, $\tau$ : Kendall’s Tau. . . . .	105
8.2	Metrics Spearman correlation score against human judgment on perturbed texts. Here, $f$ : fluency, $s$ : semantic similarity, $g$ : grammatical correctness. . . . .	105
8.3	Spearman correlation ( $\rho$ ) scores computed from the metric scores with respect to the human evaluation scores on BAGEL and SFHOTEL. Baseline model’s results are reported form [113]. Here, <b>Info</b> , <b>Nat</b> and <b>Qual</b> refer to <i>informativeness</i> , <i>naturalness</i> , and <i>quality</i> , respectively. . . . .	106
8.4	Component-wise qualitative analysis. . . . .	106
8.5	Qualitative analysis. . . . .	107
8.6	Metrics Spearman’s correlation coefficient ( $\rho$ ) with human judgment on dialogue datasets. . . . .	107
8.7	Ablation Study. . . . .	108
9.1	A high-level overview of the contributions correspond to the research questions. . . .	111